

## OpenText™ AppEnhancer

### **Cloud Deployment Guide**

This document provides information about the containerization of Docker images for OpenText AppEnhancer.

EAXCORE230200-ACD-EN-1

---

**OpenText™ AppEnhancer  
Cloud Deployment Guide**  
EAXCORE230200-ACD-EN-1  
Rev.: 2023-June-12

**This documentation has been created for OpenText™ AppEnhancer 23.2.**

It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

**Open Text Corporation**

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111

Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440

Fax: +1-519-888-0677

Support: <https://support.opentext.com>

For more information, visit <https://www.opentext.com>

**Copyright © 2023 Open Text. All Rights Reserved.**

Trademarks owned by Open Text.

One or more patents may cover this product. For more information, please visit <https://www.opentext.com/patents>.

**Disclaimer**

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

---

# Table of Contents

<b>1</b>	<b>Overview .....</b>	<b>5</b>
1.1	Installation Requirements .....	5
<b>2</b>	<b>Configuring Docker containers .....</b>	<b>7</b>
2.1	Setting up the AppEnhancer Administrator Container .....	7
2.2	Setting up the AppEnhancer Web Access Container .....	8
2.3	Setting up the AppEnhancer Render Server Container .....	9
2.4	Setting up the AppEnhancer REST Services Container .....	10
2.5	Setting up the AppEnhancer Web Services Container .....	11
2.6	AppEnhancer Container Flag Properties .....	12
2.7	Creating an Impersonation Account in a Container .....	12
2.8	Enabling File Sharing Between a Host and a Container .....	13
<b>3</b>	<b>Troubleshooting .....</b>	<b>15</b>
3.1	Troubleshooting checklist .....	15



# Chapter 1

## Overview

OpenText AppEnhancer stores, organizes, and manages documents, files, and other business-critical information, and provides fast, security-controlled access to information from Microsoft™ Windows™ or web-based clients. AppEnhancer integrates document imaging, reports management, workflow, and document management services within an easy-to-use Windows-based system.

OpenText provides container images that can be used to install AppEnhancer with the Web Access and Render Server components, or the REST or Web Services containers in a generic Kubernetes cluster.

### 1.1 Installation Requirements

To perform the steps outlined in this document, you will need the items listed below. Refer to the AppEnhancer Release Notes for further information on the specific platforms and versions that are supported.



**Note:** The instructions in this document include examples of how to use the required third-party applications. To obtain additional in-depth information, consult the application's official documentation.

---

#### Microsoft Windows

You must be running Windows 10 Enterprise or Server 2019 with the latest patches installed.

---

#### A Configured Database

You must have a supported database set up and fully configured on the host machine.

---

#### AppEnhancer license

Your AppEnhancer product must have a valid license server running within your environment.

---

#### An Impersonation account

An impersonation account is included in all container images with the user name `admin` and `Appenhancer22` set as the password. There are two approaches to working with the impersonation account:

- Creating an account on your host machine that matches the impersonation user `admin` and then registering all the components using this user.
- Creating a custom impersonation user in all containers and registering all components using this user. For more information about creating the

impersonation user, see [“Creating an Impersonation Account in a Container”](#)  
on page 12.

---

## Chapter 2

# Configuring Docker containers

This section contains instructions on pulling the container images and setting them up on your machine.

## 2.1 Setting up the AppEnhancer Administrator Container

To set up the AppEnhancer Administrator container, you must complete the following steps:

1. On the Docker host machine, open a command prompt window with Administrator privileges.
2. You must create a named volume to share files between your containers and the host with `aev` set as the name of the volume. At the command prompt, use the following command:

```
docker volume create aev
```

3. Next, you must pull the container images from the OpenText registry to your machine. At the command prompt, use the following command:

```
docker pull registry.opentext.com/ae-webadministrator:22.2.0.0
```

4. With the container images downloaded to your host machine, you can run the Administrator container by using the following command:

```
docker run -it --name <containerName> -p <host-port>:8080 --network nat -v aev:c:\data <container>
```



**Note:** The storage root is set to `c:\data` (mounted volume).

You must set the variable values with the properties used in your environment. For more information about these container options, see [“AppEnhancer Container Flag Properties” on page 12](#).

A complete command should appear similar to the following:

```
docker run -it --name webadmin -p 9000:80 --network nat -v aev:c:\data registry.opentext.com/ae-webadministrator:22.2.0.0
```

5. You must either create a new AppEnhancer system or connect to an existing database.

- To create a new AppEnhancer system, run the following command:

```
.\ComponentSetup.exe NewSystem xscm <impersonationUser>  
<impersonationPassword>
```

A complete command should appear similar to the following:

```
.\ComponentSetup.exe NewSystem xscm admin Appenhancer22
```

- Alternatively, you can connect to an existing AppEnhancer database by running the following command:

```
.\ComponentSetup.exe ConnectDB Admin <impersonationUser>  
<impersonationPassword> "Provider=SQLOLEDB;Data  
Source=<server>;Persist Security Info=True;  
Password=<password>;User ID=sa;Initial Catalog=<database>"
```

6. Restart IIS for the changes to take effect by running the `iisreset` command.
7. On the Docker host machine, open a browser and go to `http://localhost:<port>/AppEnhancerAdmin`.
8. Set up the Administrator account.
9. Add the UNC storage paths.
10. Add the location of the license server.

## 2.2 Setting up the AppEnhancer Web Access Container

To set up the AppEnhancer Web Access container, you must complete the following steps:

1. Set up credentials in Web Access for the AppEnhancer Administrator.
2. Next, you must pull the container images from the OpenText registry to your machine. At the command prompt, use the following command:  

```
docker pull registry.opentext.com/ae-webaccess:22.2.0.0
```
3. On the Docker host machine, open a command prompt window with Administrator privileges.
4. Run the Web Access container by using the following command:  

```
docker run -it --name <containerName> -p <host-port>:8080 --network  
nat -v aev:c:\data <container>
```



**Note:** The storage root is set to `c:\data` (mounted volume).

You must set the variable values with the properties used in your environment. For more information about these container options, see [“AppEnhancer Container Flag Properties” on page 12](#).

A complete command should appear similar to the following:

```
docker run -it --name webaccess -p 9001:80 --network nat -v aev:c:  
\data registry.opentext.com/ae-webaccess:22.2.0.0
```

5. Register AppEnhancer Web Access to the Component Register Wizard (CRW) by running the following command:
 

```
.\ComponentSetup.exe ConnectDB WebServer "Provider=SQLOLEDB;Data Source=<server>;Persist Security Info=True; Password=<password>;User ID=sa;Initial Catalog=<database>"
```
6. Restart IIS for the changes to take effect by running the `iisreset` command.
7. On the Docker host machine, open a browser and go to `http://localhost:<port>/AppEnhancerAdmin`.


## 2.3 Setting up the AppEnhancer Render Server Container

To set up the AppEnhancer Render Server container, you must complete the following steps:

1. Set up credentials in Render Server for the AppEnhancer Administrator.
2. Next, you must pull the container images from the OpenText registry to your machine. At the command prompt, use the following command:
 

```
docker pull registry.opentext.com/ae-renderserver:22.2.0.0
```
3. On the Docker host machine, open a command prompt window with Administrator privileges.
4. Run the Render Server container by using the following command:
 

```
docker run -it --name <containerName> --network nat -v aev:c:\data <container>
```

 **Note:** The storage root is set to `c:\data` (mounted volume).

You must set the variable values with the properties used in your environment. For more information about these container options, see [“AppEnhancer Container Flag Properties” on page 12](#).

A complete command should appear similar to the following:

```
docker run -it --name renderserver --network nat -v aev:c:\data registry.opentext.com/ae-renderserver:22.2.0.0
```

5. Register AppEnhancer Render Server to CRW by running the following command:
 

```
.\ComponentSetup.exe ConnectDB Rendering "Provider=SQLOLEDB;Data Source=<Server>;Persist Security Info=True; Password=<password>;User ID=sa;Initial Catalog=<database>"
```

## 2.4 Setting up the AppEnhancer REST Services Container

To set up the AppEnhancer REST Services container, you must complete the following steps:

1. Set up credentials in REST Services for the AppEnhancer Administrator.
2. Next, you must pull the container images from the OpenText registry to your machine. At the command prompt, use the following command:

```
docker pull registry.opentext.com/ae-rest:22.2.0.0
```

3. On the Docker host machine, open a command prompt window with Administrator privileges.

4. Run the REST Services container by using the following command:

```
docker run -it --name <containerName> -p <host-port>:8080 --network nat -v aev:c:\data <container>
```



**Note:** The storage root is set to c:\data (mounted volume).

You must set the variable values with the properties used in your environment. For more information about these container options, see [“AppEnhancer Container Flag Properties” on page 12](#).

A complete command should appear similar to the following:

```
docker run -it --name restservice -p 9002:80 --network nat -v aev:c:\data registry.opentext.com/ae-rest:22.2.0.0
```

5. Register AppEnhancer REST Services to CRW by running the following command:

```
.\ComponentSetup.exe ConnectDB RESTService "Provider=SQLOLEDB;Data Source=<server>;Persist Security Info=True; Password=<password>;User ID=sa;Initial Catalog=<database>"
```



**Note:** The database connection used for the initial catalog must be in an existing AppEnhancer data source.

6. Restart IIS for the changes to take effect by running the `iisreset` command.
7. On the Docker host machine, open a browser and go to `http://localhost:<port>/AppEnhancerREST`.

## 2.5 Setting up the AppEnhancer Web Services Container

To set up the AppEnhancer Web Services container, you must complete the following steps:

1. Set up credentials in Web Services for the AppEnhancer Administrator.

2. Next, you must pull the container images from the OpenText registry to your machine. At the command prompt, use the following command:

```
docker pull registry.opentext.com/ae-webservices:22.2.0.0
```

3. On the Docker host machine, open a command prompt window with Administrator privileges.

4. Run the Web Services container by using the following command:

```
docker run -it --name <containerName> -p <host-port>:8080 --network nat -v aev:c:\data <container>
```



**Note:** The storage root is set to c:\data (mounted volume).

You must set the variable values with the properties used in your environment. For more information about these container options, see [“AppEnhancer Container Flag Properties” on page 12](#).

A complete command should appear similar to the following:

```
docker run -it --name webservice -p 9003:80 --network nat -v aev:c:\data registry.opentext.com/ae-webservices:22.2.0.0
```

5. Register AppEnhancer Web Services to CRW by running the following command:

```
.\ComponentSetup.exe ConnectDB AXServices "Provider=SQLOLEDB;Data Source=<Server>;Persist Security Info=True; Password=<password>;User ID=sa;Initial Catalog=<database>"
```



**Note:** The database connection used for the initial catalog must be in an existing AppEnhancer data source.

6. Restart IIS for the changes to take effect by running the `iisreset` command.

7. On the Docker host machine, open a browser and go to `http://localhost:<port>/AppEnhancerServices/AXServicesInterface.aspx`.

## 2.6 AppEnhancer Container Flag Properties

The AppEnhancer containers feature numerous configuration options that you set according to the properties for your environment when running commands.

---

**-name <containerName>**

Assigns a name to the container, where *<containerName>* is the name you want to use for the container.

---

**-p <containerPort>:<hostPort>**

Binds the container port running TestEngine to a port on the host machine. For example, `-p 9000:80` maps container port 9000 to host port 80.

---

**-network nat**

Connects the container to the specified network, **nat**.

---

**-v <volumeName>:<folderLocation>**

Bind mounts a volume into your container. For example, `-v aev:c:\data`.

---

## 2.7 Creating an Impersonation Account in a Container

A PowerShell file to create a custom impersonation account in a container is included with all container images. The file is named `AddAccountToLogonAsService.ps1` and is stored in the `C:\SetupTemp` folder.

1. In a container, go to the `C:\SetupTemp` folder.
2. To run the command used to create a new impersonation account, type:  
`net user <impersonationAccountName> <password> /ADD`
3. Next, you must run the command to add the impersonation account to the container:

```
powershell ".\AddAccountToLogonAsService.ps1"  
"<impersonationAccountName>"
```

## 2.8 Enabling File Sharing Between a Host and a Container

A bind mount volume linking a folder inside a container and a folder on the host machine can be established to share files. When using the following command to enable file sharing, it creates a folder named `aev` in your current user's home folder and bind mounts it to the `C:\data` folder in the container. Changes to the contents of these folders are reflected in real time.

To enable file sharing between a host and container, use the following command argument:

```
docker run -it --name <containerName> -p <host-port>:8080 --network  
<nat> -v aev:c:\data <container>
```



## Chapter 3

# Troubleshooting

If you are having difficulty setting up the containers, you can try the following methods to help diagnose the issue.

### 3.1 Troubleshooting checklist

You can troubleshoot an issue by checking the following areas to determine why the set up process is not working as expected so you can resolve the problem.

#### Checking the event logs inside the Docker container

The event logs record messages that may be helpful in finding a solution. To check the event log in a Docker container:

1. On the Docker host machine, open a command prompt window with Administrator privileges.
2. Create a PowerShell session for the container by using the following command:  
`docker exec -it <containerName> powershell`
3. Next, you can retrieve the latest event logs by using the following command:  
`Get-Eventlog -newest 20 application`



**Note:** You can also inspect the container's index values to verify whether the configuration settings used are correct. After running the command to retrieve the event logs, the corresponding index IDs are shown. To retrieve the index values from an event log, run the following command:

```
(Get-Eventlog -index <indexID> application).message
```

#### Debugging a Docker container

The AppEnhancer containers can be debugged in many third-party debugging tools, such as Microsoft® Visual Studio with a Docker debug extension installed.

#### Verifying the file paths in the container

Confirm that the file paths you are using correspond to those set inside the container. These defaults are used for file paths in the containers:

---

##### Component Register Wizard

```
'C:\Program Files\Common Files\XtenderSolutions\CSW'
```

---

##### wwwroot

```
'C:\inetpub\wwwroot'
```

---

**File sharing**

'C:\data'

---

**Other required files**

'C:\setuptemp'

---