Commercial Compass Installation Guide

Abstract

This guide will walk the reader through installation and configuration of the Commercial Compass application. It is expected that the reader has already completed the requisite setup and configuration of the Windows Server, Sql Server, and Internet Information Services (IIS), as described in the previous section(s). Readers of the guide are expected to be comfortable with moving files on a file system and modifying text files.

Some steps in this guide require modifying files that use xml and json formats. This guide does not assume that the reader is familiar with these formats, and it provides instructions and examples to aid the reader to become familiar enough to make the required modifications.

Table of Contents

Code Extraction

1- Extract compressed application code

The reader should have received a compressed file (EHRLogic.zip) accompanying these instructions. This file contains all the application code required for the Commercial Compass, but it is stored in a compressed format and must be extracted before it can run.

To extract the code, double-click the compressed file (EHRLogic.zip) to open it. Once open, it should display its contents: a single folder called “EHRLogic”. Click and drag that folder into a temporary location, such as the desktop, to extract it. Once extracted, the code needs to be configured, as described in the following steps, before it is moved to the desired destination.

Configuration

Some manual changes are required in order for the Commercial Compass to work on the reader's system. These modifications will be made to files in the EHRLogic folder. For simplicity, all paths described in this section are relative to the EHRLogic folder. (For example, "\App_Data" is the folder "C:\EHRLogic\App_Data")

2- Configure connection strings to use the correct Sql Server

The application must be able to connect to the sql database in order to operate correctly. In this section, the reader will be guided through updating configuration files so that the application will use the correct database.

Open the folder "\WebConfig\prod". The folder should contain several files, including one called "connection.config". Open the connection.config file in any text editor.

The connection.config is a file containing xml formatted data (xml is short Extensible Markup Language, and is an industry standard for configurable data). The file is a list of connection strings which are used by the application (connection strings are an industry-standard format for describing the location and type of database). It should contain the following text:

```
<connectionStrings>
  <add name="Master" connectionString="Data Source=;Initial
Catalog=CommercialCompass;Integrated Security=true;"
providerName="System.Data.SqlClient" />
  <add name="elmah-logger" connectionString="Data Source=;Initial
Catalog=Logging;Integrated Security=true;"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

There are two connection strings, one called “Master” which is used for the bulk of the application behavior, and one called “elmah-logger” which is used to record any errors that occur in the application. The type information for these connection strings are already filled in, but the location is not.

Update the “Data Source” for the “Master” connection string with the name of the sql server that will be used by this application. Example below shows connecting to a sql server named “Example” (added text is highlighted):

```
<connectionStrings>
  <add name="Master" connectionString="Data Source=Example;Initial
Catalog=CommercialCompass;Integrated Security=true;"
providerName="System.Data.SqlClient" />
  <add name="elmah-logger" connectionString="Data Source=;Initial
Catalog=Logging;Integrated Security=true;"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

Next, update the “Data Source” for the “elmah-logger” so to the sql server where the application should record errors. (If desired, this can be the same sql server as the “Master”.) Example below shows using the same sql server, named “Example”, for both the Master and elmah-logger connection strings:

```
<connectionStrings>
  <add name="Master" connectionString="Data Source=Example;Initial
Catalog=CommercialCompass;Integrated Security=true;"
providerName="System.Data.SqlClient" />
  <add name="elmah-logger" connectionString="Data Source=Example;Initial
Catalog=Logging;Integrated Security=true;"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

The example below shows using a different sql server for Master (named “Example”) and for the elmah-logger (named “DifferentExample”).

```
<connectionStrings>
  <add name="Master" connectionString="Data Source=;Initial
Catalog=CommercialCompass;Integrated Security=true;"
providerName="System.Data.SqlClient" />
  <add name="elmah-logger" connectionString="Data
Source=DifferentExample;Initial Catalog=Logging;Integrated Security=true;"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

Choosing between using the same sql server or a different sql server is largely dependent on the preferences of the organization. Using the same sql server is generally less expensive, but it also means that, in the event of sql server issues, information about the issues may not be logged. Using a different sql server for error logging makes it possible to detect and record all errors, even if the “Master” database is unreachable, but owning and maintaining multiple sql servers generally requires a larger financial and time investment.

Once the desired changes are complete, save the file.

3- Configure the Hospital(s) that will be available in the application

Typical usage of the Commercial Compass application includes creating Assessments. Each Assessment is associated with a hospital. This section will guide the user through configuring the list of available hospitals.

Open the folder “\App_Data\SeedData\prod”. Inside that folder there are multiple files, including one called “Hospitals.json”. This is a file containing a Json formatted list of all the hospitals that will be available in the application (Json is short for Javascript Object Notation, and is an industry standard format for storing data so that it is easy for both humans and software to read.)

Open the “Hospitals.json” file in any text editor. It should contain the following text:

```
{
    "Hospitals": [

    ]
}
```

Update this file to include one or more Hospital objects. Hospital objects contain only two fields, a CommonName and a LegalName. The CommonName will appear in the application, and should be one that users can easily associate with the hospital. The LegalName should be the exact name of the hospital. For readers that are unfamiliar with Json, additional guidance is provided in the next two sections, “Using Json to Configure Hospitals” and “Examples”.

Using Json to Configure Hospitals

Inside this file, the word “Hospitals” appears in double-quotation marks ("), followed by a colon (:) then followed by an open square bracket ([ ). After that open square bracket ([) but before the closed square bracket (]), the list of hospitals must be supplied.

Each hospital has two required fields, a "CommonName" and a "LegalName". The CommonName will appear in the application, and should be one that users can easily associate with the hospital. The LegalName should be the exact name of the hospital. (For example, a hospital with the name “Example Regional Medical Center” might have “Example Regional” as the CommonName and “Example Regional Medical Center” as the LegalName.)

Json files have a small set of rules that define their syntax, and when adding a hospital the rules of the syntax must be followed:

1- Any object (in this case a hospital) must start with an opening curly brace ({) and end with a closing curly brace (})
2- Inside each object is a list of comma-separated pairs of key and value. The Key is surrounded by double quotation marks ("), and a colon (:) is used to separate the Key and the Value.
3- Values can be a variety of types, but the CommonName and LegalName of the hospital must be Strings. A string must both start and end with a double-quotation mark (").
4- If a value is an array, it must start with an open square bracket ([ ) and end with a close square bracket ( ]).
5- Entries in an array must be separated by commas (,) (in this case, multiple hospitals would be objects separated with a comma).
6- White space ( ) or tabs ( ) may be placed between any of the tokens. This is optional, and is primarily used to make the file more readable.
7- The file must begin with either an object or an array. (This rule is included for completeness. The Hospital.json file already begins with an object, so no action is required.)

Examples

Below is an example of how those rules are used for the example provided earlier ("Example Regional"):

```
{
    "Hospitals": [
        {
            "CommonName": "Example Region",
            "LegalName": "Example Regional Medical Center"
        }
    ]
}
```

Below is an example of how the rules might be used for an example health system that wants two hospitals to be available:

```
{
    "Hospitals": [
        {
            "CommonName": "ESU",
            "LegalName": "Example State University Hospital",
        },
        {
            "CommonName": "EGMC",
            "LegalName": "Example General Medical Center",
        }
    ]
}
```

Once the desired changes are complete, save the file.

4- Configure which users will have access to the application

Users of the Commercial Compass application must be granted permission to use the application. This can be setup by modifying a Json file, similar to how the Hospital was setup.

Warning: The Json file is only for initial setup of users- once users are added the application will ignore the file.

Open the folder “\App_Data\SeedData\prod”. Inside that folder there are multiple files, including one called “InitialPermissions.json”. This is a file containing a Json formatted list of all the users that will have access to the application, and what permissions those users have (Json is short for Javascript Object Notation, and is an industry standard format for storing data so that it is easy for both humans and software to read.)

Open “InitialPermissions.json” in any text editor. It should contain the following text:

```
{
    "Users": [

    ]
}
```

This file contains an empty list of “Users”, which is represented as the key “Users” followed by a colon (:), then an open square braket ([ ), and then a close square bracket ( ]). Following the syntax rules of Json (as documented in “Using Json to Configure Hospitals”), one or more users may be added.

User objects have 4 required fields:

1- FirstName – The first name of the user
2- LastName – The last name of the user. The First and Last name are displayed together in the application to identify the user (for example, when displaying who completed a specific assessment)
3- UserName – The name of the user in the Active Directory system (the user's Windows login name). It must contain only the user name, and not the domain name (more on this later).
4- Roles – A list of strings. The string "Compass User" gives the user the role to access the application. The string "ElmahAccess" gives the user the role to read error logs. A user may be assigned one or both of these roles.

UserName

The Commercial Compass application integrates with Windows Active Directory. The UserName is key to accomplishing this: when a user accesses the application, the application will confirm that the username provided by Active Directory matches the UserName of one of the user permissions. For example, if a user in the domain "example.com" has the name "jdoe", their entry in the InitialPermissions.json should be "UserName": "jdoe".

It is important to note that multiple domains are not supported.

Roles

There are two roles available for users, "Compass User" and "ElmahAccess".

"Compass User" should be given to any user that will interact with the application (including, but not limited to, creating assessments, reviewing existing assessments, or completing assessments). Personally identifiable information (PII) or Protected Health Information (PHI) may be stored in assessments, so "Compass User" should only be granted to users that should be able to access that information.

The "ElmahAccess" role will grant that user access to the elmah.axd page, which contains all the application error logs. Error logs may contain sensitive system information, such as machine names or Internet Protocol (IP) addresses, so the "ElmahAccess" role should only be granted to users that should be able to access that information.

Examples

A series of examples are provided below (Like all examples, all names are fictitious and any resemblance to actual persons is entirely coincidental):

A single user (“John Doe”) that can access Commercial Compass, and no users that can access the error logs.

```
{
    "Users": [
        {
            "FirstName": "John",
            "LastName": "Doe",
            "UserName": "jdoe",
            "Roles": [
                "Compass User"
            ]
        }
    ]
}
```

Two Users, one that can access Commercial Compass (“John Doe”) and one that provides IT support by accessing error logs (“Jane Smith”):

```
{
    "Users": [
        {
            "FirstName": "John",
            "LastName": "Doe",
            "UserName": "jdoe",
            "Roles": [
                "Compass User"
            ]
        },
        {
            "FirstName": "Jane",
            "LastName": "Smith",
            "UserName": "jsmith",
            "Roles": [
                "ElmahAccess"
            ]
        }
    ]
}
```

Two Users, both of whom can only access Commercial Compass:

```
{
    "Users": [
        {
            "FirstName": "John",
            "LastName": "Doe",
            "UserName": "jdoe",
            "Roles": [
                "Compass User"
            ]
        },
        {
            "FirstName": "Jane",
            "LastName": "Baker",
            "UserName": "jbaker",
            "Roles": [
                "Compass User"
            ]
        }
    ]
}
```

One User who can access both Commercial Compass and the error logs:

```
{
    "Users": [
        {
            "FirstName": "John",
            "LastName": "Adams",
            "UserName": "jadams",
            "Roles": [
                "Compass User",
                "ElmahAccess"
            ]
        }
    ]
}
```

While the above examples do not exhaustively cover all possible combinations (it is not possible to do so as there is no limit to the number of users,) they are provided with the hope that they are informative regardless of the reader's actual user setup needs.

5- Move the application code to the desired location

Now that the code is configured, it needs to be moved to the location where Internet Information Services (IIS) is able to access it. Open a window to display the "C:\" location. Click and drag the folder extracted in the previous step, and drop it into "C:\" so that the folder path is "C:\EHRLogic" (this will match the configuration in IIS).

Verify the application configuration

Once all configuration changes are completed, it is important to verify that the application is setup as desired. A user with the "Compass User" role is required in order to completely verify the application

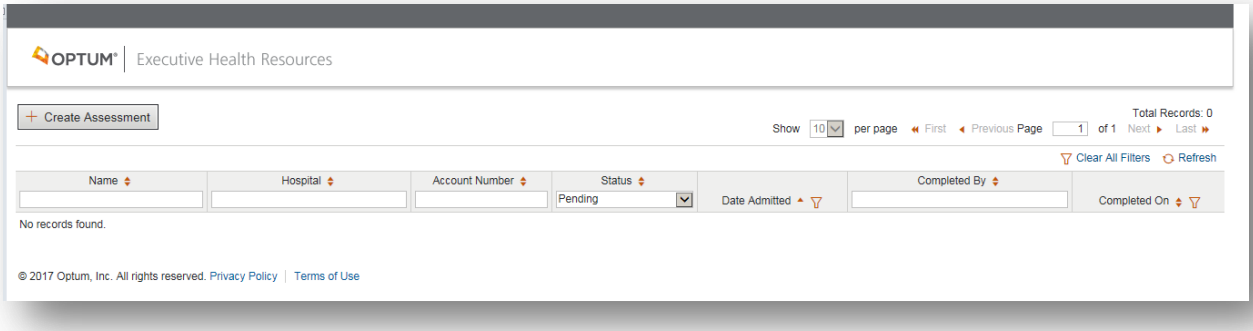
configuration. In case it is not possible, partial verification may be performed by any user (even one without access to the application).

Before verification begins, be sure to save and close any files that were opened during the configuration process. Also, be sure to have the url of the site configured in Internet Information Services (IIS) handy, as it will be required in the following steps.

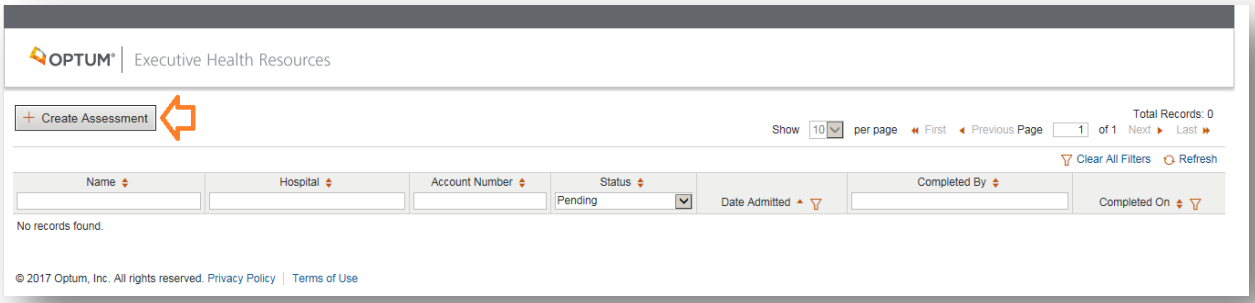
6- Verify Configuration

A User with “Compass User” role should navigate a web browser (such as Internet Explorer) to the url for the Commercial Compass site. This will cause the Commercial Compass application to initialize itself, including (but not limited to) creating the necessary data structures in the database and importing user permissions. This may take a few minutes, as numerous changes must be made (this is the only time that the application should take a while to start. Once the database is configured, future startup times should be much faster.)

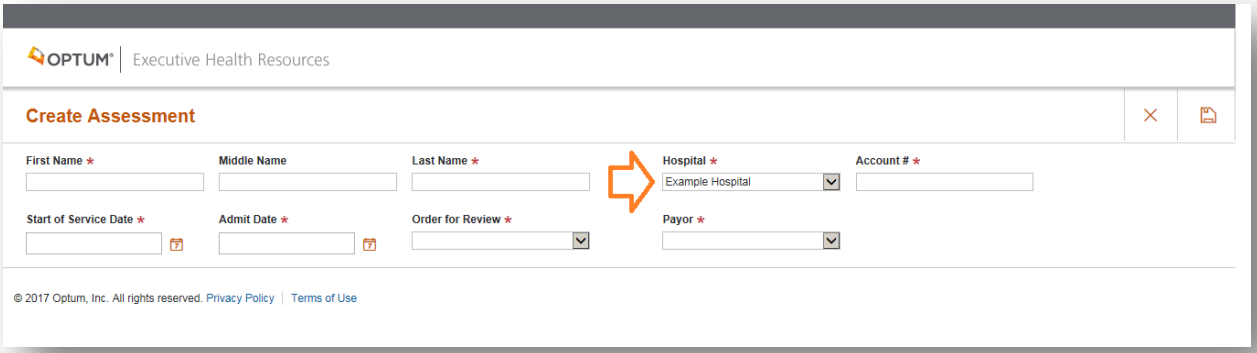
Once the application completes loading, the web browser will display the home page of Commercial Compass. (If the above steps are performed by someone who is not a user with “Compass User” role, they will be unauthorized and denied access. This constitutes partial verification, as it only verifies that the application can connect to the database.)



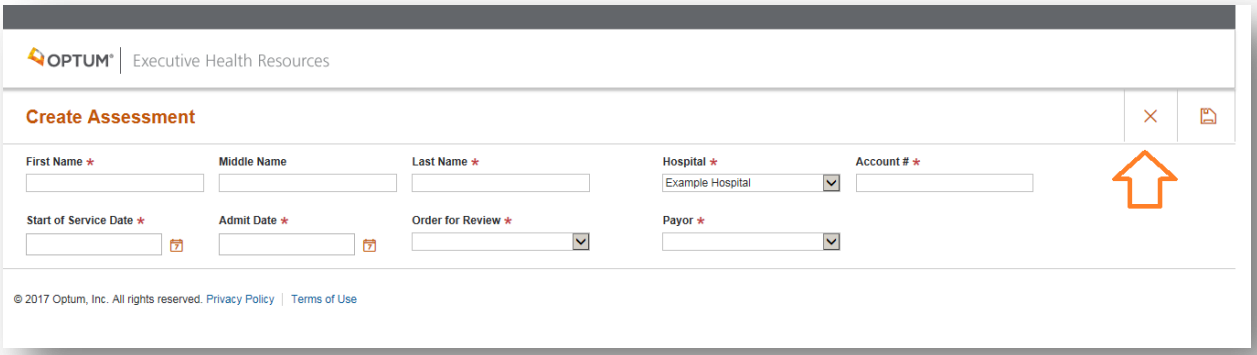
On the home page, the user should open the form to create a new assessment, by clicking on the "Create Assessment" button (indicated by the arrow below). Please note: the creation process will be cancelled without saving, so an assessment will not actually be created.



Once the form is open, observe the hospital dropdown. If only one hospital is configured, it should be selected. If more than one hospital is configured, click on the dropdown to open it, and verify that all the desired hospitals appear.



Once verified, cancel creating the assessment by clicking on the X icon in the top-right.



Close the browser window, all verifications are now complete.

Installation successful

The Commercial Compass is now successfully installed and configured.