

Using Environment Properties

<https://cms.pilotfishtechnology.com/how-to-use-environment-properties-tutorial-pilotfish>

eiConsole v.24R1
Tutorial & Working Directory

eiConsole Tutorial - Topology

Using Environment Properties

Note: The Quick Start and Foundation tutorials should be completed before proceeding with the Topology Tutorials.

Overview

In this tutorial, we'll cover the uses of environment properties, how to configure them, and what they are used for. In addition, this tutorial expands on the eiConsole Basics tutorials, such as configuring [Listeners](#) and [Transports](#), as well as testing.

Before You Begin



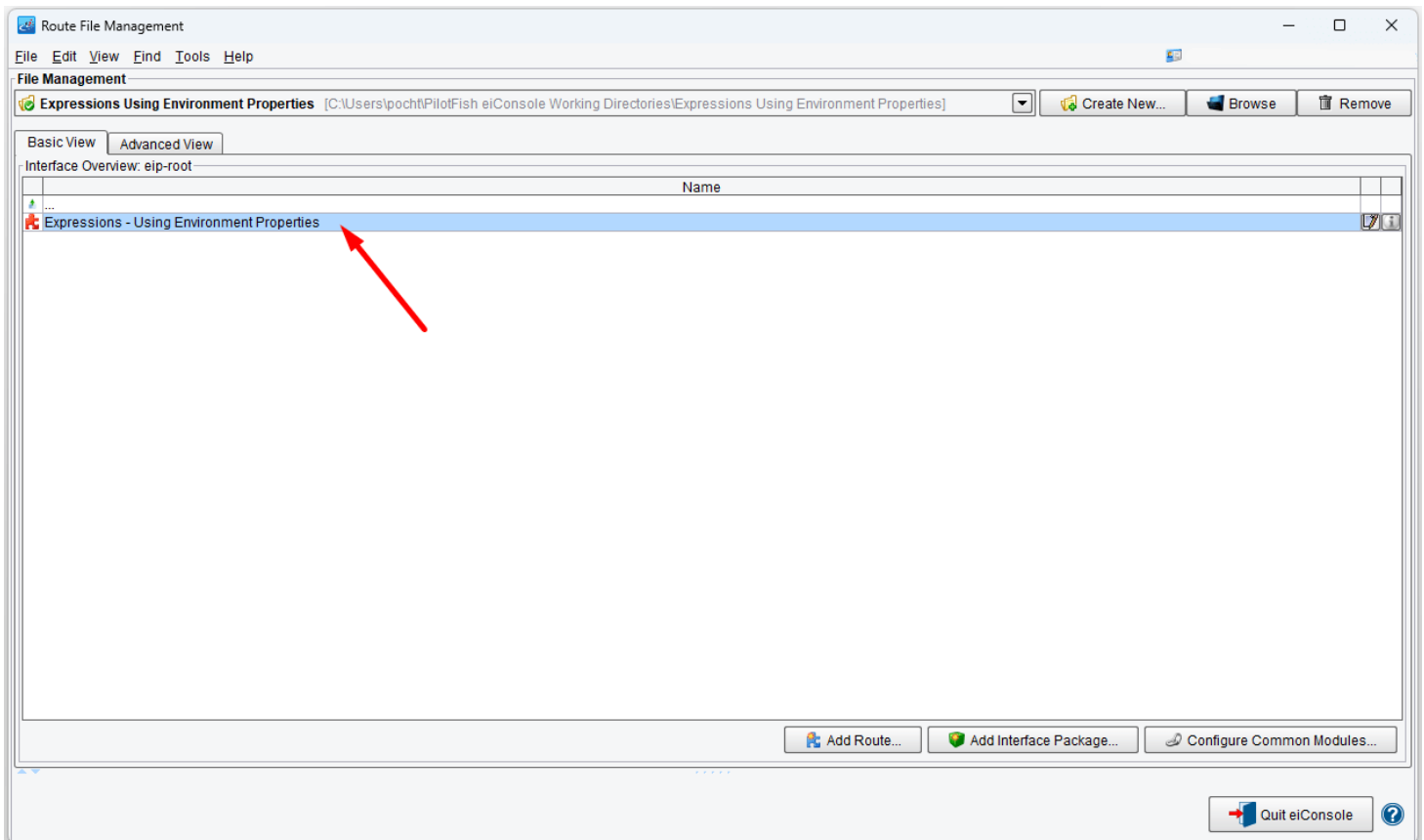
Using Environment Properties
Working Directory

Download the **Using-Environment-Properties-Working-Directory.zip** file with the sample Working Directory and unzip it on your computer to a convenient place. In our case, it's `c:\Users\{USER_NAME}\PilotFish eiConsole Working Directories\Expressions Using Environment Properties` where `{USER_NAME}` is the user's name.

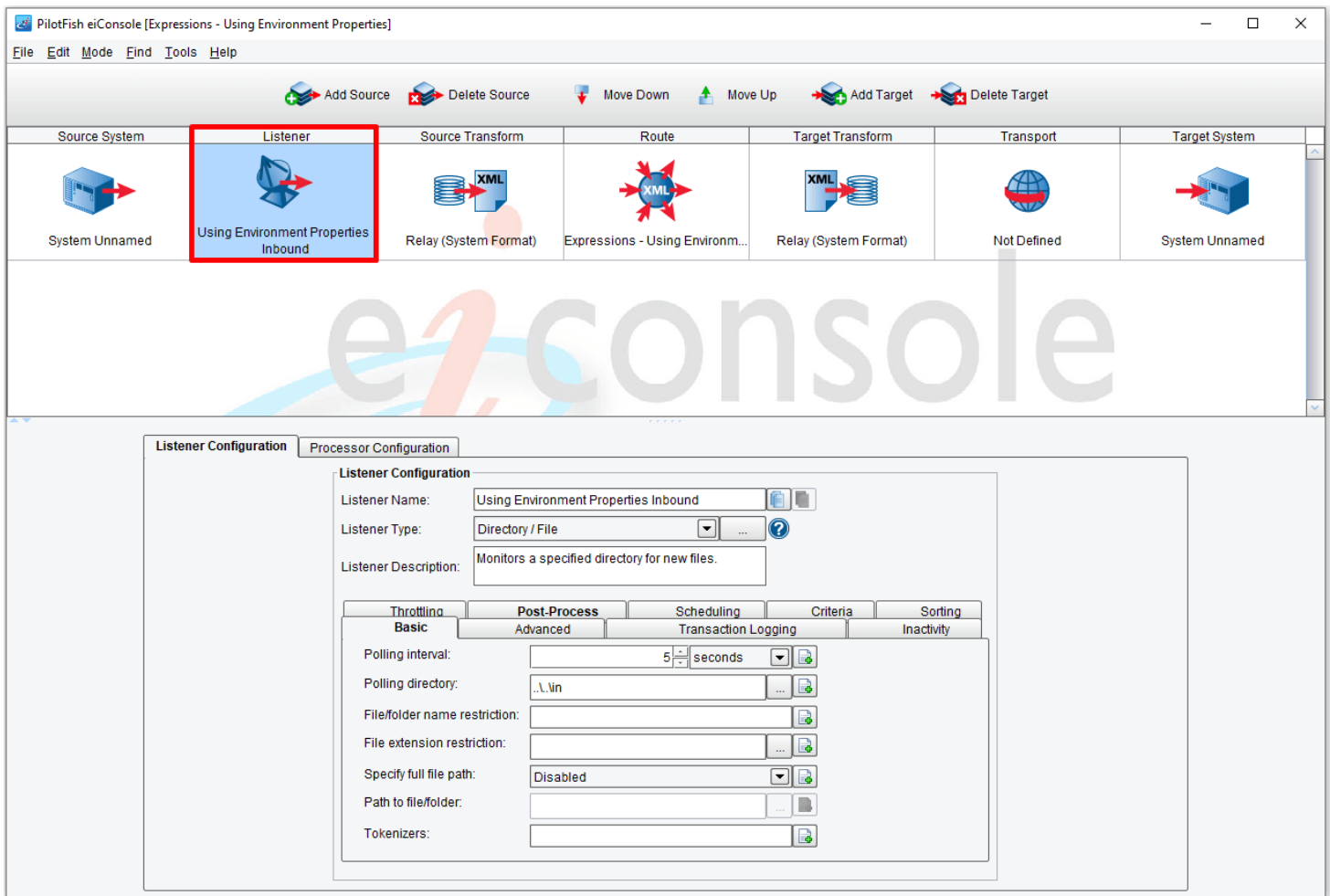
Open the eiConsole, locate your Simple Route directory, and open it. The **Expressions – Using Environment Properties** route is included in the [working directory](#). Your Route File Management screen will open, as shown below.

Next, follow the tutorial and walk through it step-by-step.

Click on the **Expressions – Using Environment Properties** route.



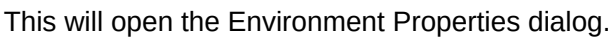
Most modules in the eiConsole, such as Listeners, [Processors](#), Transports, and so forth, make use of configuration items, which, so far, we've primarily configured as static, constant values. For example, we'll use a Directory/File Listener's configuration:



In this case, we've typed the value for the **Polling directory** item. It's currently set to "`..\..\lin`." We've also set the **Polling interval** to check the specified directory for new files every 5 seconds. This approach has been quite workable for building our [interfaces](#) and testing them. However, suppose we need to move this interface from our workstation to another or, more importantly, to a testing or development environment. As is, we'd need to load up the route in an eiConsole on that machine and make the appropriate changes. Then, when we move the interface from the testing machine to a staging environment or even to production, we would once again need to edit the route.

Now, suppose our interface is made up of many [routes](#), each with multiple Listeners and Transports. Some might be interacting with folders, meaning we'd need to change the paths each time. Others might be querying and updating databases, meaning we'd need to change connectivity information, including usernames, passwords, connection strings, driver names, etc., each time. The problem of moving interfaces from environment to environment quickly scales up as the complexity of an interface grows.

Fortunately, there's a pretty simple, straightforward solution: Environment Properties. Environment properties are property values defined in an external file that exists, relative to an interface, at a fixed location (the root directory). We can edit and use the contents of this file from within the eiConsole by going to the Edit menu and selecting **Environment Properties**.



Here we have the contents of the Environment Properties file shown in a table. The "key" is the name of the property and the "value" is its current value. Let's try defining our source directory and polling interval here by adding two rows: the first key is **source_directory**, and its value is `..\..lin`; the second key is **polling_interval**, and its value is **5**.

Environment Properties Dialog

Search:

Search

<

>

Clear Search

Key	Value	Ancestor Value
source_directory	..\\..\\	
polling_interval	5	

Ancestor Key	Ancestor Value	Ancestor Name
--------------	----------------	---------------

☐ Encrypt environment properties

Save

Merge to Root

Add File Path

Add Folder Path

Remove

Reload Environment Properties

Close

We could add as many keys and values as we wished, so long as each key has a unique name. It's generally a good idea to qualify keys by intent or through some namespace. We can then click the **Save** button to save our changes.

Environment Properties Dialog

Search:

Key	Value	Ancestor Value
source_directory	..\\..\\	
polling_interval	5	

Ancestor Key	Ancestor Value	Ancestor Name
--------------	----------------	---------------

☐ Encrypt environment properties

After saving, click **Close** to close the dialog. Returning to our Directory/File Listener's configuration, we'll note a button visible to the right of each configuration item with a document+ icon on it.

Listener Configuration

Listener Name:  

Listener Type:   

Listener Description:

Throttling	Post-Process	Scheduling	Criteria	Sorting
Basic	Advanced	Transaction Logging		Inactivity

Polling interval:   seconds  

Polling directory:  

File/folder name restriction: 

File extension restriction:  

Specify full file path:  

Path to file/folder:  

Tokenizers: 

This button enables **Enhanced Configuration Values**, meaning it provides access to various dynamically evaluated expressions, including our defined environment properties. Click the button and it should be replaced by three more.

Listener Configuration

Listener Name:  

Listener Type:   

Listener Description:

Throttling	Post-Process	Scheduling	Criteria	Sorting
Basic	Advanced	Transaction Logging		Inactivity

Polling interval:   seconds  

Polling directory:   

File/folder name restriction: 

File extension restriction:  

Specify full file path:  

Path to file/folder:  

Tokenizers: 

The first button allows us to select from defined environment properties. The second will enable us to construct

OGNL expressions ([covered in another tutorial](#)). Finally, the third returns us to "standard" configuration values. Click the first button and it will open a dialog showing our defined properties:

Environment Properties Dialog

Search:

Search

<

>

Clear Search

Key	Value	Ancestor Value
polling_interval	5	
source_directory	..\\in	

Ancestor Key	Ancestor Value	Ancestor Name
--------------	----------------	---------------

Edit Mode

Select

Reload Environment Properties

Close

For this item, we'll select "source_directory." Select it in the table, then click the **Select** button:

Environment Properties Dialog

Search:

Key	Value	Ancestor Value
polling_interval	5	
source_directory	..\\..\\	

Ancestor Key	Ancestor Value	Ancestor Name
--------------	----------------	---------------

Our configuration item's value now looks like this:

Listener Configuration

Listener Name:  

Listener Type:  

Listener Description:

Throttling	Post-Process	Scheduling	Criteria	Sorting
Basic	Advanced	Transaction Logging		Inactivity

Polling interval:   seconds 

Polling directory:   

File/folder name restriction: 

File extension restriction: 

Specify full file path:  

Path to file/folder: 

Tokenizers: 

The "\$\$source_directory" text tells us that we're looking for an environment property named "source_directory." The "=..\.in" simply tells us the current value of that property, as understood by the eiConsole.

Repeat the exercise for the Polling Interval item:

Listener Configuration

Listener Name:  

Listener Type:  

Listener Description:

Throttling	Post-Process	Scheduling	Criteria	Sorting
Basic	Advanced	Transaction Logging		Inactivity

Polling interval:   

Polling directory:   

File/folder name restriction: 

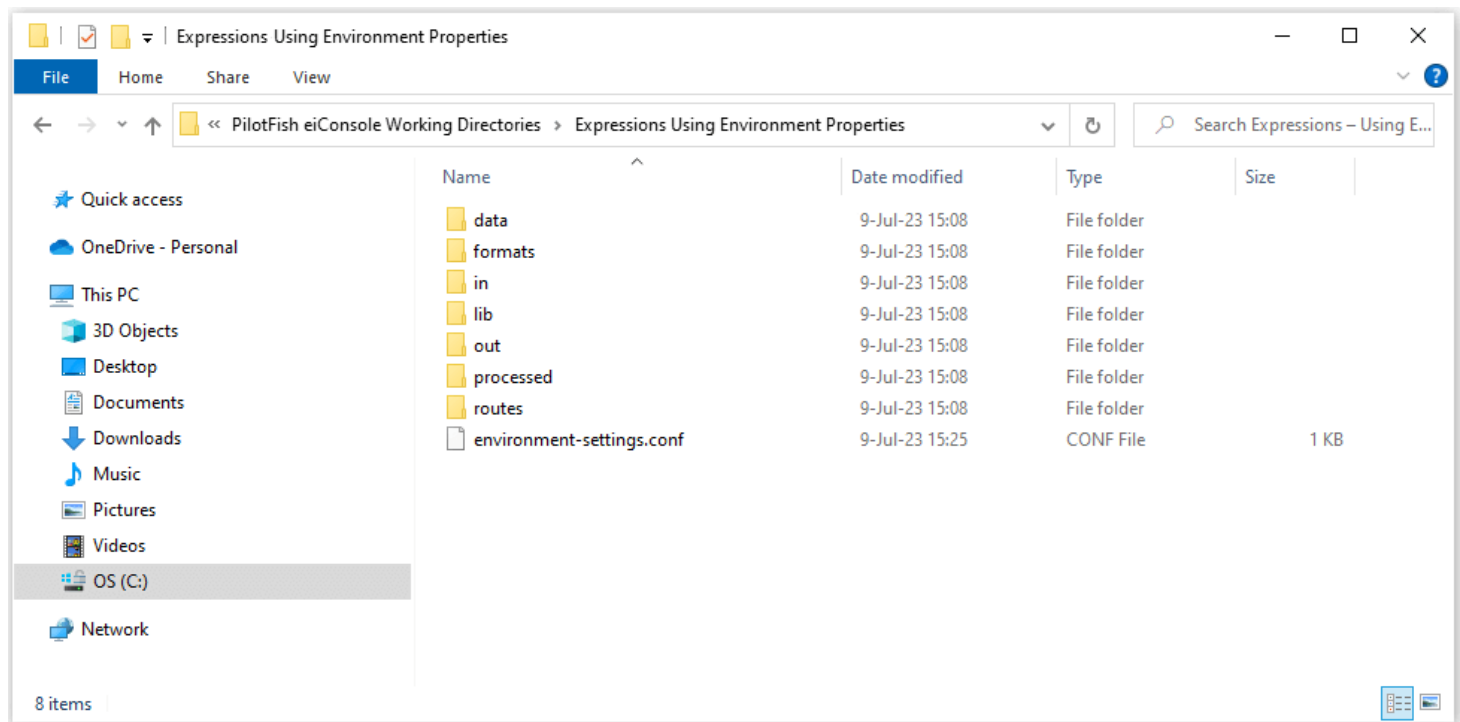
File extension restriction: 

Specify full file path:  

Path to file/folder: 

Tokenizers: 

Now, whenever we move this Route between systems, we can either make our changes in the eiConsole Environment Properties dialog or edit the file directly. If we examine our working directory, we'll see it now looks like this:



We now have a file named "environment-settings.conf" - this is the external file. You can open it in any standard text editor; do so now and you should see the following:

```
1 #eiPlatform environment properties
2 polling_interval=5
3 source_directory=..\..\in
4
```

When we move our interface between systems, we only need to update this single file. If the interface exists already on a particular system then our deployment is simply a matter of not copying over it.

Using Environment Properties in the eiConsole module configuration items allows for externally defined values for Listeners and Transports which you can now use. Next, move on to more advanced topics like [Expressions - Using OGNL](#).