SMA TECHNOLOGIES

OPCON

EASE Connector

Published February 2019

Table of Contents

Initial Setup

Install Scripts

SMA Technologies will provide the following OpCon API powershell scripts as part of the EASE Connector. Additional documentation on each script can be found in this document.

- EASE-LETTERFILE-FTP.ps1
- EASE-LETTERFILE-RENAME.ps1
- EASE-MONITOR.ps1
- EASE-PROMPT.ps1
- EASE-RESET.ps1
- EASE-RSJ.ps1
- EASE-RSJEDIT.ps1
- EASE-SEQ-FTP.ps1
- EASE-SEQ-PROMPT.ps1

Jack Henry Associates (JHA) Connection Setup

Introduction

JHA will provide the connection credentials for both their OpCon API and FTP credentials to their DMZ. Both need to be established for the EASE Connector to be effective.

The FTP credentials will need to be loaded into a Command Line driven FTP utility. SMA Technologies recommends WSFTP Pro because it is inexpensive and provides all the functionality we need, but any commend line drive FTP solution will work.

The OpCon API credentials will need to be loaded into OpCon. They are listed as "Static Parameters" within our script because they will very rarely change. There will be four components to the OpCon API connection. These components are listed below:

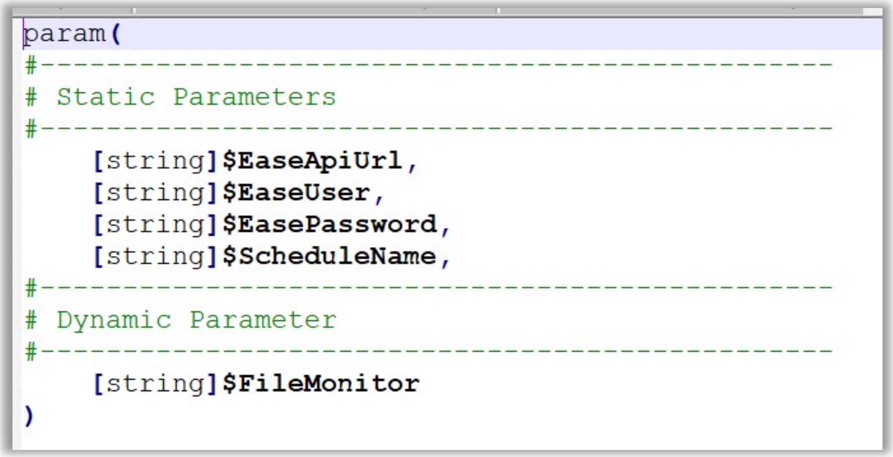
1. The OpCon API URL.
2. An OpCon User Account the API will use to connect to EASE's environment.
3. The password for the OpCon User Account.
4. The Schedule you will access within EASE's environment through the EASE Connector.

Connection - Global Property

It is required to store the OpCon API connection credentials into an encrypted Global Property within your OpCon environment. This will do a few things for you:

- Store the credentials in a centralized location.
- Simplify your command lines combining 4 parameters into one Global Property
- Keep the credentials secure by encrypting the Global Property value.

At the top of each script you will see a section listing all Static Parameters as well as Dynamic Parameters. The Dynamic parameters will be entered on the command line individually. The Static will be added by including the Global Property. Below is an example of the parameters listed at the top of the script.



```
param(
#-------------------------------------------------
# Static Parameters
#-------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#-------------------------------------------------
# Dynamic Parameter
#-------------------------------------------------
    [string]$FileMonitor
)
```

Example

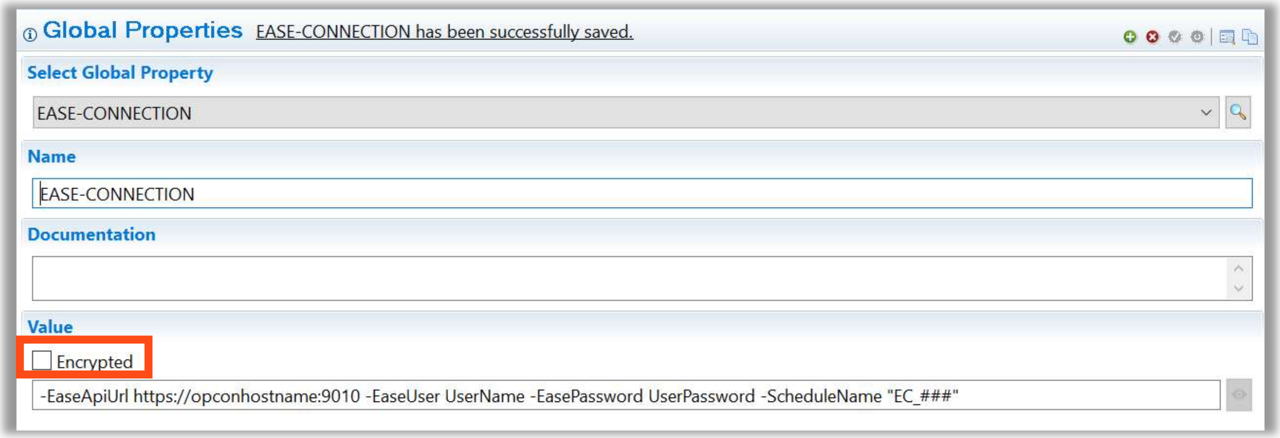
Let’s pretend the connection credentials are the following.

- **URL** = “**https://opconhostname:9010**”
- **User Name** = “**EC_999**”
- **Password** = “**examplepassword**”
- **Schedule Name** = “**EC_999**”

Then the four command line parameters would be:

- -EaseApiUrl “https://opconhostname:9010”
- -EaseUser = “EC_999”
- -EasePassword = “examplepassword”
- -ScheduleName = “EC_999”

The Value of the Global Property will look like the following *BEFORE* it is encrypted. Notice we list the command line switch followed by the appropriate value and then the next command line switch until we complete all four.



Global Properties EASE-CONNECTION has been successfully saved.

Select Global Property

EASE-CONNECTION

Name

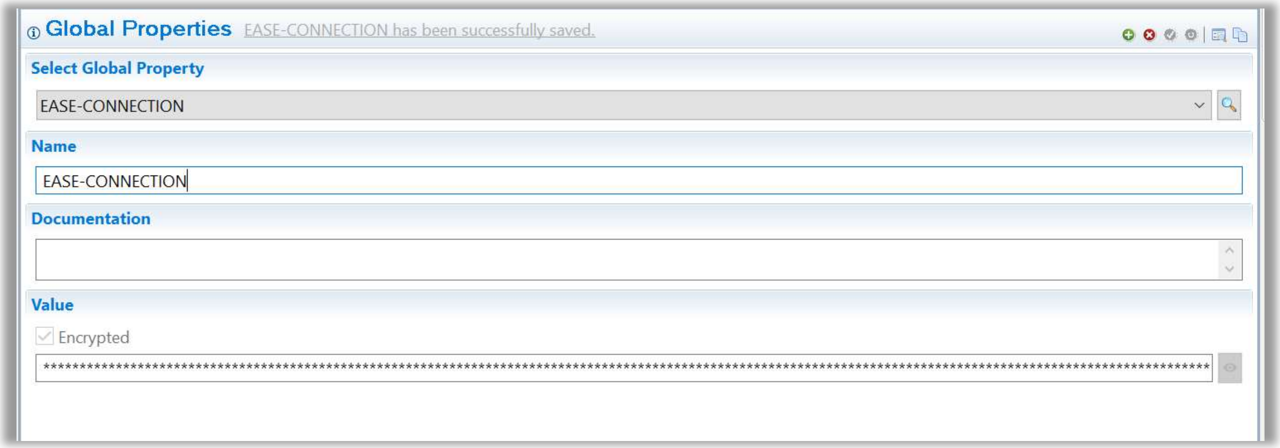
EASE-CONNECTION

Documentation

Value

□ Encrypted

-EaseApiUrl https://opconhostname:9010 -EaseUser UserName -EasePassword UserPassword -ScheduleName "EC_###"



Global Properties EASE-CONNECTION has been successfully saved.

Select Global Property

EASE-CONNECTION

Name

EASE-CONNECTION

Documentation

Value

☑ Encrypted

****

EASE Connector Concepts

Uploading an Edit File

Anytime you need to load a file into Episys it will consist of at least three Jobs.

1. An **FTP Job** to upload the file from your network to JHA’s DMZ.
    a. WSFTP Pro or another command line driven FTP utility.
2. A **File Monitor Job** running on Episys to detect the arrival or a file.
    a. EASE-MONITOR.ps1
3. The **RSJ Job** running the Episys Edit Job.
    a. EASE-RSJEDIT.ps1

The FTP Job transfers the file from your environment to JHA’s DMZ. They will automatically relay the file from their DMZ to the Letter Files directory within your production SYM on Episys.

The File Monitor Job is necessary because that relay from JHA’s DMZ to Episys is not immediate. After we successfully transfer the file to the DMZ we need the File Monitor to verify the file has arrived before moving forward.

If there are no prompts to answer, you can then move forward with running the Edit Job. You will run

Transfer a Report off Symitar

When transferring a report off Symitar four Jobs are required.

1. An EASE Connector sequence number lookup.
2. An EASE Connector Job to copy the report into Letter Files.
    a. This is where the FTP Jobs are expecting the files.
3. An EASE Connector FTP Job transferring the file to JHA’s DMZ.
4. A local FTP Job pulling the file from JHA’s DMZ to your local system.

Three of those four Jobs are consolidated into one EASE Connector script (**EASE-SEQ-FTP.ps1**).

Transfer a Letter File off Symitar

When transferring a file off Symitar three Jobs are required.

1. An EASE Connector Job to either copy or rename the Letter File appending the prefix necessary to transfer the file to JHA’s DMZ.
    a. Use the Move if the Episys requires the file to be cleared.
    b. Use a Copy if the file is overwritten by the Episys Job creating the file.
2. An EASE Connector FTP Job transferring the file to JHA’s DMZ.
3. A local FTP Job pulling the file from JHA’s DMZ to your local system.

Two of those three Jobs are consolidated into one EASE Connector script (**EASE-LETTERFILE-FTP.ps1**).

EASE Connector Scripts

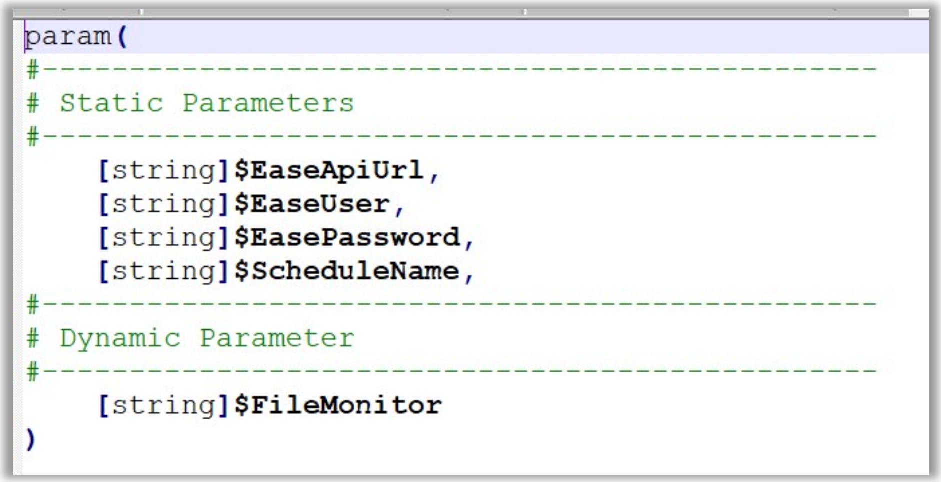
File Monitor

The **EASE-MONITOR.ps1** script uses the OpCon API to add a **File Monitor Job** within EASE's OpCon environment. This Job monitors the Episys Letter Files directory for the file you uploaded to the JHA DMZ. JHA automatically passes the file from their DMZ to the Letter Files directory. Once the file arrives the script will finish successfully and report back to your OpCon system. This will let your OpCon system know when it is ok to continue with processing.

Note: A prefix will be added to the file name when it is relayed to Episys. Ignore that prefix in the file name specified in your Job.

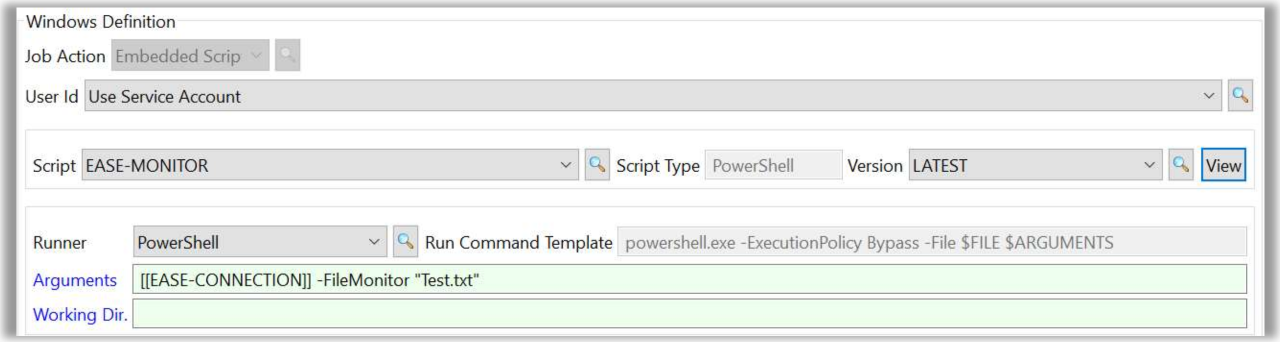
Example: File **test.txt** becomes **EC999_test.txt**. The **-FileMonitor** parameter on your Command Line should look like "**test.txt**" and **NOT** "**EC999_test.txt**".

This Job only has one Dynamic Parameter.



```
param(
#------------------------------------------------
#  Static Parameters
#------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#------------------------------------------------
#  Dynamic Parameter
#------------------------------------------------
    [string]$FileMonitor
)
```

The Job will consist of the EASE-CONNECTION Global Property followed by the -FileMonitor parameter. This parameter will consist of the name of the file you uploaded.



Windows Definition

Job Action: Embedded Scrip

User Id: Use Service Account

Script: EASE-MONITOR

Script Type: PowerShell

Version: LATEST

Runner: PowerShell

Run Command Template: powershell.exe -ExecutionPolicy Bypass -File $FILE $ARGUMENTS

Arguments: [[EASE-CONNECTION]] -FileMonitor "Test.txt"

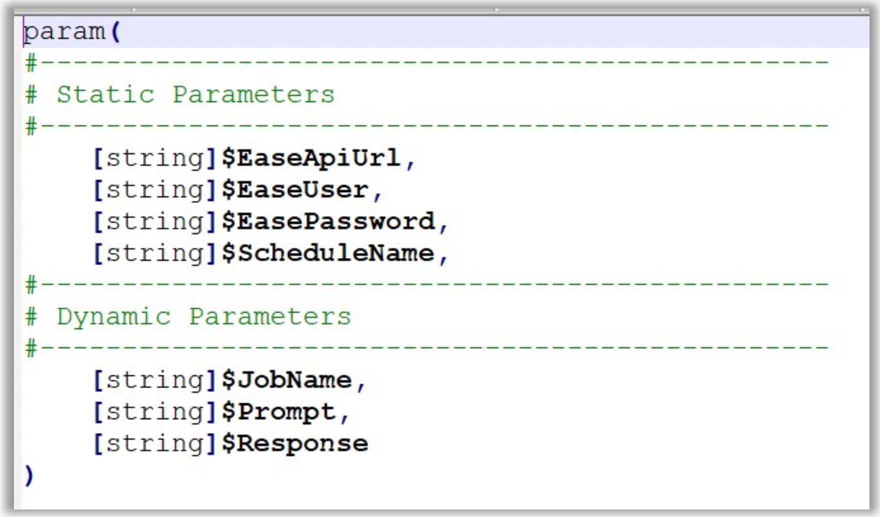
Working Dir.:

Answer a Prompt

The **EASE-PROMPT.ps1** script uses the OpCon API to add a Job updating an Episys prompt within EASE's OpCon environment. This script will be used any time you want to run an Episys Job which has a prompt (or prompts) that need to be answered. A few things to consider with this EASE Connector script are:

1. You can only answer one prompt each time the Job is run. The script passes a single prompt and single response. If the same Episys Job has multiple prompts we need to break this up into multiple EASE Connector Jobs.
2. We have a different form of the script if the answer to the prompt is the sequence number of a report. For more information on this, read about the **EASE-SEQ-PROMPT.ps1** script.

The parameters within this script are:

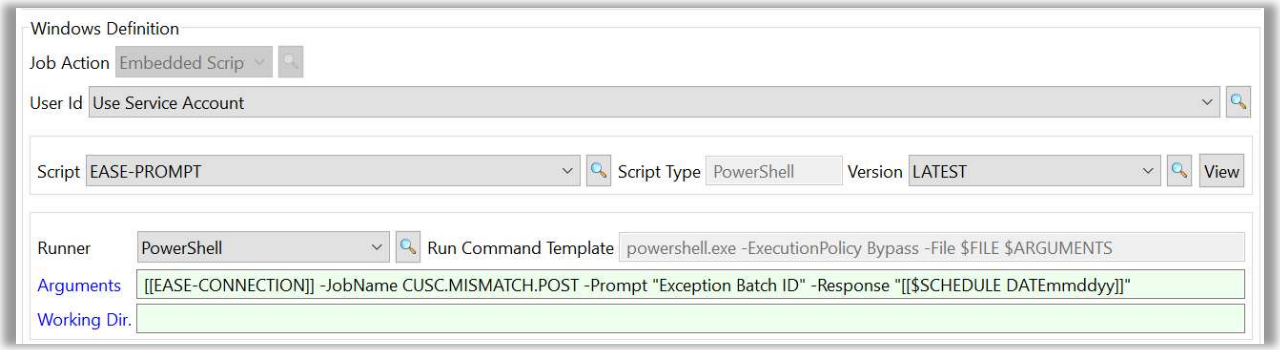


```
param(
#------------------------------------------------
# Static Parameters
#------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#------------------------------------------------
# Dynamic Parameters
#------------------------------------------------
    [string]$JobName,
    [string]$Prompt,
    [string]$Response
)
```

You may notice that the Static Parameters are the same as the previous script, but the Dynamic Parameters are different. This will be true for all the scripts.

This script has three Dynamic Parameters:

- **-JobName**
  The name of the nested Episys Job which contains the prompt. In the case of a large batch Job containing many smaller Jobs, we need to specify the smaller Job which actually contains the prompt.
- **-Prompt**
  The prompt within the job. This needs to be the actual string of text.
- **-Response**
  This parameter will contain the desired response to that prompt.



Windows Definition

Job Action: Embedded Scrip

User Id: Use Service Account

Script: EASE-PROMPT

Script Type: PowerShell

Version: LATEST

Runner: PowerShell

Run Command Template: powershell.exe -ExecutionPolicy Bypass -File $FILE $ARGUMENTS

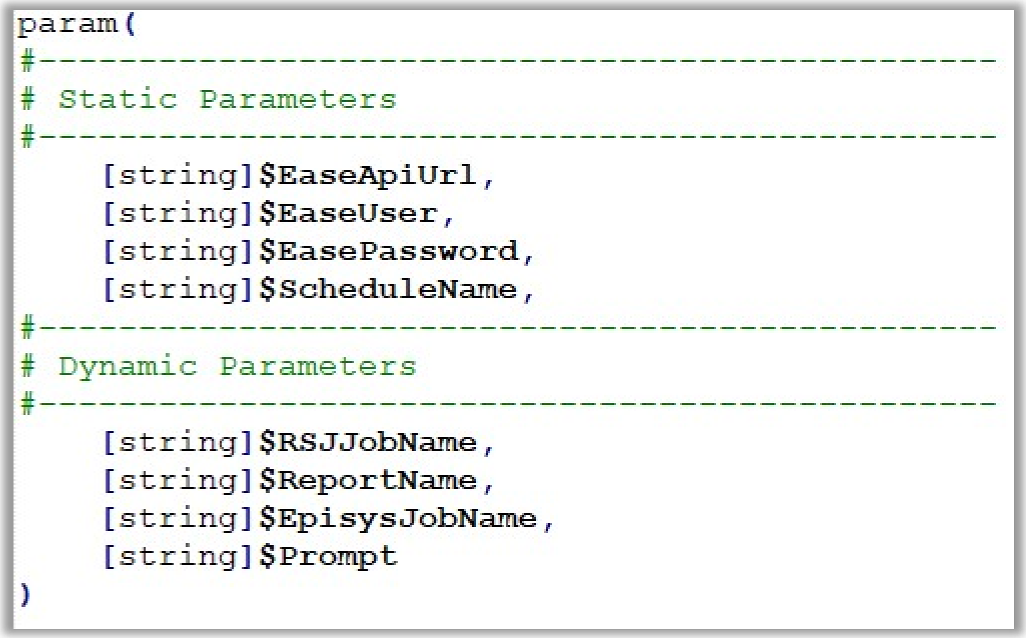
Arguments: [[EASE-CONNECTION]] -JobName CUSC.MISMATCH.POST -Prompt "Exception Batch ID" -Response "[[$SCHEDULE DATEmmddyy]]"

Working Dir.

Answer a Prompt with a Sequence Number

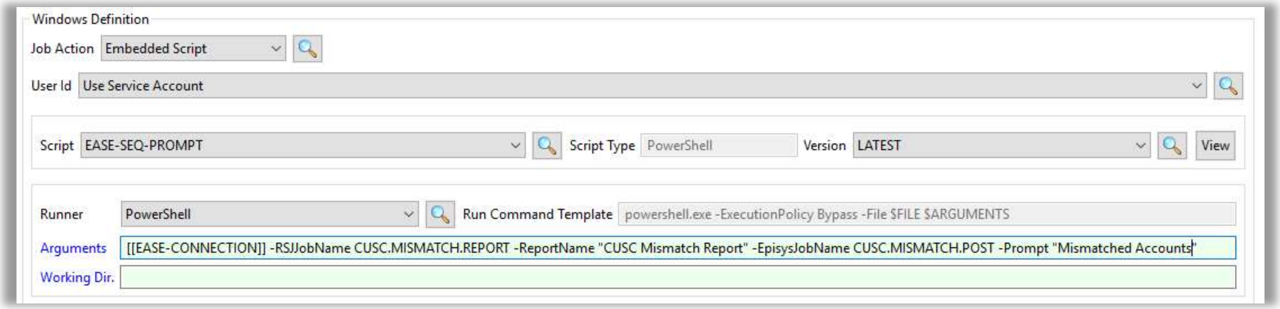
The **EASE-SEQ-PROMPT.ps1** script uses the OpCon API to add two Jobs to EASE's environment. This means one Job within your environment will control two Jobs in the EASE environment. The two Jobs do the following:

1. The first Job searches for the sequence number of the defined report and stores it in a Schedule Instance Property within the EASE Schedule.
    a. **Note**: All sequence number Jobs update the same Schedule Instance property. Because of this you need to make sure only one of these Jobs run at a time. OpCon Resource Dependences are recommended.
2. The second Job answers the specified prompt in the specified Episys Job with the sequence number collected during the first job.



```
param(
#-------------------------------------------------
#  Static Parameters
#-------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#-------------------------------------------------
#  Dynamic Parameters
#-------------------------------------------------
    [string]$RSJJobName,
    [string]$ReportName,
    [string]$EpisysJobName,
    [string]$Prompt
)
```

Notice that the only difference between the EASE-PROMPT and EASE-PROMPT-SEQ parameters is we dropped the $Answer from the EASE-PROMPT-SEQ script.



Windows Definition

Job Action: Embedded Script

User Id: Use Service Account

Script: EASE-SEQ-PROMPT | Script Type: PowerShell | Version: LATEST

Runner: PowerShell | Run Command Template: powershell.exe -ExecutionPolicy Bypass -File $FILE $ARGUMENTS

Arguments: [[EASE-CONNECTION]] -RSJJobName CUSC.MISMATCH.REPORT -ReportName "CUSC Mismatch Report" -EpisysJobName CUSC.MISMATCH.POST -Prompt "Mismatched Accounts"

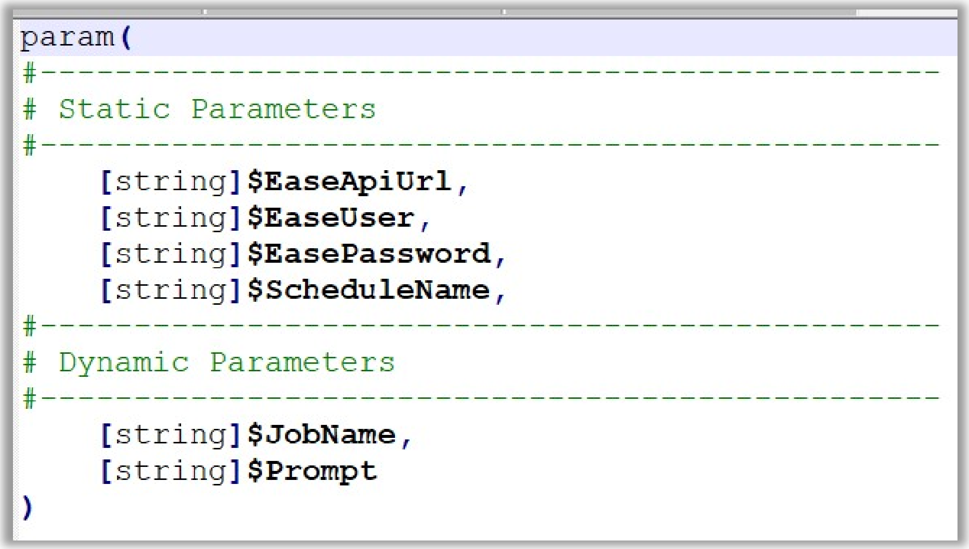
Working Dir.:

This script has four Dynamic Parameters:

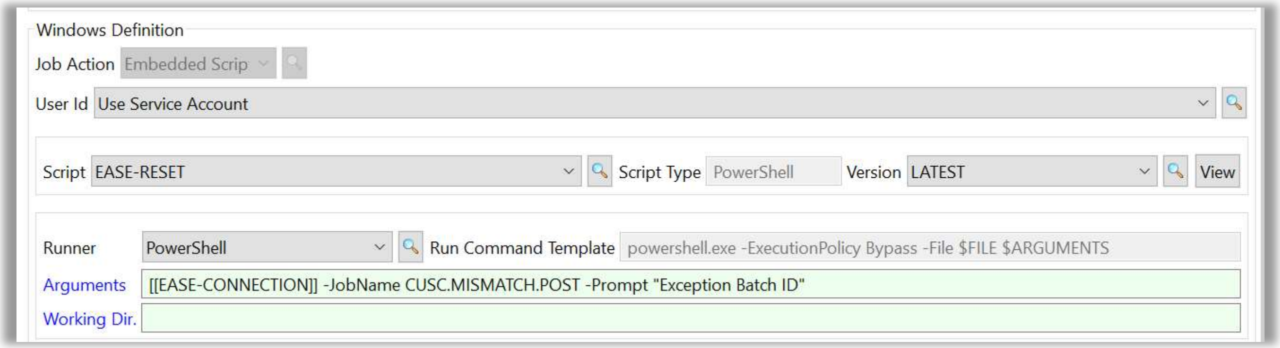
- **-RSJJobName**
  The name of the Episys Job which was run to create the report whose sequence number we are searching for.
- -**ReportName**
  The title of the report whose sequence number you need to collect. This will be used as the answer to the -Prompt listed below.
- **-EpisysJobName**
  The name of the nested Episys Job which contains the prompt you are answering with the sequence number collected. In the case of a large batch Job containing many smaller Jobs, we need to specify the smaller Job which contains the prompt.
- **-Prompt**
  The prompt within the job. This needs to be the actual string of text.

Reset a Prompt

The **EASE-RESET.ps1** script uses the OpCon API to add a Job resetting an Episys prompt back to a value which requires manual input. It is best practice to do this for each prompt answered after running the RSJ Job. We do not what to leave old hard-coded values.



```
param(
#-----------------------------------------------------
#  Static Parameters
#-----------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#-----------------------------------------------------
#  Dynamic Parameters
#-----------------------------------------------------
    [string]$JobName,
    [string]$Prompt
)
```



Windows Definition

Job Action: Embedded Scrip

User Id: Use Service Account

Script: EASE-RESET Script Type: PowerShell Version: LATEST

Runner: PowerShell Run Command Template: powershell.exe -ExecutionPolicy Bypass -File $FILE $ARGUMENTS

Arguments: [[EASE-CONNECTION]] -JobName CUSC.MISMATCH.POST -Prompt "Exception Batch ID"

Working Dir.:

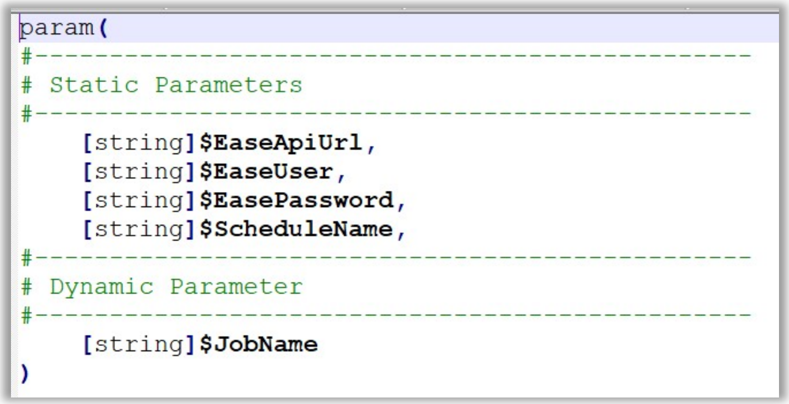
This script has two Dynamic Parameters:

- **-JobName**
  The name of the nested Episys Job which contains a prompt which needs to be reset.
- -**Prompt**
  The prompt within the Job above which needs to be set back to requiring manual input.

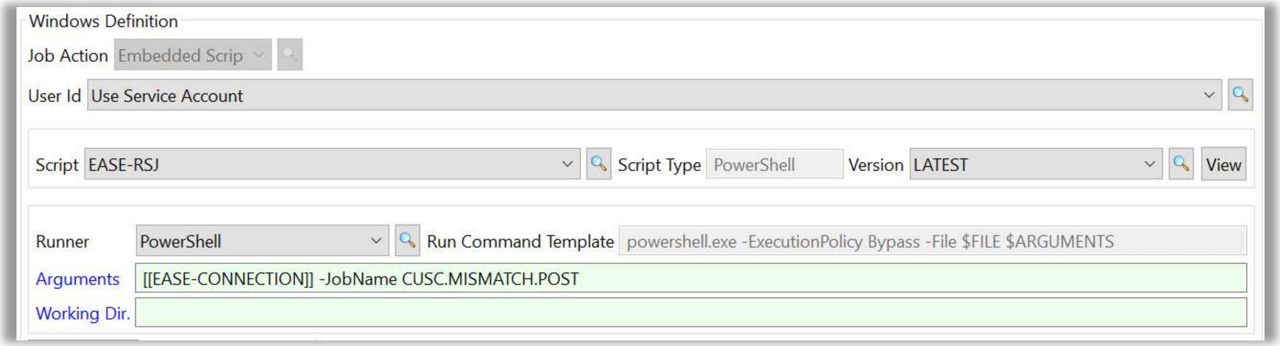
Run Symitar Job (RSJ)

The **EASE-RSJ.ps1** script uses the OpCon API to add a Job running an Episys Batch Job. A few things to know about this script:

- The RSJ Jobs are always set to single_thread.
- The Edit File switch is not provided by this Job.
- The Restart Point parameter is not provided by this job.



```
param(
#-------------------------------------------------
# Static Parameters
#-------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#-------------------------------------------------
# Dynamic Parameter
#-------------------------------------------------
    [string]$JobName
)
```



Windows Definition

Job Action: Embedded Scrip

User Id: Use Service Account

Script: EASE-RSJ

Script Type: PowerShell

Version: LATEST

Runner: PowerShell

Run Command Template: powershell.exe -ExecutionPolicy Bypass -File $FILE $ARGUMENTS

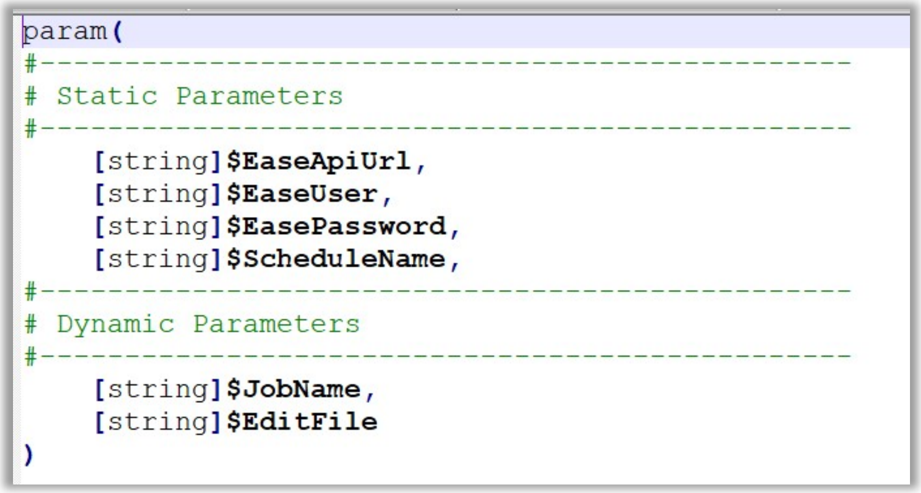
Arguments: [[EASE-CONNECTION]] -JobName CUSC.MISMATCH.POST

Working Dir.:

The only parameter to add to the command line is the -JobName. This will be the name of the Episys Job you are running.

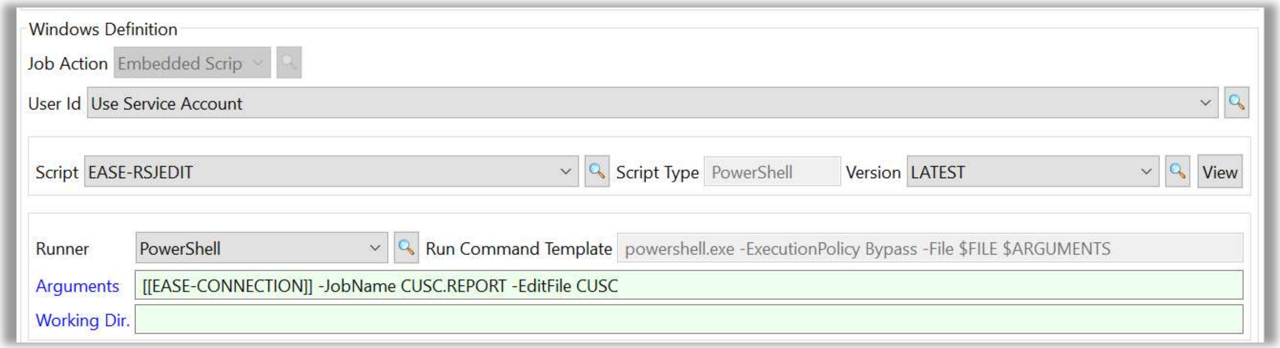
Run Symitar Job (RSJ) with an Edit File

The **EASE-RSJEDIT.ps1** script uses the OpCon API to add a Job running an Episys Batch Job which is doing an Edit Run.



```
param(
#-----------------------------------------------------
# Static Parameters
#-----------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#-----------------------------------------------------
# Dynamic Parameters
#-----------------------------------------------------
    [string]$JobName,
    [string]$EditFile
)
```

This is like the EASE-RSJ script, but it has an additional -EditFile switch. This will specify the name of the Edit File. The file is expected to be in the Letter Files directory.



Windows Definition

Job Action: Embedded Scrip

User Id: Use Service Account

Script: EASE-RSJEDIT

Script Type: PowerShell

Version: LATEST

Runner: PowerShell

Run Command Template: powershell.exe -ExecutionPolicy Bypass -File $FILE $ARGUMENTS

Arguments: [[EASE-CONNECTION]] -JobName CUSC.REPORT -EditFile CUSC

Working Dir.:

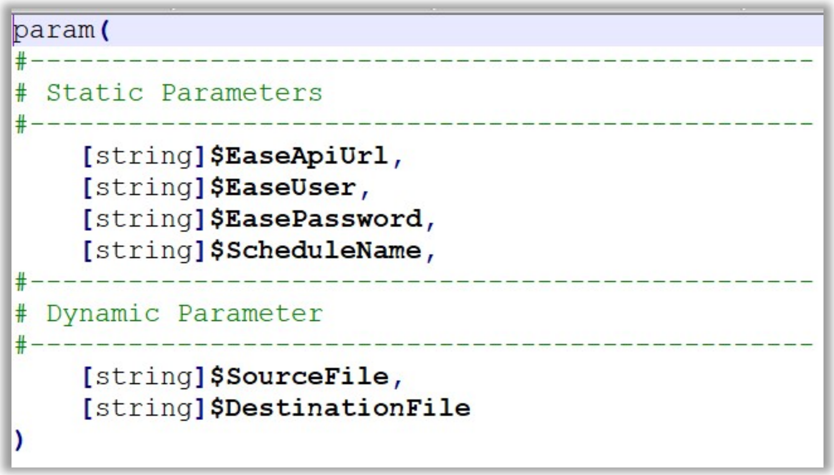
There are two parameters in this Job:

- **-JobName**
  The name of the Episys Job to run.
- **-EditFile**
  The name of the file to use as the edit file.

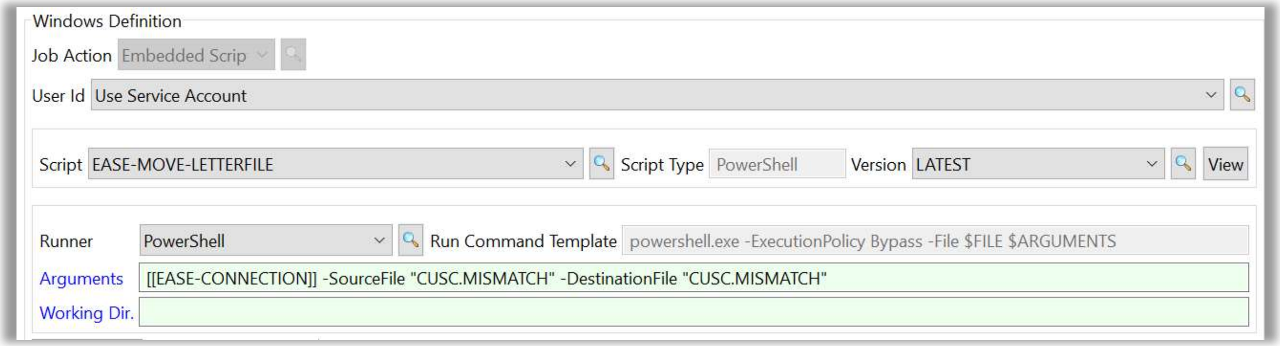
Rename a Letter File delivered from JHA’s DMZ

The **EASE-LETTERFILE-RENAME.ps1** script uses the OpCon API to add a Job which renames a Letter File removing the prefix MoveIT added to the file name. This is required if the Letter File name is hard coded in the Episys Job using the Letter File.

Note: A prefix is added to the file name during the transfer to Episys. This prefix should NOT be included in the -SourceFile parameter. It is hard coded in the EASE Job so you only need to include the name before the prefix was added.



```
param(
#--------------------------------------------------
# Static Parameters
#--------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#--------------------------------------------------
# Dynamic Parameter
#--------------------------------------------------
    [string]$SourceFile,
    [string]$DestinationFile
)
```



Windows Definition

Job Action: Embedded Scrip

User Id: Use Service Account

Script: EASE-MOVE-LETTERFILE Script Type: PowerShell Version: LATEST

Runner: PowerShell Run Command Template: powershell.exe -ExecutionPolicy Bypass -File $FILE $ARGUMENTS

Arguments: [[EASE-CONNECTION]] -SourceFile "CUSC.MISMATCH" -DestinationFile "CUSC.MISMATCH"

Working Dir.:

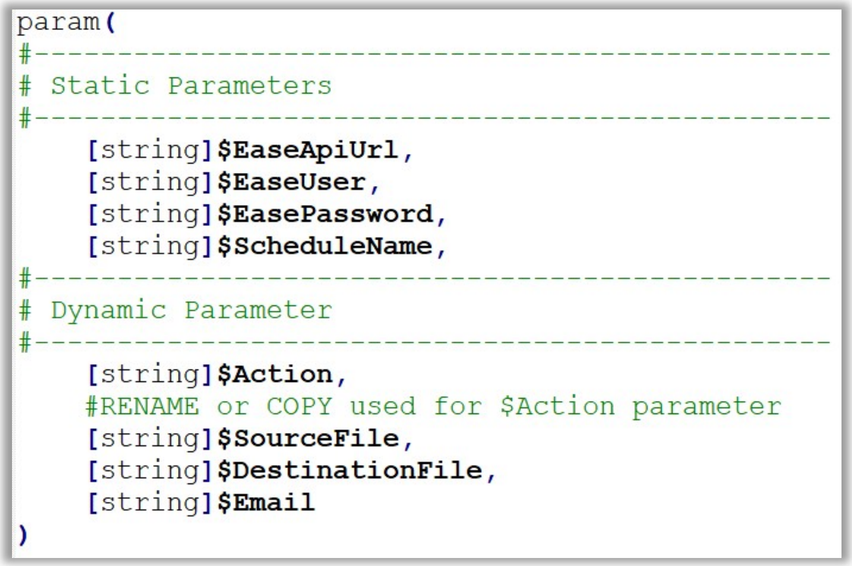
This Job has two parameters:

- **-SourceFile**
  This is the name of the Letter File we are copying.
  - Remember to not include the Prefix added during the transfer from JHA’s DMZ to Episys.
  - Example:
    The original file name is CUSC.MISMATCH. When the file gets transferred from the JHA DMZ to Symitar’s Letter Files Directory it gets renamed to EC999_ CUSC.MISMATCH. When running the rename script the -SourceFile should be CUSC.MISMATCH and NOT EC999_ CUSC.MISMATCH

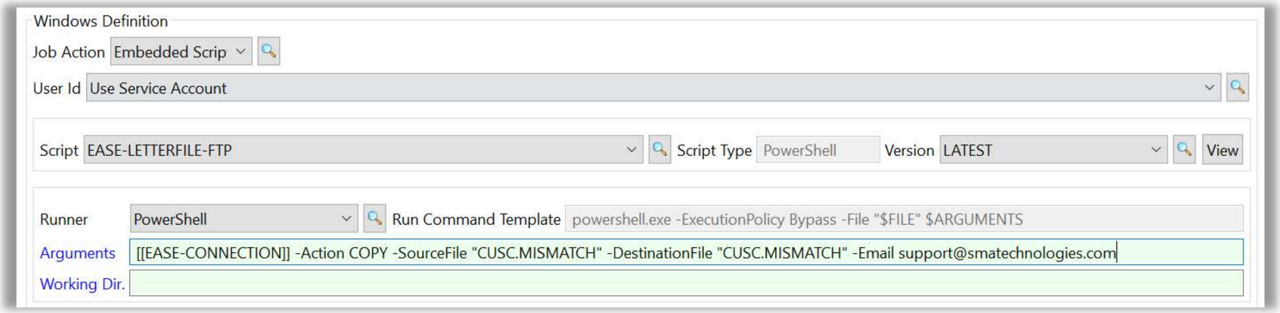
- **-DestinationFile**
  This is the name that will be given to the destination file. A prefix will automatically be added to the name supplied. Because of this, you can use the same name supplied in the -SourceFile parameter.

Copy or Rename a Letter File before transferring it to JHA’s DMZ

The **EASE-LETTERFILE-FTP.ps1** script uses the OpCon API to add a Job which either copies or renames a Letter File adding the prefix to the file name which MoveIT looks for when transfer the file to JHA’s DMZ.



```
param(
#------------------------------------------------
# Static Parameters
#------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#------------------------------------------------
# Dynamic Parameter
#------------------------------------------------
    [string]$Action,
    #RENAME or COPY used for $Action parameter
    [string]$SourceFile,
    [string]$DestinationFile,
    [string]$Email
)
```



Windows Definition

Job Action: Embedded Scrip

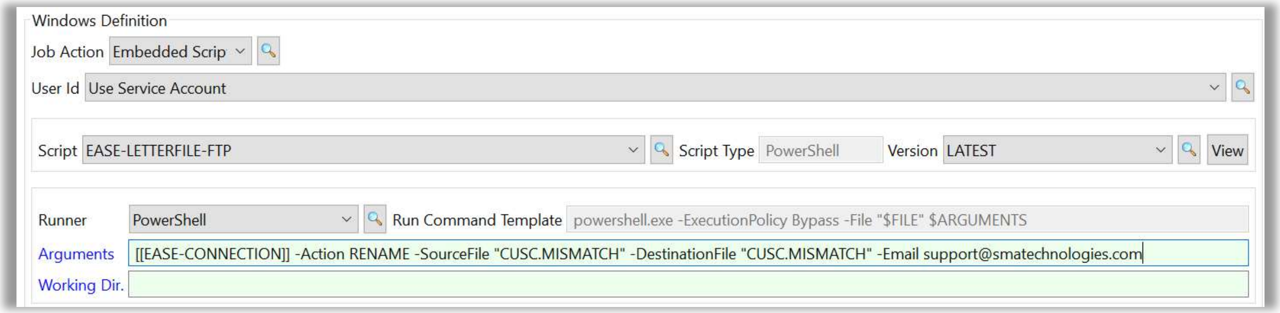
User Id: Use Service Account

Script: EASE-LETTERFILE-FTP Script Type: PowerShell Version: LATEST

Runner: PowerShell Run Command Template: powershell.exe -ExecutionPolicy Bypass -File "$FILE" $ARGUMENTS

Arguments: [[EASE-CONNECTION]] -Action COPY -SourceFile "CUSC.MISMATCH" -DestinationFile "CUSC.MISMATCH" -Email support@smatechnologies.com

Working Dir.:



Windows Definition

Job Action: Embedded Scrip

User Id: Use Service Account

Script: EASE-LETTERFILE-FTP Script Type: PowerShell Version: LATEST

Runner: PowerShell Run Command Template: powershell.exe -ExecutionPolicy Bypass -File "$FILE" $ARGUMENTS

Arguments: [[EASE-CONNECTION]] -Action RENAME -SourceFile "CUSC.MISMATCH" -DestinationFile "CUSC.MISMATCH" -Email support@smatechnologies.com

Working Dir.:

This Job has four parameters:

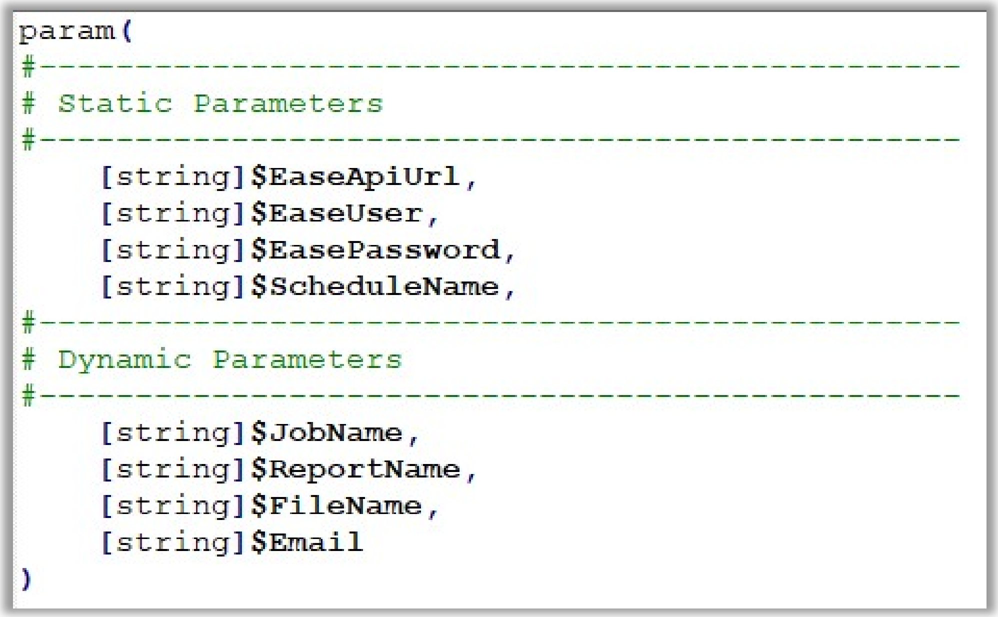
- **-Action**
  There are two options for this parameter.
  - **COPY** is used if the Letter File is constantly appended two and you need to keep the Letter File.
  - **RENAME** is used when you need to delete the Letter File before the process is run again.
- **-SourceFile**
  This is the name of the Letter File we are copying.
  - Remember to not include the Prefix added during the transfer from JHA's DMZ to Episys.
  - <u>Example</u>:
    The original file name is CUSC.MISMATCH. When the file gets transferred from the JHA DMZ to Symitar's Letter Files Directory it gets renamed to EC999_ CUSC.MISMATCH. When running the rename script the -SourceFile should be CUSC.MISMATCH and NOT EC999_ CUSC.MISMATCH

- **-DestinationFile**
  This is the name that will be given to the destination file. A prefix will automatically be added to the name supplied. Because of this, you can use the same name supplied in the -SourceFile parameter.
- **-Email**
  This is the email address which will used to notify once the file is ready for pickup on the DMZ. This is a required parameter. Using a group email account is the best practice.

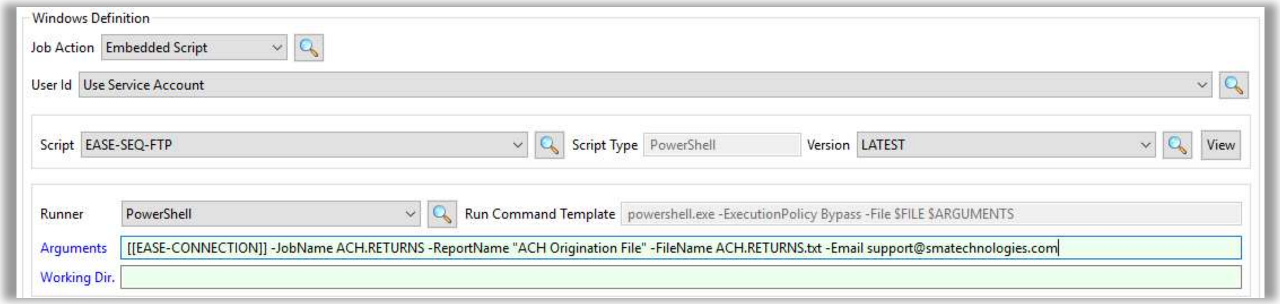
Transfer a Report off Episys

The **EASE-SEQ-FTP.ps1** script uses the OpCon API to add three Jobs into the EASE environment. These three Jobs perform the following functions:

1. The first Job searches for the sequence number of the defined report and stores it in a Schedule Instance Property within the EASE Schedule.
    a. <u>**Note**</u>: All sequence number Jobs update the same Schedule Instance property. Because of this you need to make sure only one of these Jobs run at a time. OpCon Resource Dependences are recommended.
2. The second Job copies this report from the “reports” directory into the Letter Files directory.
    a. This is done because the following FTP Job deletes the source file. We do not want to delete the file from the Reports directory, so a copy is made which can be safely transferred and deleted.
    b. A prefix is added to the report during the copy. This is to prevent overwriting other files in the Letter Files directory.
3. The third Job FTPs the file from Symitar to JHA’s DMZ.
    a. <u>**Note**</u>: A prefix will be added to the file name when it is copied from the “reports” directory into Letter Files.
    b. <u>**Example**</u>: File **test.txt** becomes **EC999_test.txt**.
    c. The -Email parameter is a required parameter. It is used by MoveIT to notify the specified email once the transfer is complete.



```
param(
#------------------------------------------------
#  Static Parameters
#------------------------------------------------
    [string]$EaseApiUrl,
    [string]$EaseUser,
    [string]$EasePassword,
    [string]$ScheduleName,
#------------------------------------------------
#  Dynamic Parameters
#------------------------------------------------
    [string]$JobName,
    [string]$ReportName,
    [string]$FileName,
    [string]$Email
)
```



Windows Definition

Job Action: Embedded Script

User Id: Use Service Account

Script: EASE-SEQ-FTP

Script Type: PowerShell

Version: LATEST

Runner: PowerShell

Run Command Template: powershell.exe -ExecutionPolicy Bypass -File $FILE $ARGUMENTS

Arguments: [[EASE-CONNECTION]] -JobName ACH.RETURNS -ReportName "ACH Origination File" -FileName ACH.RETURNS.txt -Email support@smatechnologies.com

Working Dir.:

This Job has four parameters:

- **-JobName**
  The name of the Episys Job which was run to create the report whose sequence number we are searching for.
- -**ReportName**
  The title of the report whose sequence number you need to collect. This will be used as the answer to the -Prompt listed below.
- **-FileName**
  This parameter is used to give the destination file name of the Job copying the report to Letter Files, as well as to define the source file for the Job which will transfer the file to JHA's DMZ.
  - The path is hard-coded within the Job in the EASE environment. DO NOT include the path in this parameter.
  - A prefix is added to the file name when it is copied to the Letter Files directory. That prefix is hard-coded within the subsequent FTP Job.
- **-Email**
  This is the email address which will used to notify once the file is ready for pickup on the DMZ. This is a required parameter. Using a group email account is the best practice.

Common Exit Codes

The EASE Connector could receive exit codes from four different areas. There is a possibility that the same exit could be used by multiple of these areas. The full message of within the Job Output should be able to help identify which area to start troubleshooting.

1. Defined Script Errors
2. Generic Powershell Errors
3. Generic REST API Errors
4. Program Errors

Defined Script Errors

The following errors are trapped and returned by the EASE Connector scripts. The EASE team will be the appropriate resource to help troubleshoot these errors.

- **600**
  This error occurs if the Schedule within EASE's environment is not built for the day. Please contact the EASE team to investigate why the Schedule was not built.
- **601**
  This error occurs when the Schedule within EASE's environment is in an **On Hold** status. Please check with the EASE team to determine when they will reactivate the Schedule.
- **602**
  This error occurs if a Job is run before the Schedule within EASE's environment is in a running status (if it is still in a Wait to Start status). The Schedule within EASE's environment starts when your agreed upon maintenance window ends.
- **603**
  This error occurs if a Job is run after the Schedule within EASE's environment closes for the day. The Schedule within EASE's environment ends when your agreed upon maintenance window begins.
- **650**
  This error occurs when the script is not successful in adding the specified Job within EASE's environment. Please have the EASE team verify that the proper privileges are setup for the Job Add request.
- **651**
  After the Job is added, the script polls to check its status. The 451 error occurs if the script cannot find the added Job. This could be caused by the lack of privileges or by something deleting the Job within EASE's environment. Please check with EASE to resolve this issue.
- **699**
  This error occurs when the script is unable to connect to EASE's OpCon API.
- **999**
  This is a timeout error which indicates the script has not been able to verify the status of the Job Add in an appropriate amount of time.

Generic Powershell Errors

The EASE scripts are all Powershell scripts. There is the potential for Powershell specific errors to occur. SMA Support will be the appropriate resource to help troubleshoot these errors. A few scenarios which would return a Powershell scripting exit code are:

- A required parameter is not entered
- A parameter name was mistyped causing it to be unrecognized
- The Batch User running the Job does not have the proper execution policy defined within Powershell
- Etc

Generic REST API Errors

We trapped many of these errors within the errors defined without the EASE scripts, but there is a chance you could run into REST API errors. SMA Support will be the appropriate resource to help troubleshoot these errors.

The following link describes the generic REST API errors.

https://restfulapi.net/http-status-codes/

Program Errors

The EASE scripts return the exit code provided by the program being run within the EASE environment. Programs run by the EASE scripts are:

- RSJ and its supporting programs
- OpCon UNIX Agent's File Arrival
- The AIX Move or Copy commands
- MoveIT Central