

Step-by-Step Guide for Masking Files on Mount Points

Version 2.0

By Serge De La Sablonniere

Table of Contents

<i>Step-by-Step Guide for Masking Files on Mount Points</i>	<i>1</i>
INTRODUCTION	3
THE DEMO LANDSCAPE.....	3
PREPARING THE FILES TO BE MASKED	3
CONFIGURING THE NFS SERVER.....	4
FIREWALL RULES AND PORTS.....	5
USING THE MASKING API CLIENT TO DEFINE THE MOUNT POINTS	5
PROVIDING THE FLAT FILE FORMAT TO DELPHIX MASKING	10
CREATING THE MASKING PROJECT	12

INTRODUCTION

In earlier versions of Delphix Masking, the only way to mask files was via FTP/SFTP. Starting with Delphix Masking 6.0, we enable users to directly mount and mask files over NFS & CIFS. This should dramatically simplify the process of file masking. For example, it will be possible to mask files on AWS through mounting S3 buckets.

This guide provides a visual aid to the steps needed to create a masking project for files residing in a shared NFS mount.

Please note that this guide is not intended to replace formal training. Please consult the Delphix Masking documentation and knowledge base to get the most up to date information about our product. Delphix also have eLearning available on Masking (please consult your local Delphix contact for more information)

Delphix Masking Documentation: <https://maskingdocs.delphix.com/>

Delphix Masking Knowledge Base: https://support.delphix.com/Delphix_Masking_Engine

THE DEMO LANDSCAPE

You will need the following components:

Component	IP Address in my demo FYI
Delphix Masking Engine 6.0	172.16.55.129
Linux Server	172.16.55.190

PREPARING THE FILES TO BE MASKED

In this demo, we will put some files that we need to mask in a folder. We will then export this folder as an NFS share so that the masking engine can connect to it via NFS.

Capture the IP address of your Linux server (you'll need it later)

```
[delphix@devops ~]$ hostname -I  
172.16.55.190 192.168.122.1
```

Create a new directory structure that will contain the source files and the destination target directory that will hold the results of our masked files.

```
[delphix@devops ~]$  
[delphix@devops ~]$ mkdir Masking_Demo_Files  
[delphix@devops ~]$ mkdir Masking_Demo_Files/Source_Files  
[delphix@devops ~]$ mkdir Masking_Demo_Files/Target_Files  
[delphix@devops ~]$
```

Create some files to mask and place them in the newly created “Source_Files” directory. In my demo, I will create 2 csv files delimited with “|”. In a later step, we will define the file structure so that our masking engine can understand the field mappings.

```
[delphix@devops Source_Files]$ pwd  
/home/delphix/Masking_Demo_Files/Source_Files  
[delphix@devops Source_Files]$  
[delphix@devops Source_Files]$ ls  
007.csv  GoT.csv  
[delphix@devops Source_Files]$  
[delphix@devops Source_Files]$ cat 007.csv  
12345|Daniel|Craig|Skyfall  
67890|Pierce|Brosnan|GoldenEye  
65324|Sean|Connery|You Only Live Twice  
72663|Roger|Moore|Moonraker  
[delphix@devops Source_Files]$  
[delphix@devops Source_Files]$  
[delphix@devops Source_Files]$ cat GoT.csv  
12345|Emilia|Clark|The Mother of Dragons  
67890|Sophie|Turner|The Queen of Winterfell  
65324|Kit|Harrington|Commander of the Night Watch  
72663|Lena|Headley|Queen of the Seven Kingdoms  
[delphix@devops Source_Files]$
```

CONFIGURING THE NFS SERVER

My Linux demo server is built on CentOS 7. Good instructions for setting up NFS for this OS can be found here: <https://www.howtoforge.com/tutorial/setting-up-an-nfs-server-and-client-on-centos-7/>. There are the steps I took for my demo server:

Install the NFS server packages

```
$ sudo yum install nfs-utils
```

Enable and start the NFS server service

```
$ sudo systemctl enable nfs-server.service  
$ sudo systemctl start nfs-server.service
```

Edit the file `/etc/exports` and add the directory to be shared as an NFS export. The IP address we add will limit the share to be only visible accessible from the masking engine. You will also need the userid and groupid associated with the directory you created. In my case the user is delphix (userid 1000) and the group is delphix (groupid 1000)

```
[delphix@devops Source_Files]$ cat /etc/exports
/home/delphix/masking6.0.mount 172.16.55.129(rw, sync)
/home/delphix/Masking_Demo_Files 172.16.55.129(rw, all_squash, anonuid=1000, anongid=1000)
```

Whenever you update the file `/etc/exports`, you have run

```
$ sudo exportfs -a
```

That should do it for the configuration steps needed on the Linux server...Now let's move on to the Delphix Masking Engine.

FIREWALL RULES AND PORTS

Before heading to your masking engine, make sure you don't have any firewall rules or closed ports that would prevent communication between the Masking Engine and the NFS server. For Linux, Port 111 (TCP and UDP) and 2049 (TCP and UDP) are used for the NFS server.

USING THE MASKING API CLIENT TO DEFINE THE MOUNT POINTS

In the previous steps, we created a Linux NFS export share `"/home/delphix/Masking_Demo_Files"`. We will now provide to the Masking Engine the information it needs in order to register & connect to this NFS share. For Delphix Masking version 6.0 we can use the [Masking API Client](#) to accomplish this.

Additional reading material:

https://maskingdocs.delphix.com/Connecting_Data/Managing_Remote_Mounts/#mount-information

Access the Delphix Masking API Client

```
172.16.55.129/masking/api-client
```

Initiate the login of the API client to the masking engine.

Click login

Authorize			
Masking API			
Schema for the Masking Engine API			
algorithm	Show/Hide	List Operations	Expand Operations
logging	Show/Hide	List Operations	Expand Operations
application	Show/Hide	List Operations	Expand Operations
applicationSettings	Show/Hide	List Operations	Expand Operations
asyncTask	Show/Hide	List Operations	Expand Operations
columnMetadata	Show/Hide	List Operations	Expand Operations
databaseConnector	Show/Hide	List Operations	Expand Operations
databaseRuleset	Show/Hide	List Operations	Expand Operations
domain	Show/Hide	List Operations	Expand Operations
encryptionKey	Show/Hide	List Operations	Expand Operations
environment	Show/Hide	List Operations	Expand Operations
execution	Show/Hide	List Operations	Expand Operations
executionComponent	Show/Hide	List Operations	Expand Operations
executionEvent	Show/Hide	List Operations	Expand Operations
sync	Show/Hide	List Operations	Expand Operations
fileConnector	Show/Hide	List Operations	Expand Operations
fileDownload	Show/Hide	List Operations	Expand Operations
fileFieldMetadata	Show/Hide	List Operations	Expand Operations
fileFormat	Show/Hide	List Operations	Expand Operations
fileMetadata	Show/Hide	List Operations	Expand Operations
fileRuleset	Show/Hide	List Operations	Expand Operations
fileUpload	Show/Hide	List Operations	Expand Operations
knowledgeBaseInfo	Show/Hide	List Operations	Expand Operations
login	Show/Hide	List Operations	Expand Operations
mainframeDatasetConnector	Show/Hide	List Operations	Expand Operations
mainframeDatasetFieldMetadata	Show/Hide	List Operations	Expand Operations
mainframeDatasetFormat	Show/Hide	List Operations	Expand Operations
mainframeDatasetMetadata	Show/Hide	List Operations	Expand Operations
mainframeDatasetRuleset	Show/Hide	List Operations	Expand Operations
workspaceInfo	Show/Hide	List Operations	Expand Operations

You can click on the Example Value box to pre-fill the login connection box (it's the box with the white background).

Enter the username and password to connect to your masking engine.

Click the **Try it out** button.

login

Show/Hide | List Operations | Expand Operations

POST /login

Log in to the Masking Engine

Response Class (Status 201)

Success

Model Example Value

{
 "Authorization": "415aac5d-SOME-RANDOM-STRING-af6cf78dc49e"
}

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
login	{ "username": "admin", "password": "XXXXXXXXXX" }	The login object	body	Model Example Value

Parameter content type: application/json

{
 "username": "maskinguser",
 "password": "secret123"
}

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized access		

Try it out!

You should observe a Response Code of 200 (the API call succeeded).
Copy the Authorization key as you will need it in the next step.

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized access		

[Try it out!](#) [Hide Response](#)

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/' --header 'username: "Admin" \ --header 'password: "Admin-12" \ --header 'http://172.16.55.129/masking/api/login'
```

Request URL

```
http://172.16.55.129/masking/api/login
```

Response Body

```
{  "Authorization": "d9e82d07-62fa-4257-b102-abbaeb5fb63b"}
```

Response Code

```
200
```

Create a Session with the masking engine using the Authorization Key from the previous step. Enter the key and click the Authorize button.

Masking API
Schema for the Masking Engine API

Schema	Show/Hide	List Operations	Expand Operations
algorithm	Show/Hide	List Operations	Expand Operations
logging	Show/Hide	List Operations	Expand Operations
application	Show/Hide	List Operations	Expand Operations
applicationSettings	Show/Hide	List Operations	Expand Operations
asyncTask	Show/Hide	List Operations	Expand Operations
columnMetadata	Show/Hide	List Operations	Expand Operations
databaseConnector	Show/Hide	List Operations	Expand Operations
databaseRuleset	Show/Hide	List Operations	Expand Operations
domain	Show/Hide	List Operations	Expand Operations
encryptionKey	Show/Hide	List Operations	Expand Operations
environment	Show/Hide	List Operations	Expand Operations
execution	Show/Hide	List Operations	Expand Operations
executionComponent	Show/Hide	List Operations	Expand Operations
executionEvent	Show/Hide	List Operations	Expand Operations
sync	Show/Hide	List Operations	Expand Operations
fileConnector	Show/Hide	List Operations	Expand Operations
fileDownload	Show/Hide	List Operations	Expand Operations
fileFieldMetadata	Show/Hide	List Operations	Expand Operations
fileRequest	Show/Hide	List Operations	Expand Operations

Available authorizations

Api key authorization

name: Authorization

in: header

value: 082468f0-c390-415a-1

[Authorize](#) [Cancel](#)

Next step is to register the NFS Mount Point into the Masking Engine.

mainframeDatasetRuleset Show/Hide List Operations Expand Operations

maskingJob Show/Hide List Operations Expand Operations

mountFilesystem Show/Hide List Operations Expand Operations

Method	URL	Action
GET	/mount-filesystem	Get all mounts
POST	/mount-filesystem	Create filesystem mount
DELETE	/mount-filesystem/{mountID}	Delete filesystem mount
GET	/mount-filesystem/{mountID}	Get mount by ID
PUT	/mount-filesystem/{mountID}	Update filesystem mount
PUT	/mount-filesystem/{mountID}/connect	Connect filesystem mount
PUT	/mount-filesystem/{mountID}/disconnect	Disconnect filesystem mount
PUT	/mount-filesystem/{mountID}/remount	Remount filesystem mount

Enter the body payload and click the **Try it out** button.

mountName: That is the name of mount point you will see in Delphix Masking
hostAddress: The IP of the location of the NFS Server (in my case the Linux box)
type: I use NFS3 (other options available here, consult the documentation)
options: leave empty
connectOnStartup: true (if the Masking Engine reboots, it will try to reconnect)

Response Content Type: application/json

Parameters

Parameter	Value	Description
body	<pre>{ "mountName": "demo_nfs_mount", "hostAddress": "172.16.55.190", "mountPath": "/home/delphix/Masking_Demo_Files", "type": "NFS3", "options": "", "connectOnStartup": true }</pre>	The filesystem to mount

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model
400	Bad request	
401	Unauthorized access	
404	Not found	
409	Conflict	

Try it out!

You should observe a Response Code of 200 meaning the registration succeeded. Notice however the status is showing DISCONNECTED. Take a note of the mountID result (in my case 3) because you will need it to activate the mount point.

Try it out! **Hide Response**

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: b1e' \
  -d '{
    "mountName": "demo_nfs_mount",
    "hostAddress": "172.16.55.190",
    "mountPath": "/home/delphix/Masking_Demo_Files",
    "type": "NFS3",
    "options": "",
    "connectOnStartup": true
  }' http://172.16.55.129/masking/api/mount-filesystem'
```

Request URL

http://172.16.55.129/masking/api/mount-filesystem

Response Body

```
{
  "mountId": 3,
  "mountName": "demo_nfs_mount",
  "hostAddress": "172.16.55.190",
  "mountPath": "/home/delphix/Masking_Demo_Files",
  "type": "NFS3",
  "connectOnStartup": true,
  "options": "",
  "status": "DISCONNECTED"
}
```

Response Code

200

Connecting and activating the NFS Mount Point

From the API Client, select the **Connect Filesystem Mount**

mountFilesystem

Show/Hide

List Operations

Expand Operations

GET

/mount-filesystem

Get all mounts

POST

/mount-filesystem

Create filesystem mount

DELETE

/mount-filesystem/{mountID}

Delete filesystem mount

GET

/mount-filesystem/{mountID}

Get mount by ID

PUT

/mount-filesystem/{mountID}

Update filesystem mount

PUT

/mount-filesystem/{mountID}/connect

Connect filesystem mount

PUT

/mount-filesystem/{mountID}/disconnect

Disconnect filesystem mount

PUT

/mount-filesystem/{mountID}/remount

Remount filesystem mount

Enter the mountID generated by the previous step

PUT /mount-filesystem/{mountID}/connect Connect filesystem mount

Response Class (Status 200)
Success

Model: Example Value

```
{
  "mountName": "my_mount",
  "hostAddress": "some.host.com",
  "mountPath": "/path/to/my/mount",
  "type": "NFS4",
  "options": "comma,delimited,options,list",
  "connectOnStartup": true
}
```

Response Content Type: application/json

Parameter	Value	Description	Parameter Type	Data Type
mountID	3	The ID of the mount to connect	path	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Bad request		
401	Unauthorized access		
404	Not found		

Try it out! [Hide Response](#)

Curl

```
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/j
```

Request URL

```
http://172.16.55.129/masking/api/mount-filesystem/3/connect
```

Response Body

```
{
  "mountId": 3,
  "mountName": "demo_nfs_mount",
  "hostAddress": "172.16.55.190",
  "mountPath": "/home/delphix/Masking_Demo_Files",
  "type": "NFS3",
  "connectOnStartup": true,
  "options": "rw,nosuid,nodev,noexec,relatime,vers=3,rsize=262144,wsiz
```

Response Code

200

We have now executed all the required steps to register the NFS share into the Masking engine.

We can now proceed to create our masking project and run it.

PROVIDING THE FLAT FILE FORMAT TO DELPHIX MASKING

Remember the files we created earlier... We need to define the structure to Delphix Masking. It's very easy to do.

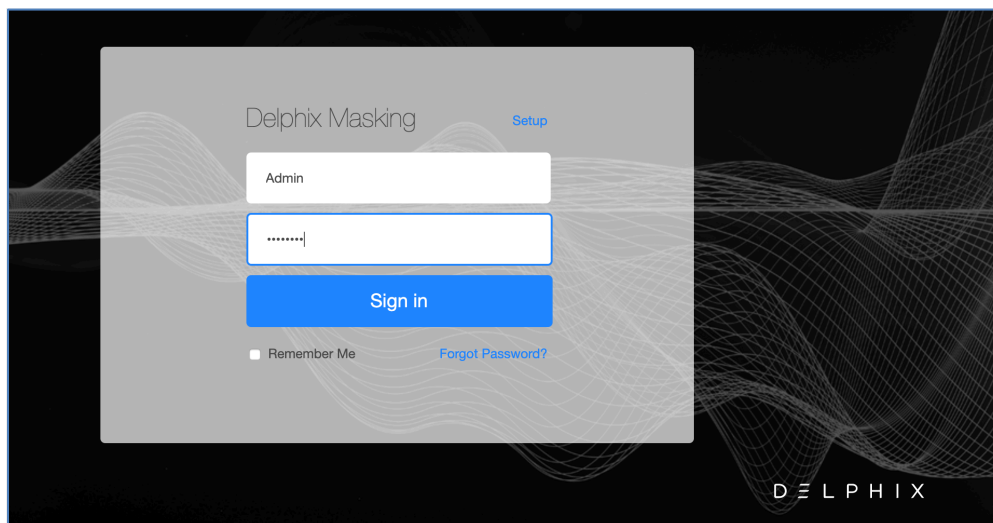
```
[delphix@devops Source_Files]$ pwd
/home/delphix/Masking_Demo_Files/Source_Files
[delphix@devops Source_Files]$
[delphix@devops Source_Files]$
[delphix@devops Source_Files]$ cat GoT.csv
12345|Emilia|Clark|The Mother of Dragons
67890|Sophie|Turner|The Queen of Winterfell
65324|Kit|Harrington|Commander of the Night Watch
72663|Lena|Headley|Queen of the Seven Kingdoms
[delphix@devops Source_Files]$
```

Create a file containing the name of the fields in the delimited flat file (you can create this file on your local desktop)



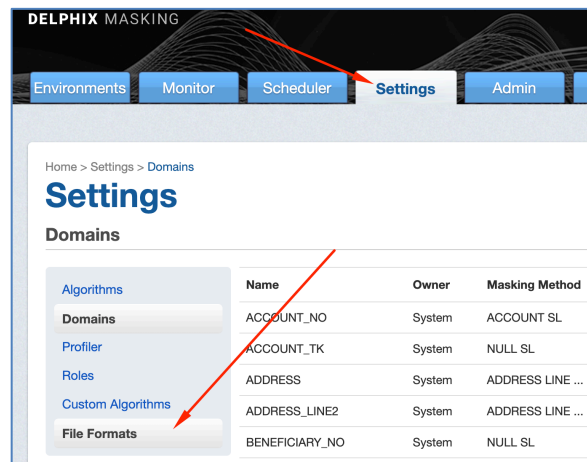
Connect to the Masking Engine

172.16.55.129/masking/login.do

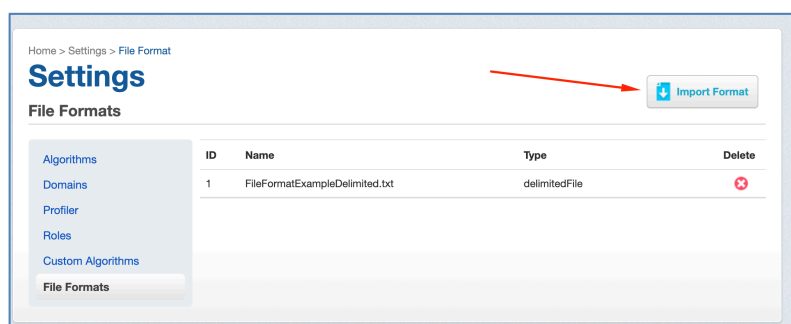


Load the file as a new file format type in Delphix Masking

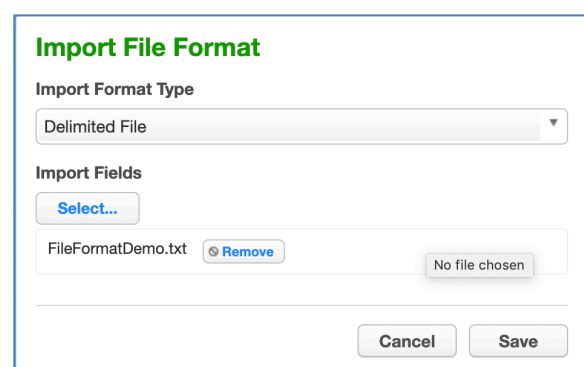
Go to Settings and then Select File Formats



Click Import Format

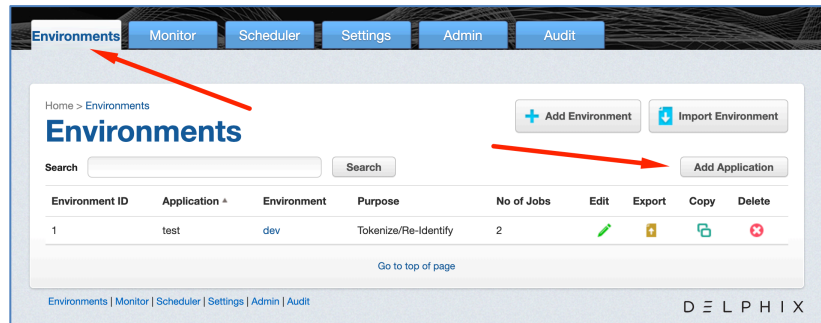


From the Import File Format dialogue box, select “Delimited File” as the Import Format Type, then click on “Select” and load the format file we just created. Finish this step by clicking Save.

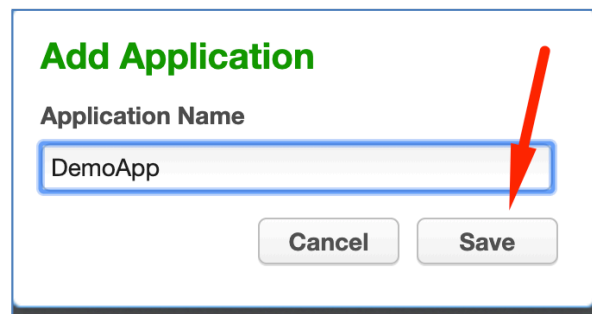


CREATING THE MASKING PROJECT

Navigate to Environment and Add a new Application



Choose a name for your Application and click



Its best practice for file masking to read the source files from one location and save the masked files into another location. We call that “On-The-Fly Masking”.

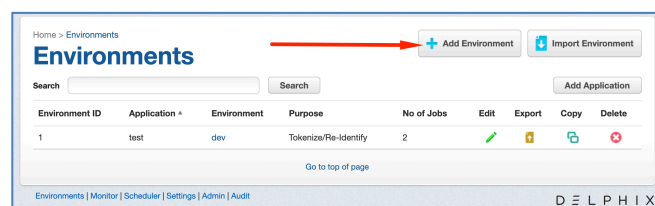
To prepare for that, we created 2 directories under “/home/delphix/”:

Source_Files will contain all the files we want to mask

Target_Files will be where Delphix masking will store the files once they are masked (this way we do not alter the source files and can repeat the masking process with consistent results)

Let’s first create the Environment that will be used to identify where are the source files

Select Add Environment, and give it the name “File_SRC_Env” and pick ‘Mask’ as the purpose



Add Environment

Application Name
 DemoApp

Environment Name
 File_SRC_Env

Purpose
 Mask

☐ Enable Approval Workflow

Cancel Save Save & View

Click on the Environment Name we just created

Home > Environments

Environments

Search Search Add Application

Environment ID	Application *	Environment	Purpose	No of Jobs	Edit	Export	Copy	Delete
6	DemoApp	File_SRC_Env	Mask	0				

Go to top of page

Select Connector

Environments Monitor Scheduler Settings Admin Audit

Overview Connector Rule Set Inventory

Home > Environments > File_SRC_Env

File_SRC_Env

Profile Mask

Environment

Name	File_SRC_Env
Purpose	Mask
Application Name	DemoApp
Approval workflow	Disabled

Click on Create Connection

Overview Connector Rule Set Inventory

Home > Environments > File_SRC_Env > Connector

File_SRC_Env

Create Connection

Connector ID	Connector	Meta Data Source	Type	Edit	Delete
* indicates an extension to included connectors					

Environments | Monitor | Scheduler | Settings | Admin | Audit

DELPHIX

Enter these values:

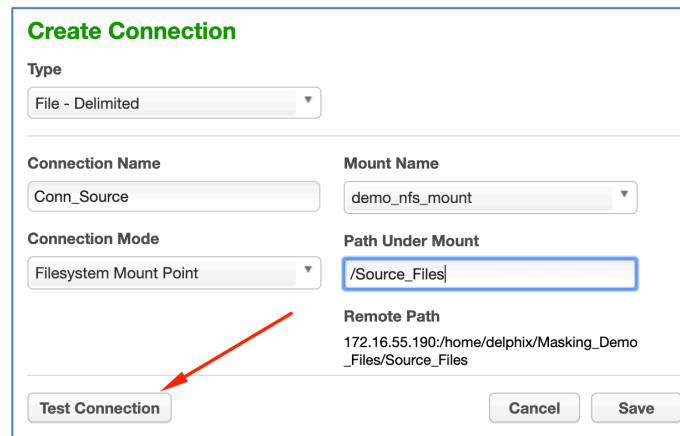
Type = File – Delimited

Connection Name = name of your choice (I use “Conn_Source”)

Connection Mode = Filesystem Mount Point

Mount Name = the name of the mount you created earlier using the Masking API Client. In my example I specify the subdirectory containing the source files I want to mask.

Before hitting Save, test the connection



Create Connection

Type: File - Delimited

Connection Name: Conn_Source

Mount Name: demo_nfs_mount

Connection Mode: Filesystem Mount Point

Path Under Mount: /Source_Files

Remote Path: 172.16.55.190:/home/delphix/Masking_Demo_Files/Source_Files

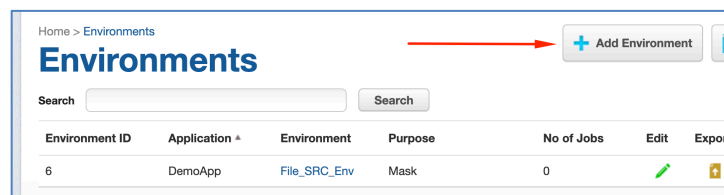
Test Connection Cancel Save

We’re done with the creation of our source environment. Now let’s build the destination environment...

Click on the bread crumb “Environment”



Click “Add Environment”, then create your new target environment by giving it this time the name “File_Dest_Env” and click “Save & View”



Add Environment

Application Name
 DemoApp

Environment Name
 File_Dest_Env

Purpose
 Mask

☐ Enable Approval Workflow

Cancel Save **Save & View**

Create a new connection

Overview Connector Rule Set Inventory

Home > Environments > File_Dest_Env > Connector

File_Dest_Env

+ Create Connection

Connector ID	Connector	Meta Data Source	Type	Edit	Delete
* indicates an extension to included connectors					

Environments | Monitor | Scheduler | Settings | Admin | Audit

D E L P H I X

Enter these values:

Type = File – Delimited

Connection Name = name of your choice (I use “Conn_Target”)

Connection Mode = Filesystem Mount Point

Mount Name = the name of the mount you created earlier using the Masking API Client. In my example I specify the subdirectory containing where I want Delphix to save the files once they are masked.

Create Connection

Type
 File - Delimited

Connection Name
 Conn_Target

Mount Name
 demo_nfs_mount

Connection Mode
 Filesystem Mount Point

Path Under Mount
 /Target_Files

Remote Path
 172.16.55.190:/home/delphix/Masking_Demo_Files/Target_Files

Test Connection Cancel Save

Next click on Rule Set

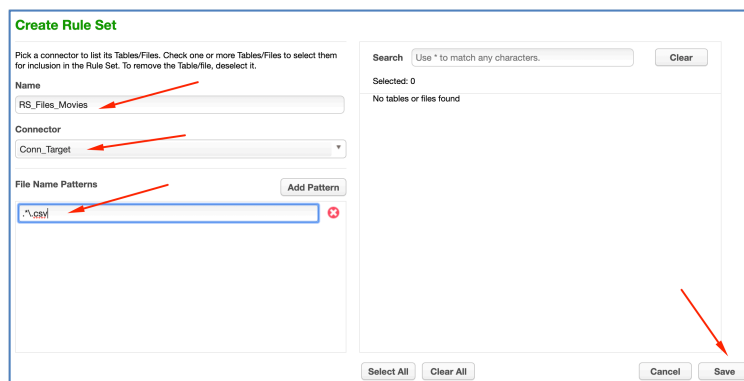


Click Create Rule Set

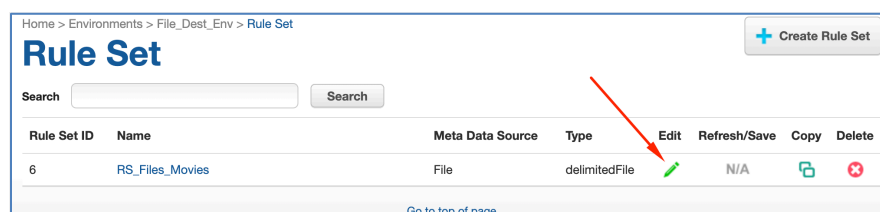


Give your Rule Set a name and pick the Connector we just created.
Since we want to use File Name Pattern matching, enter the regular expression as following:

`.*\..csv`



We must specify the format of the file that we will be masking. For this select the green pen to Edit the Rule Set, followed by the green pen next to the file



Select again the red pencil to edit the property for the file format matching the pattern “.*\.csv”

Home > Environments > File_Dest_Env > Rule Set > RS_Files_Movies

File_Dest_Env

Search Search

Rule Set Name

File or Pattern	Edit	Delete
.*\.csv		

1 - 1 of 1 items

[Go to top of page](#)

Then with the Edit File panel up, pick the following:

File Format: <pick the name for the file format we loaded earlier>

End of Record: Pick “LF terminated (Unix)”

Delimiter: Pick “|” – this is for my example. Your delimiter could be different off course

Edit File

Connector

File or Pattern

File Format

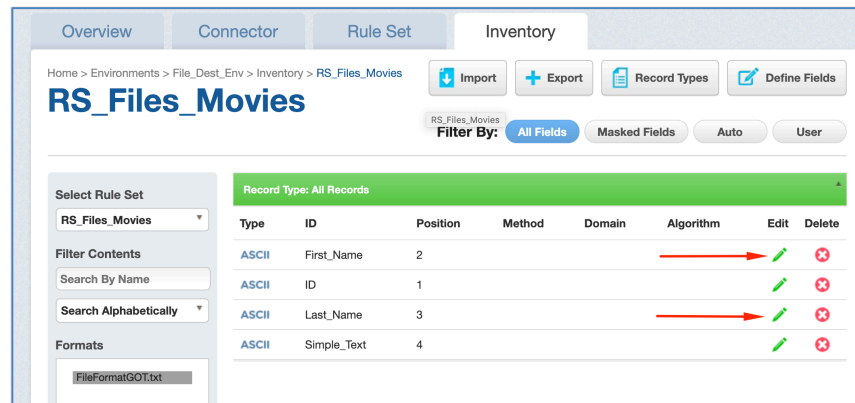
End Of Record

Delimiter

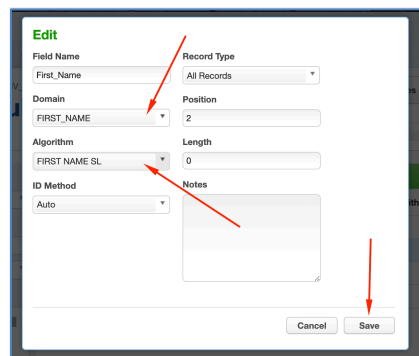
Text Enclosure

Cancel Save

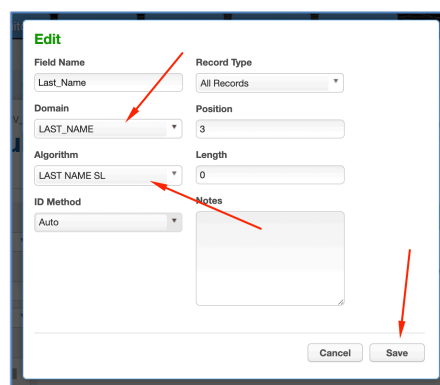
Navigate to the Inventory tab. Notice that no algorithms are assigned... Let's mask the first and last name. Click on the green pencil to the right of First_Name.



Assign FIRST_NAME as the Domain and FIRST_NAME_SL as the Algorithm



Repeat the same process for the LAST_NAME



Home > Environments > File_Dest_Env > Inventory > RS_Files_Movies

RS_Files_Movies

Filter By: **All Fields** Masked Fields Auto User

Select Rule Set
RS_Files_Movies

Filter Contents
Search By Name
Search Alphabetically

Formats
FileFormatGOT.txt

Record Type: All Records

Type	ID	Position ▲	Method	Domain	Algorithm	Edit	Delete
ASCII	ID	1					
ASCII	First_Name	2	Mask	FIRST_NAME	FIRST NAME SL		
ASCII	Last_Name	3	Mask	LAST_NAME	LAST NAME SL		
ASCII	Simple_Text	4					

Great work! All you now have to do is create a masking job and run it.

Navigate back to the Overview tab and select Mask

Overview **Connector** Rule Set Inventory

Home > Environments > File_Dest_Env

File_Dest_Env

Profile Mask

Environment

Name	File_Dest_Env
Purpose	Mask
Application Name	DemoApp
Approval workflow	Disabled

Give your job a name, select 'On-The-Fly' as a Masking Method and pick the Rule Set you created earlier. Specify the Source Environment and Source Connector then click Save.

Create Masking Job

Job Name: Mask_Movie_Actors

Feedback Size:

Masking Method: On-The-Fly

Comments:

Target: File_Dest_Env

☐ Multi Tenant

Rule Set: RS_Files_Movies

Email: serge@delphix.com

Source Environment: File_SRC_Env

Source Connector: Conn_Source






No. of Streams: 1

Min Memory: In MB






Max Memory: In MB

If Nonconforming Data is encountered
☐ Stop job on first occurrence






Start the masking job by clicking on the blue play button

Job ID ▾	Name	Rule Set	Completed	Status	Action	Edit	Delete
5	 Mask_Movie_Actors	RS_Files_Movies	...	 Created			

The status will change to Running

Job ID ▾	Name	Rule Set	Completed	Status	Action	Edit	Delete
5	 Mask_Movie_Actors	RS_Files_Movies	...	 Running			

When masking is completed, the status will change to Succeeded

Job ID ▾	Name	Rule Set	Completed	Status	Action	Edit	Delete
5	 Mask_Movie_Actors	RS_Files_Movies	2020-04-21 05:22	 Succeeded			

Now validate that masking did its job.... Look at the before and after and validate that First_Name and Last_Name on the csv files got masked!

File 007.csv

Before	After
<pre>/home/delphix/Masking_Demo_Files/Source_Files [delphix@devops Source_Files]\$ cat 007.csv 12345 Daniel Craig Skyfall 67890 Pierce Brosnan GoldenEye 65324 Sean Connery You Only Live Twice 72663 Roger Moore Moonraker</pre>	<pre>/home/delphix/Masking_Demo_Files/Target_Files [delphix@devops Target_Files]\$ cat 007.csv 12345 Janel Bunyan Skyfall 67890 Genoveva Cutting GoldenEye 65324 Vernie Bristed You Only Live Twice 72663 Roy Berkeley Moonraker</pre>

File GoT.csv

Before	After
<pre>/home/delphix/Masking_Demo_Files/Source_Files [delphix@devops Source_Files]\$ cat GoT.csv 12345 Emilia Clark The Mother of Dragons 67890 Sophie Turner The Queen of Winterfell 65324 Kit Harrington Commander of the Night Watch 72663 Lena Headley Queen of the Seven Kingdoms</pre>	<pre>/home/delphix/Masking_Demo_Files/Target_Files [delphix@devops Target_Files]\$ cat GoT.csv 12345 Maria Banvard The Mother of Dragons 67890 Corina Redden The Queen of Winterfell 65324 Eddie Stapleton Commander of the Night Watch 72663 Anabel Sellick Queen of the Seven Kingdoms</pre>

Congratulations, you have taken your first steps in file masking on NFS mount points.