FILENET P8

Identifying Embedded Links in Content Services Documents

WHITE PAPER

April 2006

The Issue

FileNet Panagon IDM Web Services was the thin client environment for accessing Content Services and Image Services content via a web browser. The architecture of IDM Web Services retrieved content via the document ID. Virtually all links to content from within IDM Web Services follows at minimum the following structure.

http://<servername>/idmws/doccontent.asp?DocId=011980003

This structure has remained the same throughout the lifecycle of IDM Web Services with the majority of links containing additional information in the query string. If the query string was created with a custom program and not the IDMWS tool it is possible the exact query string may be difficult to predict. Many innovative FileNet customers have come to understand this structure and put it to use in new ways in their organizations. They have used these links outside of the Web Services applications in many innovative ways. For example, they might send these links in email messages to colleagues or customers. They might use them in non FileNet web applications. They also have used these links inside other documents stored in Content Services. The figure below describes this scenario.

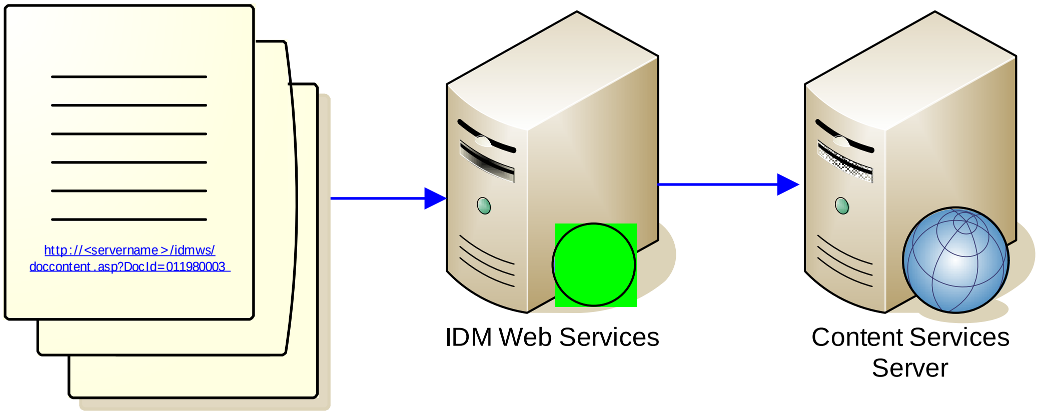


Figure1

This strategy for utilizing links inside other documents is known as “embedded links”. The links are embedded in other documents in the library. This strategy works well while the documents are in the Content Services library which the links refer to. When the documents are migrated to another system, such as from Content Services to P8 Content Manager, the embedded links are no longer relevant. The server name is typically no longer valid. The URL structure is no longer the same, and finally the document id is no longer valid either.

Options

There are a few options available to resolve this issue, each of which start with identifying the existing links. Once the links have been identified, they may be modified or deleted.

Identifying the Embedded Links

Identification of embedded links can be done in a semi-automated fashion by using the tools available with the operating system of the storage manager server. The procedure described below is for Microsoft Windows servers. Similar methods may be used on UNIX servers as well, although the syntax would be different.

Batch Process

For this process we will be using the batch file included in this document as Appendix A. To use the batch file select all the text from the Appendix A and paste it into a new text file using a text editor such as notepad. Save the file with a .bat extension. It can then be executed as a batch file.

The batch file will search for all documents in all shelves which contain the text "**doccontent.asp**". If you use the batch file be sure to edit the line starting with "`SET LIBRARY=`" to reflect the directory structure on your server. It will write the name of each file to a file called "**Links.txt**" in the stacks directory on the storage server. The file will look something like this.

```
shelf001\_26y2s__.__2
shelf001\_26y2s__.__3
shelf002\_26y2s__.__3
```

We will then need to strip out the part of each line that identifies the shelf such as `shelf001\`. This will result in a file with only the encrypted file names. We will edit the file using "**edlin**". **Edlin** is a command line text editor included with the operating system. It is an old tool dating back to the days of MS-DOS. It is useful for our purposes as we can somewhat automate the editing process. It is also extremely fast for the specific type of editing we want to do.

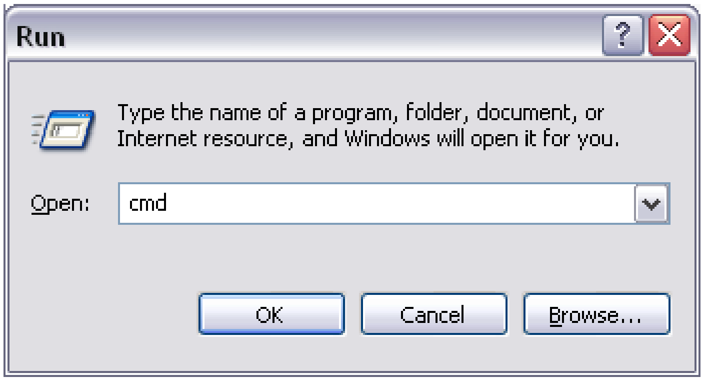
Once we have completed the editing of the file, the batch will once again take over and look up each Item ID and Version ID based on the encrypted file name. Once we have the Item ID and Version ID we can then locate the documents for editing.

Please note; the intent of this whitepaper is to help locate the documents inside Content Services which contain the embedded links. It does not find links outside of Content Services that may exist on user PCs or imbedded within emails. In addition, if a file within Content Services is encrypted, the link will not be found. The actual editing of the documents is beyond the scope of this document.

The procedure below is performed via the Command Prompt.

Open the command prompt

Start -> Run -> Type "cmd"



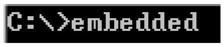
Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

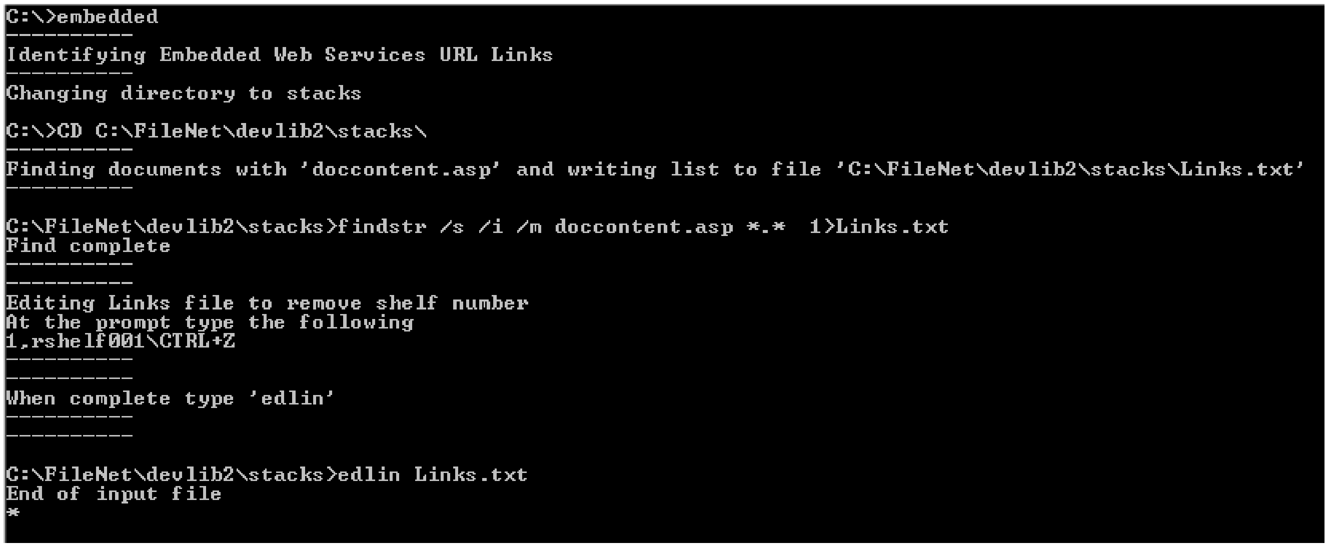
Execute the Embedded.bat batch file.

```
C:\>
```



```
C:\>embedded
```

You will see the following information on the command line…

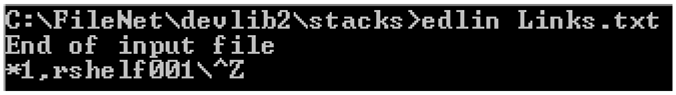


```
C:\>embedded
-----------
Identifying Embedded Web Services URL Links
-----------
Changing directory to stacks

C:\>CD C:\FileNet\devlib2\stacks\
-----------
Finding documents with 'doccontent.asp' and writing list to file 'C:\FileNet\devlib2\stacks\Links.txt'
-----------

C:\FileNet\devlib2\stacks>findstr /s /i /m doccontent.asp *.*  1>Links.txt
Find complete
-----------
-----------
Editing Links file to remove shelf number
At the prompt type the following
1,rshelf001\CTRL+Z
-----------
-----------
When complete type 'edlin'
-----------
-----------

C:\FileNet\devlib2\stacks>edlin Links.txt
End of input file
*
```

At this point the system is waiting for input. For each shelf, you will need to type in the following command.

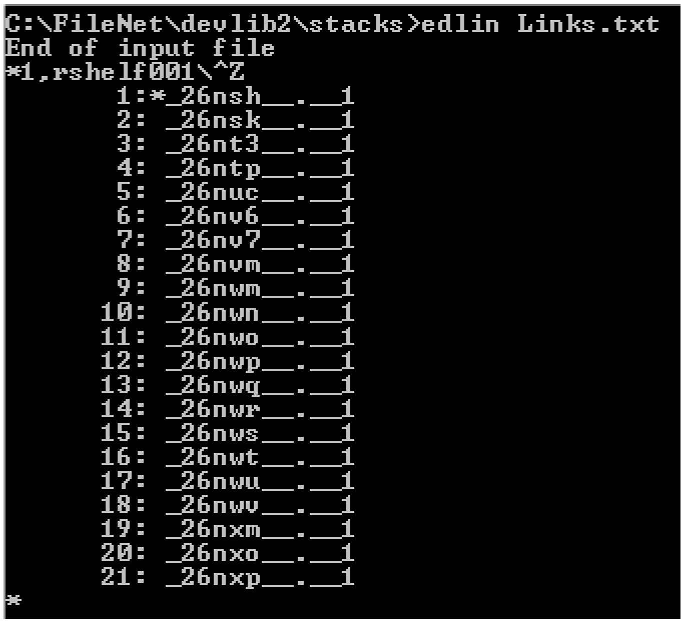
`1,rshelf001\CTRL+Z` *Note that for each subsequent shelf or archive, repeat the process but replace the shelf001 in the previous command for the name of the other shelf or archive.*

The important part of the command above is **`shelf001\`**. The command will replace all instances of **`shelf001\`** with an empty space. We want to end up with a list of files without respect to the shelf or directory they are in. This is critical to the success of the next step. Type in everything between the double quotes without typing in the double quotes; "`1,rshelf001\`" after typing in the preceding text, simultaneously press the control key and the letter 'Z'. The resulting command should look like the input below. If so, press the enter key.



```
C:\FileNet\devlib2\stacks>edlin Links.txt
End of input file
*1,rshelf001\^Z
```

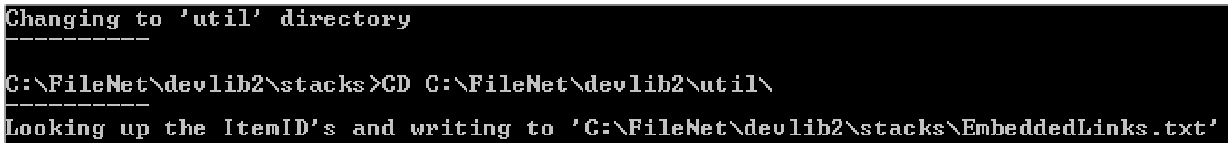
If you entered the command correctly you will see something resembling the image below.



```
C:\FileNet\devlib2\stacks>edlin Links.txt
End of input file
*1,rshelf001\^Z
       1:*_26nsh__.__1
       2: _26nsk__.__1
       3: _26nt3__.__1
       4: _26ntp__.__1
       5: _26nuc__.__1
       6: _26nv6__.__1
       7: _26nv7__.__1
       8: _26nvm__.__1
       9: _26nwm__.__1
      10: _26nwn__.__1
      11: _26nwo__.__1
      12: _26nwp__.__1
      13: _26nwq__.__1
      14: _26nwr__.__1
      15: _26nws__.__1
      16: _26nwt__.__1
      17: _26nwu__.__1
      18: _26nwv__.__1
      19: _26nxm__.__1
      20: _26nxo__.__1
      21: _26nxp__.__1
*
```

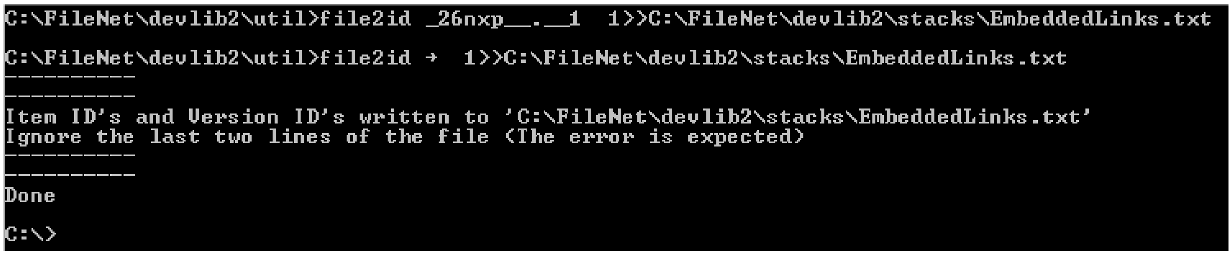
When you get to this point type the following command.
`edlin`

This will close the file you just edited and continue the batch operation. The next thing you should see is this.



```
Changing to 'util' directory
----------

C:\FileNet\devlib2\stacks>CD C:\FileNet\devlib2\util\
----------
Looking up the ItemID's and writing to 'C:\FileNet\devlib2\stacks\EmbeddedLinks.txt'
```

This will be followed by a lot of information rolling down the screen as the system resolves each Item ID and Version ID based on the encrypted file names in the 'Links.txt' file. When complete you will have an additional file called 'EmbeddedLinks.txt' with the Item ID's and Version ID's of the files with embedded links.



```
C:\FileNet\devlib2\util>file2id _26nxp__.__1  1>>C:\FileNet\devlib2\stacks\EmbeddedLinks.txt

C:\FileNet\devlib2\util>file2id →  1>>C:\FileNet\devlib2\stacks\EmbeddedLinks.txt
----------
----------
Item ID's and Version ID's written to 'C:\FileNet\devlib2\stacks\EmbeddedLinks.txt'
Ignore the last two lines of the file (The error is expected)
----------
----------
Done

C:\>
```

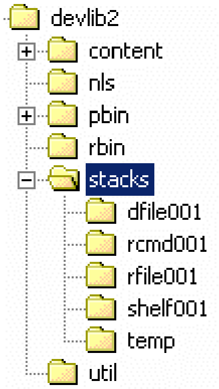
You can now locate the documents by searching on the Item IDs in the EmbeddedLinks.txt file.

Manual Process

For this process we will go through the same process as the batch file, but we will execute each step manually. This is especially good for those among us who absolutely must know what is really happening.

Finding the documents

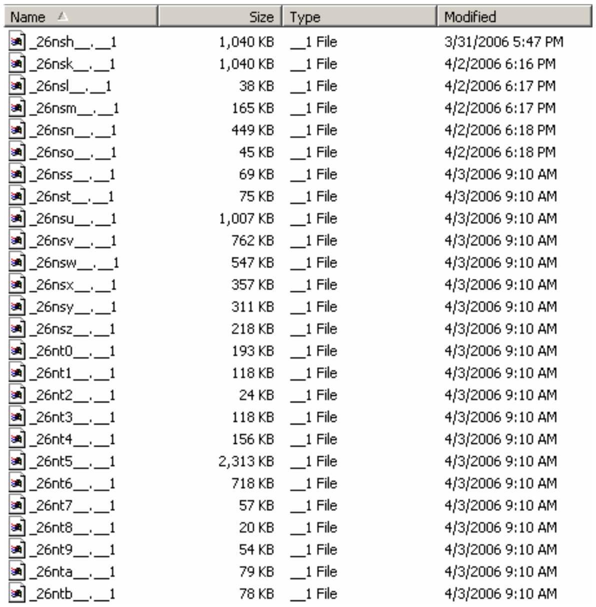
To find the documents with the embedded links we will do a search in each directory where the files are located. This means that if there are multiple storage managers, this step will need to be repeated for each storage manager. All of the document files we are interested in are located in directories under the stacks directory. The directory tree will look something like **Figure 1** below.



```
devlib2
  content
  nls
  pbin
  rbin
  stacks
    dfile001
    rcmd001
    rfile001
    shelf001
    temp
  util
```

Figure 1

In Figure 1 above ***devlib2*** represents the Content Services library. The directory structure under it is the standard Content Services server directory structure. The ***stacks*** directory contains the directories where the actual content files are stored. The ***dfile001*** directory contains the first archive repository. The ***rcmd001*** directory would contain descriptions of requested versions of documents to be reclaimed from archive. The ***rfile001*** directory temporarily stores versions immediately after they are reclaimed. The directory we are most concerned with here is the ***shelf001*** directory. It is where the online content files are stored. The files are all stored with encrypted file names. See **Figure2** below.

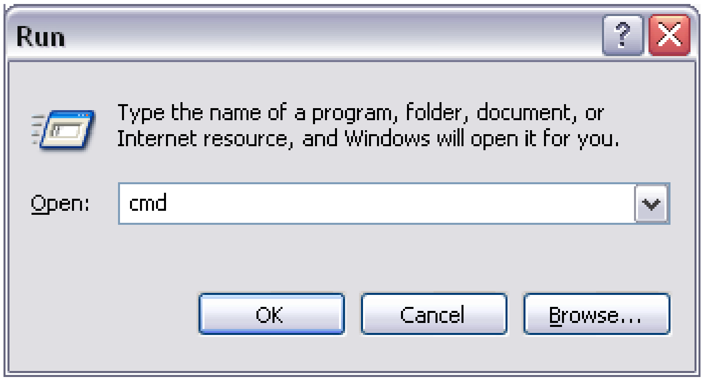


Name	Size	Type	Modified
_26nsh__.__1	1,040 KB	__1 File	3/31/2006 5:47 PM
_26nsk__.__1	1,040 KB	__1 File	4/2/2006 6:16 PM
_26nsl__.__1	38 KB	__1 File	4/2/2006 6:17 PM
_26nsm__.__1	165 KB	__1 File	4/2/2006 6:17 PM
_26nsn__.__1	449 KB	__1 File	4/2/2006 6:18 PM
_26nso__.__1	45 KB	__1 File	4/2/2006 6:18 PM
_26nss__.__1	69 KB	__1 File	4/3/2006 9:10 AM
_26nst__.__1	75 KB	__1 File	4/3/2006 9:10 AM
_26nsu__.__1	1,007 KB	__1 File	4/3/2006 9:10 AM
_26nsv__.__1	762 KB	__1 File	4/3/2006 9:10 AM
_26nsw__.__1	547 KB	__1 File	4/3/2006 9:10 AM
_26nsx__.__1	357 KB	__1 File	4/3/2006 9:10 AM
_26nsy__.__1	311 KB	__1 File	4/3/2006 9:10 AM
_26nsz__.__1	218 KB	__1 File	4/3/2006 9:10 AM
_26nt0__.__1	193 KB	__1 File	4/3/2006 9:10 AM
_26nt1__.__1	118 KB	__1 File	4/3/2006 9:10 AM
_26nt2__.__1	24 KB	__1 File	4/3/2006 9:10 AM
_26nt3__.__1	118 KB	__1 File	4/3/2006 9:10 AM
_26nt4__.__1	156 KB	__1 File	4/3/2006 9:10 AM
_26nt5__.__1	2,313 KB	__1 File	4/3/2006 9:10 AM
_26nt6__.__1	718 KB	__1 File	4/3/2006 9:10 AM
_26nt7__.__1	57 KB	__1 File	4/3/2006 9:10 AM
_26nt8__.__1	20 KB	__1 File	4/3/2006 9:10 AM
_26nt9__.__1	54 KB	__1 File	4/3/2006 9:10 AM
_26nta__.__1	79 KB	__1 File	4/3/2006 9:10 AM
_26ntb__.__1	78 KB	__1 File	4/3/2006 9:10 AM

Figure 2

In order to identify the encrypted files which contain embedded links we will recursively search all of the subdirectories of ***stacks*** for documents which contain the word "*doccontent.asp*". We will do this by using the command line utilities included with the operating system. One might ask why use command line tools when the modern world uses graphical tools like Windows Explorer. The reason is simple, we want to send the results of the search to a file. We will use this file in subsequent steps. Without fancy programming we can easily accomplish this at the command line. We could use the searching capabilities of Windows Explorer but we will then be limited to viewing the results on the screen and not have the luxury of feeding them to the next step. All of the following steps need to be executed on the storage manager server. If all the Content Services components are on a single machine then this is the Content Services server. You will need to know where the stacks directory is located.

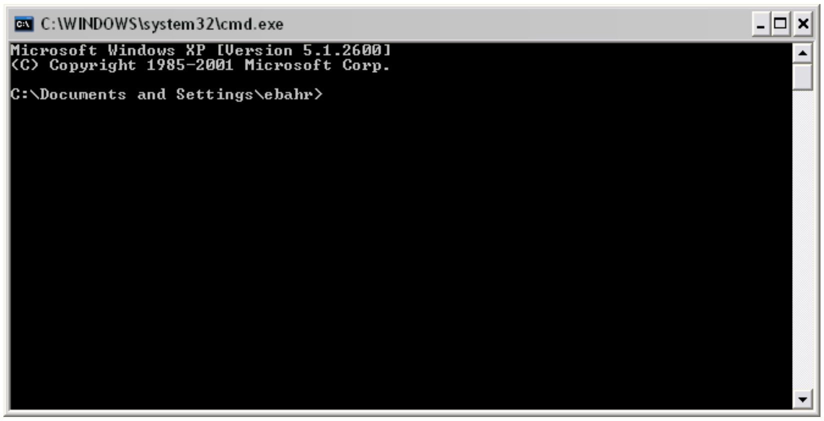
Open the command prompt. (Start -> Run -> Type "cmd" -> click OK or press enter.)



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

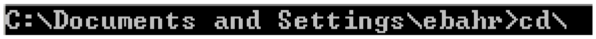
Open: cmd

Once at the command line you will then need to change directories.



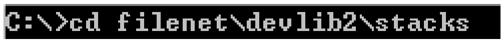
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ebahr>
```

Change directories to the stacks directory. For the purpose of this example we will assume the stacks directory is at C:\FileNet\devlib2\stacks. First we will get to the root directory by typing "`CD\`".

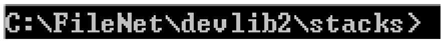


```
C:\Documents and Settings\ebahr>cd\
```

Then we will go from the root to the stacks directory by typing CD filenet\devlib2\stacks

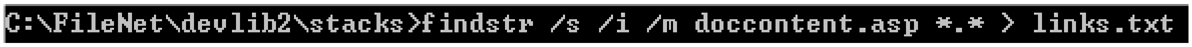


```
C:\>cd filenet\devlib2\stacks
```



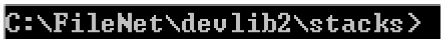
```
C:\FileNet\devlib2\stacks>
```

Now that we are at the stacks directory we will perform a recursive search for documents that contain the text "doccontent.asp". using the same command, we will redirect the results of that search to a file called "links.txt". To do this we will type "`findstr /s /i /m doccontent.asp *.* > links.txt`"



```
C:\FileNet\devlib2\stacks>findstr /s /i /m doccontent.asp *.* > links.txt
```

Since we have redirected the output to the file links.txt, we will not see the results on the screen. Instead we will return to the prompt below.

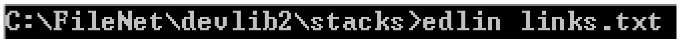


```
C:\FileNet\devlib2\stacks>
```

The file links.txt will contain one line for each file found. Each line will begin with the name of the directory in which the file was found. Following the directory name will be the encrypted file name. An example line is shown below.

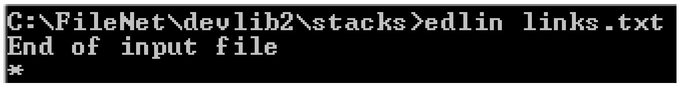
```
shelf001\_26y2s__.__2
```

The next step will be to edit the file to contain only the encrypted file names. You can do this with the text editor of your choice. The instructions included below utilize the “edlin” command utility as used in the batch file. This can be accomplished with notepad or another tool as well. At the same command prompt we will type “`edlin links.txt`”



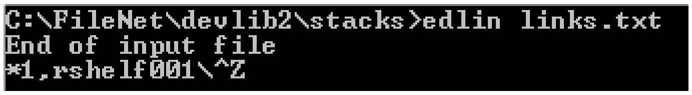
```
C:\FileNet\devlib2\stacks>edlin links.txt
```

You will see the following results…



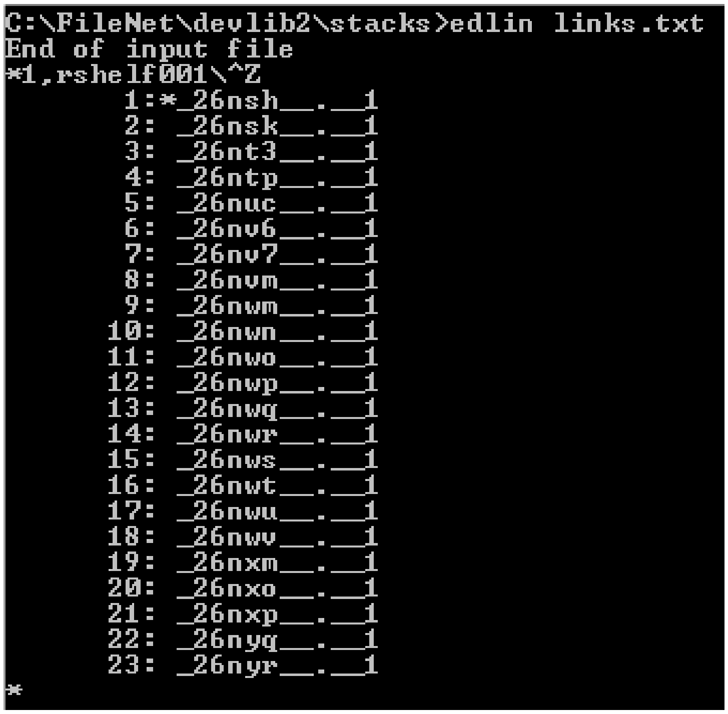
```
C:\FileNet\devlib2\stacks>edlin links.txt
End of input file
*
```

At the * prompt, type the following command “`1,rshelf001\`” and simultaneously press the Control Key and the letter ‘Z’. It will appear in the command prompt as below. If so then press enter.



```
C:\FileNet\devlib2\stacks>edlin links.txt
End of input file
*1,rshelf001\^Z
```

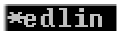
The results will look something like this.



```
C:\FileNet\devlib2\stacks>edlin links.txt
End of input file
*1,rshelf001\^Z
       1:*_26nsh__.__1
       2: _26nsk__.__1
       3: _26nt3__.__1
       4: _26ntp__.__1
       5: _26nuc__.__1
       6: _26nv6__.__1
       7: _26nv7__.__1
       8: _26nvm__.__1
       9: _26nwm__.__1
      10: _26nwn__.__1
      11: _26nwo__.__1
      12: _26nwp__.__1
      13: _26nwq__.__1
      14: _26nwr__.__1
      15: _26nws__.__1
      16: _26nwt__.__1
      17: _26nwu__.__1
      18: _26nwv__.__1
      19: _26nxm__.__1
      20: _26nxo__.__1
      21: _26nxp__.__1
      22: _26nyq__.__1
      23: _26nyr__.__1
*
```

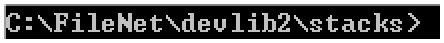
If you have additional shelves or directories that need to be stripped from the front of the file names, repeat the command once for each different directory. Substitute the “`shelf001`” in “`1,rshelf001\^Z`” with the name of the other directory. For example to repeat for shelf002 you would type “`1,rshelf002\^Z`”. Repeat this step for each directory found in the search.

At this point simply type “`edlin`” and press enter.



```
*edlin
```

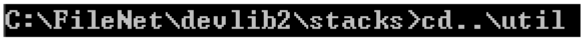
You will be returned to the command prompt.



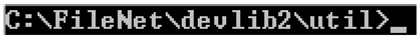
```
C:\FileNet\devlib2\stacks>
```

Now we have a file with the file names only. The next step is to take the encrypted file names and derive from them the actual Item and Version ID’s of the documents. We will utilize the FileNet utility “**file2id**”. This tool takes the encrypted file name as an argument and returns the Item and Version ID. We can utilize command line techniques to both avoid having to execute this utility once for each file found as well as to write the results all to a single text file. To do this we will use a slightly more advanced command line technique. With one line of instructions we will loop through each file listed in links.txt and resolve the IDs and write them all to a file called “embedded.txt”

First we need to change directories to the util directory. Type “`cd..\util`” and press enter.



```
C:\FileNet\devlib2\stacks>cd..\util
```

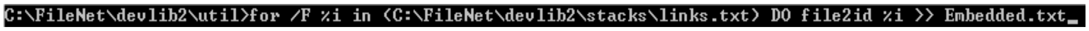


```
C:\FileNet\devlib2\util>_
```

Now type in the command below.

“`for /F %i in (C:\filenet\devlib2\stacks\links.txt) DO file2id %i >> embedded.txt`”

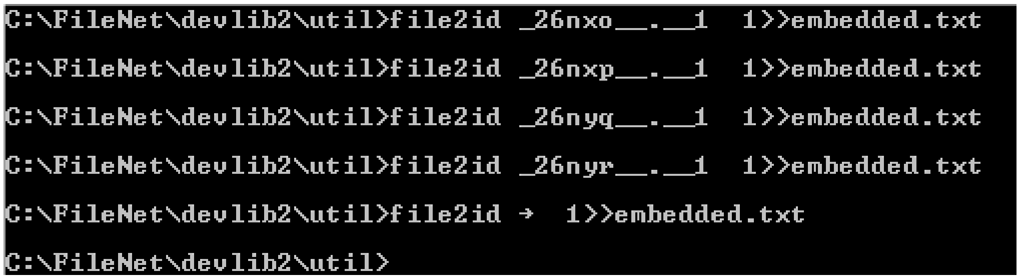
Make sure the directory you specify for links.txt is correct for your machine.



```
C:\FileNet\devlib2\util>for /F %i in (C:\FileNet\devlib2\stacks\links.txt) DO file2id %i >> Embedded.txt_
```

When you are sure that you have typed the command as shown, press Enter.

Your results will scroll down the screen and you will end with something resembling the output shown below.



```
C:\FileNet\devlib2\util>file2id _26nxo__.__1  1>>embedded.txt

C:\FileNet\devlib2\util>file2id _26nxp__.__1  1>>embedded.txt

C:\FileNet\devlib2\util>file2id _26nyq__.__1  1>>embedded.txt

C:\FileNet\devlib2\util>file2id _26nyr__.__1  1>>embedded.txt

C:\FileNet\devlib2\util>file2id →  1>>embedded.txt

C:\FileNet\devlib2\util>
```

Now if you open the embedded.txt file you should see something that looks like this.

```
    Item = 003670188, Version = 1
    Item = 003670189, Version = 1
    Item = 003670226, Version = 1
    Item = 003670227, Version = 1
    Item = 0-4206872, Version = 8742X79
    Validation Failed: Filename =
```

Disregard the last two lines. They are a result of the fact that the previous “edlin” editing operation on the “links.txt” file left the last line of the file as a carriage return. You now have a file with the list of documents containing the text “doccontent.asp” listed by Item and Version ID. The batch file process detailed in the previous section does all of the manual steps in exactly the same sequence. Depending on how many servers you need to check files in, you may decide that the batch version is a simpler method.

Modifying the Embedded Links

This solution involves identifying the documents which contain the embedded links and then modifying the links so they point to the correct location. The modification can be done either manually or automated. The automation of modifying embedded links is beyond the scope of this document as there are virtually unlimited variations on files types and formatting. There are we will focus on here is how to identify the documents which contain the embedded links.

Deleting the Embedded Links

This solution involves identifying the documents which contain the embedded links and then deleting the links so there are no references to other documents.

Appendix A

Batch File for use on Windows Storage Managers

```
@Echo Off
SETLOCAL
;
;
SET LIBRARY=C:\FileNet\devlib2\
;
;
SET UTIL=%LIBRARY%util\
SET STACKS=%LIBRARY%stacks\
SET LINKS=%STACKS%Links.txt
SET EMBEDDEDLINKS=%STACKS%EmbeddedLinks.txt
DEL %EMBEDDEDLINKS%
@Echo On
::@ECHO %LIBRARY%
::@ECHO %STACKS%
::@ECHO %UTIL%

@ECHO ----------
@Echo Identifying Embedded Web Services URL Links
@ECHO ----------
@Echo Changing directory to stacks
CD %STACKS%
@ECHO ----------
@Echo Finding documents with 'doccontent.asp' and writing list to file '%LINKS%'
@ECHO ----------
findstr /s /i /m doccontent.asp *.* > Links.txt
@Echo Find complete
@ECHO ----------
@ECHO ----------
@ECHO Editing Links file to remove shelf number
@ECHO At the prompt type the following
@ECHO 1,rshelf001\CTRL+Z
@ECHO ----------
@ECHO ----------
@ECHO When complete type 'edlin'
@ECHO ----------
```

```
@ECHO ----------
edlin Links.txt
@Echo Changing to 'util' directory
@ECHO ----------
CD %UTIL%
@ECHO ----------
@Echo Looking up the ItemID's and writing to '%EMBEDDEDLINKS%'
@ECHO ----------
@ECHO ----------
@ECHO ----------
for /F %%i in (%LINKS%) DO file2id %%i >> %EMBEDDEDLINKS%
@ECHO ----------
@ECHO ----------
@ECHO Item ID's and Version ID's written to '%EMBEDDEDLINKS%'
@ECHO Ignore the last two lines of the file (The error is expected)
@ECHO ----------
@ECHO ----------
@Echo Done
```