Understanding the ACIF Input Exit for Content Manager OnDemand for Multiplatforms

Last updated: March 15, 2017

Written for users of Content Manager OnDemand for Multiplatforms, this article explains why you might use the ACIF input exit, when the exit is called, why certain types of problems occur, and describes a strategy for creating correct indexing parameters when using an input exit. It also explains how to use ACIF to test your own input exit.

Introduction

Content Manager OnDemand for Multiplatforms is a vital piece of the Content Manager family. It processes the print output of application programs, extracts index fields from the data, stores the index information in a relational database, and stores one or more copies of the data on storage media. Content Manager OnDemand provides the capability to view, print, email, and fax documents, and fully integrates the capabilities of Advanced Function Presentation (AFP).

AFP Conversion and Indexing Facility (ACIF) is a powerful tool used by Content Manager OnDemand for indexing the print data streams of application programs. In this article, we will discuss the use of ACIF with an *input exit*. An input exit is a user exit that enables you to add, delete, or modify records in a Content Manager OnDemand input file.

Some customers encounter problems loading data into Content Manager OnDemand when their indexing parameters contain an input exit. The use of an input exit is indicated by the keyword INPEXIT in the ACIF parameters. The problems include errors from ACIF, such as "*Trigger not found*," or errors from the ARSLOAD program, such as "*'Field name 'Account' found in the index file does not match any expected field*" or "*Unable to convert 'Report' to a date*."

Inevitably, these types of errors are caused by the user specifying incorrect indexing parameters.

Why use an ACIF input exit?

The input exit is used to modify the input before it is indexed and loaded into Content Manager OnDemand. Here are some of the most common reasons why you would want to use an input exit:

- The input is ASCII line data and you want to convert it to EBCDIC line data and then convert it to AFP. AFP can contain ASCII line data, but most users prefer EBCDIC line data in order to use the multitude of coded fonts which are only available for EBCDIC data. Converting ASCII to EBCDIC is so common that Content Manager OnDemand provides an ACIF input exit, called *apka2e*, in order to do this.
- The input is ASCII line data and does not contain valid ANSI carriage controls. It may contain ASCII carriage return (X'0D') and form feed (X'0C') characters, or no control characters. Under most circumstances, ACIF is unable to process a file unless it contains either valid ANSI or machine carriage controls. The *asciinp* exit, provided with ACIF, will remove the X'0D' and X'0C' characters and replace them with valid ANSI carriage controls. The *asciinpe* exit, also provided, will do the same thing, and, in addition, convert the data to EBCDIC. NOTE: Starting with Content Manager OnDemand version 9.5.0.3, ACIF supports the FORMFEED parameter, which allows ACIF to process files that use the form feed character (X'0C') to designate a new page.
- The data does not contain any characters which could be used as a trigger. ACIF locates the trigger and then finds the index fields at given offsets from the trigger. For a description of this process, refer to the IBM Content Manager OnDemand Indexing Reference. The trigger should occur at regular intervals in the data, so that the data can be broken up into documents (called 'groups' in Content Manager OnDemand) of reasonable size or logical separation to be loaded into Content Manager OnDemand. Sometimes, the data is so complicated and non-uniform that it is impossible to find a character string (or even a single character) which could be used as a trigger. An input exit can insert new lines into the input file, or it can add characters to an existing line, which can be used as a trigger during the indexing process. If you do not want these lines or characters to be stored in Content Manager OnDemand, they can be removed by an ACIF output exit. In these cases, the exits would be custom-written for a particular type of input file.
- The input contains pages or lines that you would like to remove, and not store in Content Manager OnDemand. The page could be a banner page, or other pages or lines that are superfluous or could interfere with indexing the data. Like the previous scenario, the input exit would be custom-written by you or IBM Services.

When and how the exit is called

The input exit is called for each record, one record at a time, after that record has been read from the input file, but before the record is passed to the ACIF indexer. After the input exit processes the record, it is passed to the ACIF indexer, which accumulates the records needed to find the triggers and fields for a group. The records are eventually written to the output file, after optionally being converted to AFP. The important point to note here is that the input exit is called before the records are indexed. See Figure 1.

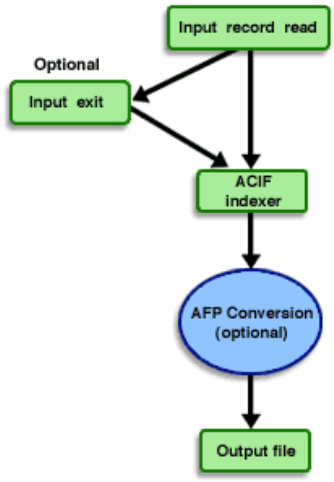

Figure 1. Calling the input exit

The fact that the input exit manipulates the data before it is indexed is an important point. However, you may not have considered these changes when you created the indexing parameters for the data. For example, assume that you are working with an ASCII file. You would like to use the new page carriage control character as the trigger and have an index named **rdate**. If you did not use the apka2e exit, you would use the values from the ASCII column of Table 1 in your indexing parameters. However, if you changed your data from ASCII to EBCDIC by using the apka2e exit, you would need to use the values from the EBCDIC column of Table 1 in your indexing parameters instead.

	ASCII	EBCDIC	
cctype	Z	A	
cpgid	850	500	
trigger	x'31'	x'F1'	/* '1' */
index1	x'7264617465'	x'998481A385'	/* 'rdate` */

Table 1. Comparison of indexing parameters for ASCII and EBCDIC data

If you were to load the ASCII file into the graphical indexer and use it to create indexing parameters to use with the apka2e exit, the graphical indexer would generate the values shown in the ASCII column, and they would be wrong.

Note that the following message appears when you start the graphical indexer:

```
<em>Warning!  This data should match the data being loaded.</em>
```

Unfortunately, many people do not heed this warning.

A strategy for generating correct indexing parameters

The following example shows how to run ACIF outside of the Content Manager OnDemand application with the *convert = no* parameter, and how to load the output file generated by ACIF into the graphical indexer.

Example - Process an ASCII file with apka2e

Step 1 - Create an ACIF parameter file to preprocess the input file.
Create a text file that contains the following ACIF parameters:

```
Cctype = a
Convert = no
Inputdd = <path> <ascii filename>
Outputdd = <path> <output filename>
Inpexit = <path> apka2e
```

Replace the values `<path>`, `<ascii filename>`, and `<output filename>` with valid path and filename specifications for your operating system. Save the file and name the file `test`.

Note the following about the `test` parameter file:

- Unless specified otherwise, ACIF assumes that the input file consists of a stream of records that are separated by the new line character x'0A'. The default value for this is `fileformat=stream,(newline=x'0A')`. If your file consists of records in some other format, you must specify the FILEFORMAT parameter. See the IBM Content Manager OnDemand Indexing Reference for more information about the FILEFORMAT parameter.
- If you specify `cctype = z`, the apka2e exit will not convert the carriage controls to EBCDIC. Use `cctype = a` so that the carriage controls will be converted along with the rest of the data in the file.

Step 2 - Run ACIF outside of the Content Manager OnDemand application. Enter the following on the command line:

```
arsacif  parmdd=<path>test
```

where `test` is the name of the parameter file that you created in Step 1.

Because you did not specify a message file in the `test` parameter file, ACIF will send its messages to the screen. ACIF will create an output file that will be identical to the input (because of the convert = no parameter), except that the output file will contain the changes that are made by the exit.

Step 3 - Verify that the exit made the correct changes to the input file. View the output file with either a text or hexadecimal editor to see the changes that were made by the exit.

Step 4 - Load the output file into the graphical indexer.

Start the Content Manager OnDemand administrative client and log on to a server. Then click the **Report Wizard** button on the toolbar to start the Report Wizard. From the Sample Data window, click **Select Sample Data**. From the Open dialog box, select the output file that was created in Step 2. The Report Wizard opens the file in the graphical indexer and displays the Indexer Properties dialog box. On the Data Format page, specify the attributes of the file, such as carriage control type, file format, and code page so that the graphical indexer can display the file properly.

After specifying the attributes of the file, click **OK** to close the Indexer Properties dialog box and switch to the graphical indexer. Create the indexing triggers and fields. For information about using

the graphical indexer, see IBM Content Manager OnDemand for Multiplatforms Administrator's Guide. By using the output file created by Step 2 in the graphical indexer, the data will match what the ACIF indexer will see and process.

Step 5 - Load the original file into Content Manager OnDemand.

After creating the indexing parameters in the Content Manager OnDemand application, run the ARSLOAD program to load the data into the system. Discard the ACIF parameter file that you created in Step 1 and the output file that was generated in Step 2. Process the original ASCII file with the ARSLOAD program. Because you have created the indexing parameters by using the data that was changed by the input exit, the data should load with no problems.

Test your own exit

You can use Step 1 and 2 above to test any input exit that you write yourself. ACIF will apply the exit to the data and create an output file, which you can then examine to determine if the exit is doing what you want it to do. To illustrate the process, here is a more complicated indexing example.

We have an ASCII file that we would like to index, and not convert to AFP. The input file is shown loaded into the Graphical Indexer in Figure 2.

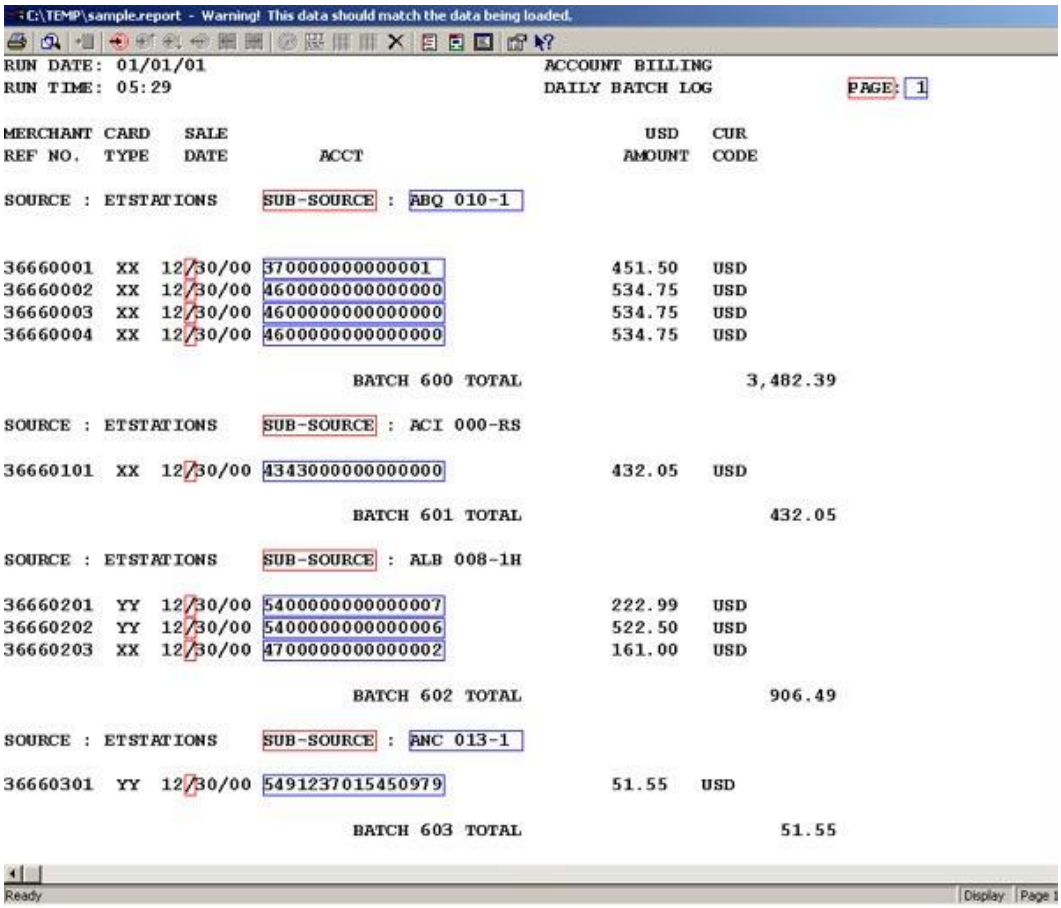

Figure 2. Input file in the graphical indexer

We have defined one group trigger, "PAGE", and two floating triggers, "SUB-SOURCE" and "/". We would like to pick up the page number and use it to break our documents. We would also like to pick up the string following "SUB-SOURCE", but ONLY if the last part of the string is numeric. Therefore, we would like to collect "ABQ 010-1" and "ANC 013-1", but not "ACI 000-RS" and "ALB 008-1H". In addition, we would like to collect the account numbers ONLY if they follow a SUB- SOURCE value which we have collected. Therefore, we would like to collect the four account numbers following "ABQ 010-1", and the single account number following "ANC 013-1", but not the others.

The graphical indexer shows us, in Figure 2, the triggers we have defined in the red boxes, and the fields we have defined in the blue boxes. Here are the indexing parameters which, as you can see in Figure 2, are NOT giving us exactly what we want:

```
TRIGGER1=*,77,X'50414745',(TYPE=GROUP)   /* PAGE  */
TRIGGER2=*,25,X'5355422D534F55524345',(TYPE=FLOAT) /* SUB-SOURCE */
TRIGGER3=*,18,X'2F',(TYPE=FLOAT)      /* /       */
FIELD1=0,82,2,(TRIGGER=1,BASE=0)
FIELD2=0,38,10,(TRIGGER=2,BASE=0,MASK='@@@=###=#%')
FIELD3=0,25,16,(TRIGGER=3,BASE=0)
```

The problem is how to distinguish between the account numbers that we want to collect and those that we do not want to collect. There is no unique character in the records containing the account numbers we want that we can use for our floating trigger.

The solution is to use an input exit to mark the lines that we want with a value that ACIF can use as a trigger. So instead of using "SUB-SOURCE" and "/" as triggers, we will use new values, which the input exit will insert. After indexing, we will use the ACIF output exit to remove the inserted values, so that they will not appear in our output.

Here is a listing of the input exit, named `docexample`, which we will use:

```
/*****************************************************************/
/*                                                               */
/* MODULE NAME: DOCEXAMPLE.C                                     */
/*                                                               */
/*                                                               */
/* SYNOPSIS:  inserts characters for indexing                    */
/* input file: sample.report                                     */
/* use docaxampleo.c as corresponding output exit to remove characters*/
/*                                                               */
/* COPYRIGHT:                                                    */
/*                                                               */
/* 5648-062,5765-140,5639-I27 (C) COPYRIGHT IBM CORPORATION 1993,2017 */
/* All Rights Reserved                                           */
/* Licensed Materials - Property of IBM                          */
/*                                                               */
/*                                                               */
/* US Government Users Restricted Rights - Use, duplication or   */
/* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  */
/*                                                               */
/*                                                               */
/* CHANGE ACTIVITY:                                              */
/*                                                               */
/* PROBNO  REL   DATE/AUTHOR DESCRIPTION                         */
/* ------ -----  ----------- -----------                         */
/*                                                               */
/*****************************************************************/
#include <string.h>
#include <ctype.h>
#include "apkexits.h"

char *recordptr;       /* pointer to record to be analyzed      */
int                    /* length of record to be analyzed       */
int
static int flag = 0;

long ACIF_EXPORT INPEXIT( INPEXIT_PARMS *parms)
{
  recordptr = parms->record;
  recordlen = parms->recordln;

  if( parms->eof == 'Y' )
    return(0);

  switch( flag )
  {
   case 0:
     if( recordlen > 45 )
     {
       if( strncmp( recordptr+24, "SUB-SOURCE", 10 ) == 0 )
       {
         if( isdigit( recordptr[45] ))
         {
           if( recordlen > 46 )
           {
             if( isdigit( recordptr[46] ) || isspace( recordptr[46] ))
               insert_trigger = 1;
             else
```

```
              insert_trigger = 0;
          }
          else
            insert_trigger = 1;
        }
        else
          insert_trigger = 0;
        if( insert_trigger )
        {
          /* mark the sub-source record with the trigger value */
          recordptr[21] = 'T';
          flag = 1;
        }
      }
    }
    parms->request = 0x00;
    break;
  case 1:
    if( recordlen > 45 )
    {
      if( recordptr[17] == '/' )
      {
        /* mark the account record with the trigger value */
        recordptr[41] = 'M';
        flag = 1;
      }
      else
      {
      if( strncmp( recordptr+42, "TOTAL", 5 ) == 0 )
        flag = 0;
      }
    }
    parms->request = 0x00;
 } /* end switch */
 return(0);
}
```

Listing 1. docexample input exit

The input exit named `docexample` examines each line of the input file looking for a numeric sub-source value. When it finds one, it marks the line with a 'T' in byte 21. ACIF will use this 'T' as a trigger during indexing. It also sets the variable "flag" to 1. Since flag is a static variable, its value will be preserved across successive calls to `docexample`. If flag is 1 when ACIF calls `docexample`, a numeric sub-source value has been found, so records with account numbers need to be marked with another value that can be used as a trigger. If an account record is found, the record is marked with an 'M' in byte 41. If the string "TOTAL" is found starting at byte 42, we have reached the end of that section of account numbers, so the variable flag is reset.

Here are the parameters to run ACIF outside of the Content Manager OnDemand application with the input exit:

```
cctype = z
convert = no
inputdd = <path>/sample.report
outputdd = <path>/<output filename>
inpexit = <path>/docexample
```

The output file generated is shown in Listing 2. The characters that have been added by the exit are shown in bold. Note: Some of the spacing has been changed so the document fits on the page.

```
RUN DATE: 01/01/01                              ACCOUNT BILLING
RUN TIME: 05:29                                 DAILY BATCH LOG PAGE:1


MERCHANT CARD   SALE                                  USD   CUR
REF NO.  TYPE   DATE        ACCT                    AMOUNT  CODE

SOURCE : ETSTATIONS T  SUB-SOURCE : ABQ 010-1


36660001  XX  12/30/00 370000000000001  M            451.50   USD
36660002  XX  12/30/00 4600000000000000 M            534.75   USD
36660003  XX  12/30/00 4600000000000000 M            534.75   USD
36660004  XX  12/30/00 4600000000000000 M            534.75   USD
                        BATCH 600 TOTAL               3,482.39

SOURCE : ETSTATIONS    SUB-SOURCE : ACI 000-RS

36660101  XX  12/30/00 4343000000000000             432.05   USD

                        BATCH 601 TOTAL                 432.05

SOURCE : ETSTATIONS    SUB-SOURCE : ALB 008-1H

36660201  YY  12/30/00 5400000000000007             222.99   USD
36660202  YY  12/30/00 5400000000000006             522.50   USD
36660203  XX  12/30/00 4700000000000002             161.00   USD

                        BATCH 602 TOTAL                 906.49

SOURCE : ETSTATIONS T  SUB-SOURCE : ANC 013-1

36660301  YY  12/30/00 5491237015450979 M            51.55   USD

                        BATCH 603 TOTAL                  51.55
```

Listing 2. Generated output file

Now we can load the output file into the graphical indexer to create the indexing parameters which will give us exactly what we want. See Figure 3.

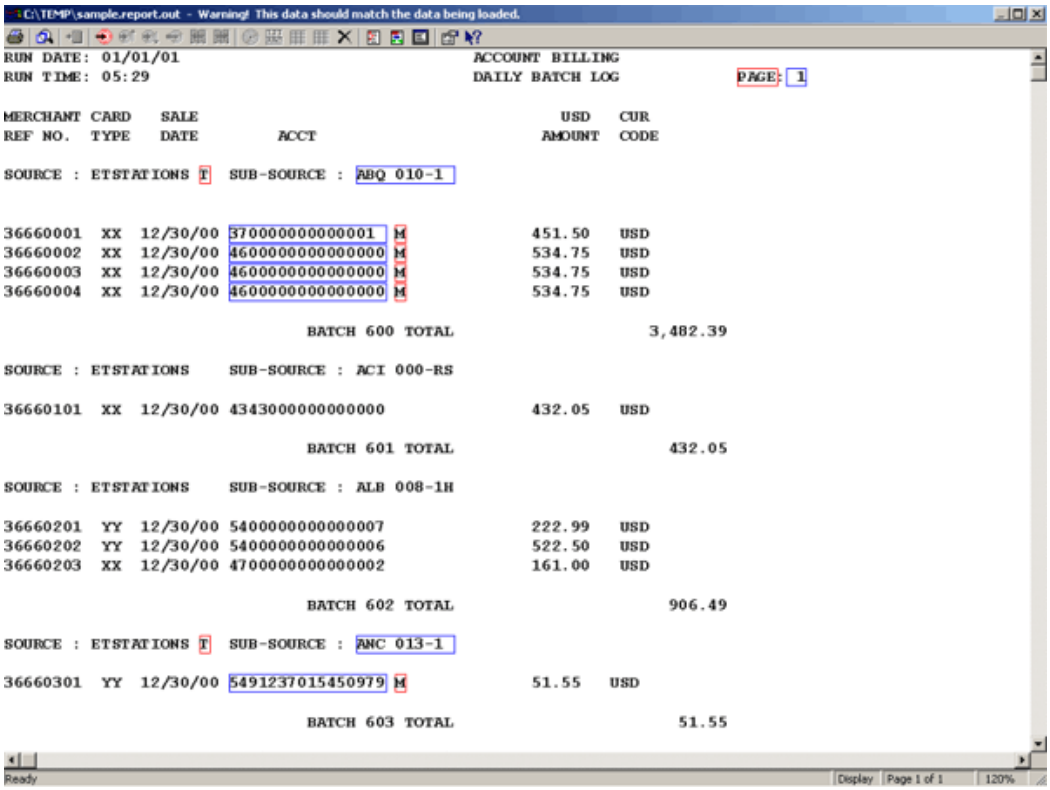

Figure 3. Output file in the graphical indexer

Listing 3 shows the indexing parameters created with the graphical indexer:

```
CC=YES
CCTYPE=Z
CONVERT=NO
CPGID=850
MCF2REF=CPCS
TRC=NO
FILEFORMAT=STREAM,(NEWLINE=X'0D0A')
TRIGGER1=*,77,X'50414745',(TYPE=GROUP) /* PAGE       */
TRIGGER2=*,22,X'54',(TYPE=FLOAT)       /* T          */
TRIGGER3=*,42,X'4D',(TYPE=FLOAT)      /* M           */
FIELD1=0,82,2,(TRIGGER=1,BASE=0)
FIELD2=0,38,10,(TRIGGER=2,BASE=0)
FIELD3=0,25,16,(TRIGGER=3,BASE=0)
INDEX1=X'70616765',FIELD1,(TYPE=GROUP,BREAK=YES) /* page */
INDEX2=X'7375625F736F75726365',FIELD2,(TYPE=GROUP,BREAK=NO) /* sub_source */
INDEX3=X'61636374',FIELD3,(TYPE=GROUP,BREAK=NO)             /* acct */
DCFPAGENAMES=NO
UNIQUEBNGS=YES
IMAGEOUT=ASIS
INDEXOBJ=GROUP
INDEXSTARTBY=1
INSERTIMM=NO
RESTYPE=NONE
```

Listing 3. Indexing parameters

In order to load this data into Content Manager OnDemand, there are two more steps. The first is to add this line to the indexing parameters:

```
INPEXIT=<path>docexample
```

The second is to use an output exit to remove the characters which we inserted with the input exit. Here is a listing of the output exit, named `docexampleo`, which we will use:

```
/*****************************************************************/
/*                                                               */
/* MODULE NAME: DOCEXAMPLEO.C                                    */
/*                                                               */
/* SYNOPSIS:  removes characters after indexing                  */
/*                                                               */
/* COPYRIGHT:                                                    */
/* 5648-062,5765-140,5639-I27 (C) COPYRIGHT IBM CORPORATION 1993,2017 */
/* All Rights Reserved                                           */
/* Licensed Materials - Property of IBM                          */
/*                                                               */
/* US Government Users Restricted Rights - Use, duplication or   */
/* disclosure restricted by GSA ADP Schedule Contract with IBM Corp. */
/*                                                               */
/* CHANGE ACTIVITY:                                              */
/*                                                               */
/*  PROBNO  REL                                                  */
/*               DATE/AUTHOR DESCRIPTION                         */
/* ------- ----- ----------- -----------                         */
                 24 March 2003
/*****************************************************************/
#include <string.h>
#include "apkexits.h"

char   *recordptr;          /* pointer to record to be analyzed      */
int    recordlen;           /* length of record to be analyzed       */

long ACIF_EXPORT OUTEXIT( OUTEXIT_PARMS *parms)
{
  recordptr = parms->record;
  recordlen = parms->recordln;
  parms->request = 0x00;
  if( recordlen > 45 )
  {
    if( strncmp( recordptr+24, "SUB-SOURCE", 10 ) == 0 )
      recordptr[21] = 0x20;
    else
    {
      if( recordptr[41] == 'M' )
        recordptr[41] = 0x20;
    }
  }
  return(0);
 }
```

Listing 4. docexampleo output exit

The output exit examines the records to see if they contain one of the inserted values. In the case of the sub-source record, it is easier to write a blank over the byte where an inserted value may occur, than to explicitly check for the value. In the case of the other records, the exit explicitly looks for the inserted value, overwriting it with a blank if it is found.

Like the input exit, the output exit can be tested to verify that it is working correctly, by running ACIF outside of the Content Manager OnDemand application, and then examining the output file.

Listing 5 shows the final indexing parameters to use with Content Manager OnDemand:

```
CC=YES
CCTYPE=Z
CONVERT=NO
CPGID=850
MCF2REF=CPCS
TRC=NO
FILEFORMAT=STREAM,(NEWLINE=X'0A')
TRIGGER1=*,77,X'50414745',(TYPE=GROUP)/* PAGE  */
TRIGGER2=*,22,X'54',(TYPE=FLOAT)     /* T      */
TRIGGER3=*,42,X'4D',(TYPE=FLOAT)     /* M      */
FIELD1=0,82,2,(TRIGGER=1,BASE=0)
FIELD2=0,38,10,(TRIGGER=2,BASE=0)
FIELD3=0,25,16,(TRIGGER=3,BASE=0)
INDEX1=X'70616765',FIELD1,(TYPE=GROUP,BREAK=YES) /* page */
INDEX2=X'7375625F736F75726365',FIELD2,(TYPE=GROUP,BREAK=NO) /* sub_source */
INDEX3=X'61636374',FIELD3,(TYPE=GROUP,BREAK=NO)           /* acct  */
DCFPAGENAMES=NO
UNIQUEBNGS=YES
IMAGEOUT=ASIS
INDEXOBJ=GROUP
INDEXSTARTBY=1
INSERTIMM=NO
RESTYPE=NONE
INPEXIT=<path>docexample
OUTEXIT=<path>docexampleo
```

Listing 5. Final indexing parameters

Note the following about the example in Listing 5:

- During the whole process, *convert=no* was used. If the data were to be converted to AFP, removing the inserted characters becomes much more complicated and beyond the scope of this simple example.

Conclusion

You should now have a better understanding of why you might use the ACIF input exit, when the exit is called, why certain types of problems occur, and how to create correct indexing parameters when using an input exit.

Disclaimer

This article contains sample code. IBM grants you ("Licensee") a non-exclusive, royalty free, license to use this sample code. However, the sample code is provided as-is and without any warranties, whether EXPRESS OR IMPLIED, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE THAT RESULT FROM YOUR USE OF THE SOFTWARE. IN NO EVENT WILL IBM OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF IBM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents.