

# ACIF Indexing with IBM® Content Manager OnDemand for Multiplatforms

## *Introduction*

IBM Content Manager OnDemand for Multiplatforms (OnDemand) is an archival/retrieval product that allows for the storing of a variety of datastreams. OnDemand can store any type of datastream, but primarily deals with line data or Advanced Function Presentation™ (AFP™) datastreams. Before data can be loaded into OnDemand, it is processed by the AFP Conversion and Indexing Facility (ACIF). ACIF is used to index the datastream (line data or AFP), which creates an index file of values extracted from the input file. ACIF can also convert line data to AFP or the line data can remain unchanged. Finally ACIF is used when the output datastream is AFP in order to collect any AFP resources (formdef, overlay, pageseg, etc). Because of the complexity of ACIF and the great number of ACIF parameters, we see many user errors that could be prevented with a little extra knowledge.

Each of the following eight sections presents a problem, diagnostic information and steps, and the solution.

## **1. Memory Allocation**

### **Problem:**

```
Message from ARSLOAD
Unable to allocate enough memory. File=arsadmin.c, Line=604
Loaded 0 rows into the database
```

### **Diagnosis:**

The input files involved were not large. ACIF completed with RC=0. Examining the index file we saw that it contained no Tag Logical Element (TLE) structured fields, and no group level Index Element (IEL) structured fields, only page level IELs. The way to tell the difference between group level and page level IELs is by running the AFPDMP tool with the -d option against the index file and examining the IELs. AFPDMP is a utility provided free by IBM Printing Systems and can be obtained from <http://www.printers.ibm.com/R5PSC.NSF/Web/tools>. AFPDMP will also show you if the index file contains any TLEs. The AFPDMP tool displays the AFP index file (which is a binary file) in a readable format.

Here is an example of a group level IEL displayed by AFPDMP:

```
...2 - IEL (Index Element)           SFI 006E D3B2A7 000000
    IEL: Object Byte Extent Triplet (57):
    IEL: - byte extent of object = 6226 (001852)
    IEL: Direct Byte Offset Triplet (2D):
    IEL: - byte offset = 170 (0000AA)
```

IEL: Object Structured Field Extent Triplet (59):  
 IEL: - num of structured fields in object = 40 (000028)  
 IEL: Object Structured Field Offset Triplet (58):  
 IEL: - structured field offset = 3 (000003)  
 IEL: Medium Map Page Number Triplet (56):  
 IEL: - sequence num of page = 1 (000001)  
 IEL: Fully Qualified Name Triplet (02):  
 IEL: - use = 0D (begin page group reference) <===== this indicates group level  
 IEL: - name = '8000-015-3024 00000001'  
 IEL: Fully Qualified Name Triplet (02):  
 IEL: - use = 8D (begin medium map reference)  
 IEL: - name = 'F28177 '  
 IEL: unrecognized triplet type (5e)  
 IEL: Process Level Triplet (81):  
 IEL: - process level = 97 (61)

And here is an example of a page level IEL displayed by AFPDMP:

...6 - IEL (Index Element) SFI 0054 D3B2A7 000000  
 IEL: Object Byte Extent Triplet (57):  
 IEL: - byte extent of object = 3417 (000D59)  
 IEL: Direct Byte Offset Triplet (2D):  
 IEL: - byte offset = 665 (000299)  
 IEL: Object Structured Field Extent Triplet (59):  
 IEL: - num of structured fields in object = 17 (000011)  
 IEL: Object Structured Field Offset Triplet (58):  
 IEL: - structured field offset = 7 (000007)  
 IEL: Medium Map Page Number Triplet (56):  
 IEL: - sequence num of page = 1 (000001)  
 IEL: Fully Qualified Name Triplet (02):  
 IEL: - use = 87 (begin page reference) <===== this indicates page level  
 IEL: - name = '00000001'  
 IEL: Fully Qualified Name Triplet (02):  
 IEL: - use = 8D (begin medium map reference)  
 IEL: - name = 'F28177 '  
 IEL: Process Level Triplet (81):  
 IEL: - process level = 3 (03)  
 IEL: unrecognized triplet type (83)

The fact that the index file contained no group IELs or TLEs implied that ACIF had not found any index values, although it had completed with RC=0. The most common reason that ACIF does not find any index values is because a TRIGGER parameter is specified incorrectly.

The problem occurred because of the following TRIGGER parameter:

TRIGGER2=0,90,X'3120',(TYPE=GROUP)

ACIF was looking for an ASCII '1' followed by a space. Looking at the input file in a text editor, we could see the '1' at the end of the line, in the correct column. However, looking at the input file in a **hexadecimal** editor, we could see that the '1' was the last character in the line, with no subsequent space. The '1' was followed by the stream delimiter X'0A', signifying the start of the next line. ACIF never found TRIGGER2, and because the INDEXSTARTBY parameter was set to a number larger than the number of pages in the file, ACIF did not report an error. As soon as we changed the INDEXSTARTBY parameter to 1 and reran the load, ACIF produced the following error message:

APK448 Indexing was requested, but neither 'TRIGGER1' nor any 'FIELD' was satisfied within the page

range specified by the INDEXSTARTBY parameter.

Since ACIF was not able to index the input file, it did not create any groups. When the files being loaded into OnDemand do not contain any groups, ARSLOAD may run out of memory looking for the beginning and ending of the groups.

**Solution:** The problem was fixed by changing TRIGGER2 in the following way:

```
TRIGGER2=0,90,X'31',(TYPE=GROUP)
```

## 2. Memory Allocation

### Problem:

```
Message from ARSLOAD
Unable to allocate enough memory. File=arsadmin.c, Line=604
Loaded 0 rows into the database
```

### Diagnosis:

Examining the data, we found that the value of the group-breaking field never changed. Here are the relevant ACIF parameters:

```
TRIGGER1 = *,1,X'F1',(TYPE=GROUP)
FIELD1 = 0,9,3
INDEX1 = X'D985979699A340D5A494828599', field1,(TYPE=GROUP,BREAK=YES)
```

If the value of FIELD1 never changes, the whole document will be one group. It is important to remember that ACIF starts a new group when the value of the field **changes**, if that field is used for an index defined as TYPE=GROUP, BREAK=YES. If the value never changes, ACIF may run out of storage processing the file, or ARSLOAD may run out of memory trying to load the data.

### Solution:

- For this problem there are several solutions, depending on the format of your data:
- Use the ACIF parameter GROUPMAXPAGES (can cause problems, see #7 below)
  - Use OnDemand Large Object Support in the application definition
  - Use a different field to break the groups, one that changes when you would like a new group to start

## 3. Memory Allocation

### Problem:

ACIF message

APK408 A VIRTUAL STORAGE REQUEST WAS UNSUCCESSFUL - REQUEST  
SIZE 80 RETURN CODE 8

### **Diagnosis:**

The data was line data being converted to AFP. There were no carriage controls in the data so ACIF thought that the whole file was a single page. ACIF holds a page at a time in memory until it processes the beginning of the next page, then it writes the held page to the output file. In this case, the "page" turned out to be 225 megabytes because there were no carriage controls! ACIF gave up.

### **Solution:**

Use the OnDemand provided asciinp user exit or a user exit to insert carriage controls, or use the ACIF LINECNT parameter if all the pages have the same number of lines

## ***4. Documents having incorrect page breaks***

### **Problem:**

The data is line data being converted to AFP. ACIF completes RC=0 and the data loads, but when viewed, successive groups have the page break in the wrong place. As one proceeds through the groups, some lines from the top of the page are missing, and lines from the next page have been included in the current page, so the page break slowly drifts down the pages.

### **Diagnosis:**

Here are the ACIF parameters that were used:

```
CC=NO
CCTYPE=A
CONVERT=YES
CPGID=500
MCF2REF=CF
TRC=NO
FILEFORMAT=RECORD
TRIGGER1=*,143,X'D781878540D5964B
FIELD1=0,55,45,(TRIGGER=1,BASE=0)
INDEX1=X'D9D7E3D5C1D4C5', FIELD1,(TYPE=GROUP, BREAK=YES)
DCFPAGENAMES=NO
UNIQUEBNGS=YES
IMAGEOUT=ASIS
INDEXOBJ=ALL
INDEXSTARTBY=99
INSERTIMM=NO
FORMDEF=F1PPHR
PAGEDEF=P1PPHR
USERLIB=\acifreslib2
```

RESTYPE=none

The customer had run ACIF with CC=NO when it should have been YES. Therefore, ACIF used the "number of lines/page" defined in the PAGEDDEF for the AFP conversion, which is a constant. Since the actual data did not have the same number of lines per page, the page break drifted around in the pages.

**Solution:**

The data had to be reloaded with CC=YES.

## ***5. Ever Seen These?***

**Problem:**

ACIF error message 104: LIN Structured Field is not allowed

**Solution:**

There is no LIN structured field! That message means that ACIF was trying to read line data in an AFP file; the problem is most likely a bad AFP structured field length.

**Problem:**

ACIF error message 104: EOF Structured Field is not allowed

**Solution:**

There is no EOF structured field. This message means that a resource or the print file has ended prematurely. The file in error has probably been truncated for some strange reason.

## ***6. One of the Most Common Problems***

**Problem:**

Message from ARSLOAD  
2 fields submitted, 3 expected (or, 0 fields submitted, 3 expected)

**Diagnosis:**

There are 2 common scenarios that produce this type of message.

## **Scenario 1**

There are two places to define default values, which can be used in the loading process: the Application Group or the ACIF field parameter. They are used in different ways.

If a field is not found in the data because the field is based off a floating trigger and the trigger was not found, then there is no TLE for it and ARSLOAD will use the default value, which was defined in the Application Group during the loading process. If no default value was defined, loading may fail with the above message. (If the trigger/triggers were found, but the record was not long enough to contain the field, then ACIF will create a TLE for that field using the default value from the ACIF Field parameter. If the record is not long enough to contain the field, and no default value has been specified in the ACIF field parameters, ACIF will fail with message 449: Index Fields Reference Outside of the Record.)

## **Solution:**

If there is a chance that a field may not exist, define a default value for it in the Application Group.

## **Scenario 2**

Understanding how ACIF processes fully composed AFP files

Fully composed AFP files contain Begin Named Page Group (BNG), End Named Page Group (ENG) and TLE structured fields in the following form:

BDT (Begin Document)

BNG (Begin Named Page Group)

TLE (group - required)

BPG (Begin Page)

TLE (page - optional)

EPG (End Page)

ENG (End Named Page Group)

EDT (End Document)

When an input file contains TLE structured fields, ACIF does not index the file. If you specify indexing parameters (such as TRIGGER, INDEX, or FIELD), for a file that contains TLE structured fields, ACIF will fail with error msg 462 - A trigger parameter was specified, but the input file is already indexed.

ACIF processes a file containing TLE structured fields in the following way:

1. For every BNG in the input, ACIF create a group IEL structured field in the index file.
2. ACIF makes a copy of the TLE structured fields from the input and places them into the index file. The original TLE structured fields remain in the input file.

Therefore, the result of ACIF processing under these circumstances is the creation of an index file. ACIF can complete normally but the load process into OnDemand may still fail if the format of the input file is incorrect:

- If the input file contains BNG - ENG pairs with no group level TLE Structured Fields between them, then the load process will fail with the message : 0 fields submitted, n expected, where n is the number of fields defined to OnDemand.

- If the input file does not contain any BNG - ENG pairs, then the load process may run out of memory looking for the start and end of the groups.

Running ACIF under these circumstances, you may find that the output file is larger than the input. Why, you may wonder, since they are both AFP files? The answer is because ACIF changes the AFP, "improves it", and usually ends up increasing the file size. Some of the changes to the AFP may include the following: creating or adding comments to the BDT Structured Field, creating or adding group names to the BNG - ENG Structured Fields, adding TLE Structured Fields (if indexing), and changing obsolete Structured Fields to current ones (e.g. MCF1 to MCF2, PTD1 to PTD2).

### **Solution:**

Make sure that **every** BNG - ENG pair has the correct number of TLEs between the BNG and the BPG.

## **7. Be Careful Using GROUPMAXPAGES**

### **Problem:**

Hits in the hit list reference a group which does not contain the index field

### **Diagnosis:**

The ACIF parameter GROUPMAXPAGES=*n* will force a new group every *n* pages. It is commonly used when there is no "natural" break in the data, i.e. when the value of the fields one is using to index never change. When ACIF forces a new group, it copies the TLEs from the previous group to the new group. ACIF is designed to do this, because, since *n* is an arbitrary number, there is no guarantee that the new group will contain any index data - and every group must have indexes. This copying of the TLEs can produce strange and unwanted results: there may be hits in the hit list that reference a group which does not contain the index field.

### **Solution:**

We recommend, instead of using GROUPMAXPAGES, use OnDemand Large Object support instead. Remember, in order to use Large Object Support, one must specify INDEXOBJ=ALL in the ACIF parameters.

## **8. More Carriage Control Problems**

### **Problem:**

Message from ARSLOAD

Output/Indexer file not created

## Diagnosis:

The data does not contain standard ANSI carriage controls, but rather "PC style" carriage returns and form feeds: X'0D0A' and X'0D0C'. Because ACIF does not recognize X'0D' and X'0C' as carriage controls, use the OnDemand provided user exits ASCIINP and ASCIINPE to remove the '0C'x and '0D'x and insert ANSI carriage controls in their place. Because ASCIINP and ASCIINPE insert carriage controls, they change the position of triggers and fields. When you are using these exits, add 1 to the column offsets for the Triggers and Fields. If you would like to see how these exits are going to change your data, here is a trick: set up ACIF to run with the exit enabled, no indexing parameters, and the ACIF parameter CONVERT=NO. Run ARSLOAD so that the files are not loaded, so that you can look at the .out file. The .out file is now in the format that the ACIF indexer will receive. By examining the .out file, you will see how the exit has changed your data, and you can make sure that you have the correct column offsets for your TRIGGER and FIELD parameters. (This is also a very useful technique for developing and debugging your own Input Exit).

ASCIPE and ASCIINP make the following conversions:

First, it inserts a '31'X (new page) as the first byte  
Changes '0C'X (form feed) ==> '31'X (new page)  
Changes '0D'X (carriage return) ==> '20'X (new line)  
Leaves '0A'X in file

Here is an example of what can happen during this process:

Hex string '0A 0D 0C' becomes '0A 20 0A 31'. Notice that an extra line has been added. Depending on how you set up your indexing, this line may have to be taken into account when calculating offsets for fields from the triggers.

Remember - if you are using asciinpe, the TRIGGERS must be in EBCDIC, and set CPGID=500, because the exit converts the data to EBCDIC before it goes to the part of ACIF that does the indexing. If you forget and put the TRIGGERS in ASCII, you may get this message from the loading process: "Output/Indexer file not created", because ACIF could not find the TRIGGERS.

A file loaded into OnDemand with CC=NO and CONVERT=NO will appear as one page when viewed, unless one uses the LINECNT parameter. Therefore, LINECNT is required under these circumstances.

## Solution:

Use the OnDemand provided user exits ASCIINP or ASCIINPE for this type of data. Here are some sample parameters for using asciinpe. Set CCTYPE=Z because the exit does not convert the carriage control.

```
CC=YES
CCTYPE=Z
CPGID=500
MCF2REF=CPCS
CONVERT=YES
TRC=NO
FILEFORMAT=STREAM, (NEWLINE=X'0A')
TRIGGER1=*...
TRIGGER2=...
```



FIELD1=  
FIELD2=  
INDEX1=  
INDEX2=  
DCFPAGENAMES=NO  
UNIQUEBNGS=YES  
IMAGEOUT=ASIS  
INDEXOBJ=GROUP  
INPEXIT=<path>asciinpe

## ***Notices***

Advanced Function Presentation, AFP, and IBM are trademarks of International Business Machines Corporation in the United States, other countries or both.