Purpose

The purpose of this document is to provides details of the enhancements provided by APAR PI56881. These enhancements affect two areas.

- One provides the ability for the report specification archive definition exit (arsuupdt) to provide indexer parameters that will be appended to the end of the indexer parameters being passed to the indexer.

- The other enhancement is a means for the 390 indexer exits to indicate that an error has occurred, and that arsload should fail the load of the report.

Report Specification Archive Definition

The high-level ArsCSXitUpdtExit structure in the arscsxit header file has had a field, AdditionalIndexerParms, added. The corresponding COBOL name in the arscsxic copybook is ArsCSXitUpdtExit-AddIndexer.

C definition:

```
typedef struct _ArsCSXitUpdtExit
{
   char              *pFileName;
   ArcI32            Function;
   char              ApplGrpName[ARCCSXIT_MAX_NAME_SIZE + 1];
   char              ApplName[ARCCSXIT_MAX_NAME_SIZE + 1];
   char              ObjServer[ARCCSXIT_MAX_SERVER_SIZE + 1];
   char              StorageNode[ARCCSXIT_MAX_NAME_SIZE + 1];
   void              *pJES;
   char              IndexerParms[ARCCSXIT_MAX_INDEXER_SIZE + 1];
   ArcCSXitCarCtl    CCType;
   ArcI32            LRECL;
   ArcCSXitDocFormat RECFM;
   ArcI32            UpdateAppl;
   char              Delim[ARCCSXIT_MAX_DELIMITER_SIZE + 1];
   char              *instance;
   char              DbFieldName[ARCCSXIT_MAX_DBCOL_NAME_SIZE + 1];
   char              DbFieldDateFormat[ARCCSXIT_MAX_DATEFMT_SIZE + 1];
   char              AdditionalIndexerParms[ARCCSXIT_MAX_INDEXER_SIZE + 1];
} ArsCSXitUpdtExit;
```

COBOL definition:

```
       01  ArsCSXitUpdtExit.
           03 ArsCSXitUpdtExit-pFileName      pointer.
           03 ArsCSXitUpdtExit-Function       pic s9(9) comp.
              88 arccsxit-process-names       value 1.
              88 arccsxit-process-parms       value 2.
           03 ArsCSXitUpdtExit-ApplGrpName    pic x(61).
           03 ArsCSXitUpdtExit-ApplName       pic x(61).
           03 ArsCSXitUpdtExit-ObjServer      pic x(255).
           03 ArsCSXitUpdtExit-Node           pic x(61).
```

```
        03 filler                          pic x(2).
        03 ArsCSXitUpdtExit-pJES           pointer.
        03 ArsCSXitUpdtExit-Indexer        pic x(32701).
        03 ArcCSXitUpdtExit-cc-type        pic x(1).
            88 ArcCSXitUpdtExit-cc-ansi        value 'A'.
            88 ArcCSXitUpdtExit-cc-machine     value 'M'.
            88 ArcCSXitUpdtExit-cc-none        value 'N'.
        03 filler                          pic x(2).
        03 ArsCSXitUpdtExit-lrecl          pic s9(9) comp.
        03 ArsCSXitUpdtExit-recfm          pic x(1).
            88 ArcCSXitUpdtExit-fixed          value x'00'.
            88 ArcCSXitUpdtExit-variable       value x'01'.
            88 ArcCSXitUpdtExit-stream         value x'02'.
        03 filler                          pic x(3).
        03 ArsCSXitUpdtExit-update-appl    pic s9(9) comp.
        03 ArcCSXitUpdtExit-delim          pic x(17).
        03 filler                          pic x(3).
        03 ArsCSXitUpdtExit-pInstance      pointer.
        03 ArcCSXitUpdtExit-DbFieldName    pic x(19).
        03 ArcCSXitUpdtExit-DbFieldDatFmt  pic x(61).
        03 ArsCSXitUpdtExit-AddIndexer     pic x(32701).
```

When the arsuupdt exit is invoked, this new field is set to an empty string. On
return from the exit, this field should contain a null-terminated string to be
appended to the end of the indexer parameters before they are passed to the
indexer. If multiple parameter lines need to be provided, the lines should be
separated by the record delimiter. On z/OS, this is a X'15'.

The supplied sample ARSUUPDT and ARSUUPDC source have been updated
to support the additional indexer parameters. If you have customized either
ARSUUPDT or ARSUUPDC samples, those customizations will need to be
refitted to the samples in order to take advantage of the new
AdditionalIndexerParms field.

If using the assembler language interfaces, additional fields
ARSUUPDA_AIXPRMP, ARSUUPDA_AIXPRML, and
ARSUUPDA_AIXPRMMXL are provided that contain a pointer to the area to
store the additional indexer parameters, the length of the data pointed to by
ARSUUPDA_AIXPRMP, and the maximum size of the area pointed to by
ARSUUPDA_AIXPRMP, respectively. The assembler routines can set the area
pointed to by ARSUUPDA_AIXPRMP to the desired data, and set
ARSUUPDA_AIXPRML to the length of the data.


```
ARSUUPDA_AIXPRMP  DC    A(0)              UI: Pointer to a null
*                                         terminated additional Indexer
*                                         Parameters string.
*
*                                         Note:  Parameters within
*                                         this string are delimited
*                                         by Newline (x'15') chars.
*
```

```
ARSUUPDA_AIXPRML  DC    F'0'            U:  The true length
*                                           (exclusive of the
*                                           terminating null) of the
*                                           Additional Indexer Parameters
*                                           Parameters string.
*
ARSUUPDA_AIXPRMMXL DC   F'0'            I:  The maximum length of
*                                           the string (exclusive of
*                                           the terminating null)
*                                           which can be placed in the
*                                           area pointed to by the
*                                           AIXPRMP field.
```

The ARSUUPDA (Assembler), ARSUUPDH (C) and ARSUUPDB (COBOL) SARSINST members have been updated for these new fields.

If you are using the sample ARSUUPDX or ARSUUPDZ programs, the samples must be re-assembled and linked in order to take advantage of the new ARSUUPDA_AIXPRMP, ARSUUPDA_AIXPRML, and ARSUUPDA_AIXPRMMXL fields.

390 Indexer exits

Prior to PI56881, only the 390 Indexer ANYEXIT could return a status to indicate an error had occurred. The INPEXIT and INDXEXIT could not, and thus had no legitimate way to indicate that the load of the report should fail. As a consequence, sample exits shipped called abort(), which terminates the invoking program.

With PI56881, all three exits can return a status indicating that an error has occurred.

For the INPEXIT, the supplied ARSINPBK has been updated to provide a REC-ERROR to the existing RECORD-STATUS to indicate an error:

```
01  RECORD-STATUS       PIC X(3).
    88 END-OF-FILE       VALUE 'EOF'.
    88 REC-ERROR         VALUE 'ERR'.
```

For the INPEXIT, the supplied ARSIN2BK has been updated to provide a ARSE2INP-ERR to the existing RECORD STATUS to indicate an error:

```
01  RECORD-STATUS                 PIC X(3).
    88 END-OF-FILE                 VALUE 'EOF'.
    88 ARSE2INP-ERR                VALUE 'ERR'.
```

For the INDXEXIT, the supplied ARSINDBK has been updated to provide a WS-

IND-STATUS that contains WS-IND-ERR to indicate an error.

```
01  WS-IND-STATUS              PIC XXX.
    88  WS-IND-OK                 VALUE '   '.
    88  WS-IND-ERR                VALUE 'ERR'.
```

When the INDXEXIT is called, the additional WS-IND-STATUS parameter is passed that can be used to indicate an error:

```
PROCEDURE DIVISION USING WS-HEADER-RECORD
                         WS-INDEX-NAME-COUNT-ORIGINAL
                         WS-INDEX-NAMES
                         WS-OS-LEVEL-FLAG
                         WS-EBCDIC-ASCII-SWITCH
                         WS-INDEXES-33
                         WS-IND-STATUS.
```

Regardless of which 390 indexer exit is called, if the exit sets the 'ERR' in the appropriate status , the load of the report will fail.

In addition, the sample 390 indexer exits have been updated to remove calls to abort(), and to instead set the 'ERR' status appropriately.