UniVerse 11.1
Indexed Subroutine Enhancement

Rocket U2™

Powering Business Solutions

Your primary source for enterprise software

Rocket®

Indexed Subroutine Enhancement

Agenda

- Overview
- Indexed Subroutine behavior
- New @variable introduced at 11.1
- Impact on existing applications
- Example
- Performance comparison

Indexed Subroutine Enhancement

Overview

- A new @-variable (@IDX.IOTYPE) has been added at UniVerse 11.1 which allows an index based on a BASIC subroutine to perform some functions similar to SQL based file triggers.
- @IDX.IOTYPE can be accessed within an indexed subroutine to determine the type of I/O operation being performed.
- An indexed subroutine will typically have less overhead than a SQL based trigger and may be more performant for certain operations.

Indexed Subroutine Enhancement

UniVerse Indexed Subroutine Behavior

- **What happens when writing/deleting a record in a file which has an Index based on a BASIC subroutine**
  - **The subroutine is called once when adding a new record to the file or when deleting an existing record.**
  - **On an update to an existing record, the subroutine is called twice.**
    - ***First to evaluate the current value of the index.***
    - ***Second to evaluate the new value of the index.***
    - ***This is done to determine if an index update is needed.***

Indexed Subroutine Enhancement

UniVerse Indexed Subroutine Behavior

- **Prior to 11.1, there was no way to determine what type of I/O operation was being performed (i.e. insert, update, or delete) while in the subroutine.**
- **At UniVerse release 11.1, the @variable @IDX.IOTYPE has been introduced.**
- **While the indexed subroutine is executing, the @IDX.IOTYPE variable contains a numeric value corresponding to the type of I/O operation being performed.**

Indexed Subroutine Enhancement

Possible Values for @IDX.IOTYPE

- **0 is returned when checked outside an indexed subroutine**
- **1 is returned for an INSERT (i.e. when a new record is being added to the file)**
- **2 is returned for a DELETE (i.e. when an existing record is being deleted from the file)**

Indexed Subroutine Enhancement

Possible values for @IDX.IOTYPE

- **3 is returned for an UPDATE when the subroutine is called to evaluate the original index value of an existing record (@RECORD contains original record contents)**

- **4 is returned for an UPDATE when the subroutine is called to evaluate the new index value of an existing record (@RECORD contains new record contents)**

Indexed Subroutine Enhancement

No Impact on Existing Applications

- **This change does not alter how UniVerse indices functioned prior to 11.1.**
- **The only change done at 11.1 related to this enhancement is that @IDX.IOTYPE is now available for use within an indexed BASIC subroutine.**
- **Indexes based on BASIC subroutines which do not use @IDX.IOTYPE will not be impacted.**

Rocket U2™ *Powering Business Solutions*

Indexed Subroutine Enhancement

Indexed Subroutine Example

```
SUBROUTINE INDEX.SUB(RTNVAL)
COMMON /INDEX.SUB/ OPENFLAG,F.AUDIT,OLDRECORD
RTNVAL = "" ;* Set index value to "" for NO.NULLS index
OPERATIONS = "INSERT":@FM:"DELETE":@FM:"UPDATE":@FM:"UPDATE"
IF NOT(OPENFLAG) THEN
   OPEN "AUDIT.FILE" TO F.AUDIT ELSE STOP "CANNOT OPEN AUDIT.FILE"
   OPENFLAG = 1
END
*
```

(continued on next page)

Indexed Subroutine Enhancement

```
* The following case statement can be used to execute any specific
* operations related to the type of operation being performed.
*
    AUDIT.REC = ''
    BEGIN CASE
      CASE @IDX.IOTYPE = 1        ; * INSERT
      CASE @IDX.IOTYPE = 2        ; * DELETE
      CASE @IDX.IOTYPE = 3        ; * UPDATE BEFORE
        OLDRECORD = LOWER(@RECORD)
      CASE @IDX.IOTYPE = 4        ; * UPDATE AFTER
        AUDIT.REC<2> = OLDRECORD
      CASE 1
        RETURN
    END CASE
    IF @IDX.IOTYPE # 3 THEN
      RECID = @DATE:"*":SYSTEM(12):"*":@ID
      AUDIT.REC<1> = OPERATIONS<@IDX.IOTYPE>
      WRITE AUDIT.REC ON F.AUDIT,RECID
    END
    RETURN
  END
```

Indexed Subroutine Enhancement

Creating an Index Subroutine

- BASIC BP INDEX.SUB
- CATALOG BP INDEX.SUB
- CT DICT TEST.IDX INDEX.ITYPE
  - 0001: I
  - 0002: SUBR(INDEX.SUB)
  - .
- CREATE.INDEX TEST.IDX INDEX.ITYPE NO.NULLS
- BUILD.INDEX TEST.IDX INDEX.ITYPE

Indexed Subroutine Enhancement

Performance Test Example

- Comparison testing done at 11.1.0 on AIX, HP, and 11.1.1 on Windows platforms.
- Testing was done using an indexed or trigger subroutine which simply returned after being called.
- Test program wrote 2 million records into both empty and full files using either index or trigger.
- Elapsed time to perform test was consistently 2 to 3 times longer for trigger than indexed subroutine.
- Your mileage may vary.

Rocket U2™
Powering Business Solutions