R$B2WR – Direct File Access Rewrite from Saved File Address

The R$B2WR routine is one of a family of sub-routines that are reserved for use by the G-3000 Middleware DLMs. R$B2WR(ITE) rewrites a record at the last saved file address.

1. Invocation

To rewrite a record in the open BDAM file code:

```
CALL R$B2WR USING record rec_len [file_addr]
```

where *record* is the PIC X(n) record area; *rec_len* is a PIC 9(4) COMP variable, or literal, containing the length of the record to write and *file_addr* is an optional PIC 9(9) COMP variable (**not a literal**) which contains the file address of the record to be rewritten.

2. STOP Codes and Exception Conditions

The following STOP codes are generated by R$B2WR:

STOP code	**Description**
43 (*sic*)	The BDAM file was not open; or no File Address has been saved (i.e. R$B-SA has not been called before the R$B2WR call).

No EXIT codes are returned by R$B2WR.

3. Programming Notes

R$B2WR(ITE) has been derived from a G-3000 Middleware DLM. Consequently, some of the functionality (e.g. the STOP codes) may be non-standard.

The BO$G3M DLM, rather than the calling program, contains the File Definition (FD) used for the various Direct Access File operations. If the *file_addr* parameter is not supplied the file address of the record is restored from a local field within the DATA DIVISION of the BO$G3M DLM.

The R$B-SA routine **must** be called before R$B2WR to save the address of the (then) current record. If the saved record number is valid R$B2WR uses the B$D-2W routine to rewrite the record. Note that the possible exception code returned by B$D-2W is **not** trapped by R$B2WR.

The current file address (at the time of the R$B2WR call) is saved in a temporary work field before the file address at the time of the R$B-SA call is used for the B$D-2W write function. After the B$D-2W function has completed the current file address (at the time of the R$B2WR call) is restored.

The R$B-SA routine either saves the current file address either in a field supplied by the calling program (if the optional single parameter is supplied) or in a local field within the DATA DIVISION of the BO$G3M DLM (if R$B-SA is called with no parameters). The saved file address used by R$B2WR is either restored from a field supplied by the calling program (if the optional third parameter is supplied) or in a local field within the DATA DIVISION of the BO$G3M DLM (if R$B2WR is called with just two parameters). No attempt is made to check the calls of R$B-SA and R$B2WR are balanced. For example:

```
    CALL R$B-SA USING Z-99C          * SAVE FILE ADDRESS IN CALLING PROGRAM
    …
```

```
        CALL R$B2WR USING REC LENG          * ATTEMPT TO RESTORE FILE
ADDRESS
                                            * FROM FIELD IN BO$G3M. THIS WILL
                                            * NORMALLY RESULT IN A STOP 43
```

Note also that the internal "saved file address" is not invalidated after it has been used by R$B2WR to call B$D-2W. Perhaps it should be to detect this potential coding error:

```
        CALL R$B-SA                         * SAVE FILE ADDRESS
        …
        CALL R$B2WR USING REC LENG          * REWRITE FROM SAVED FILE
ADDRESS
        ….
        CALL R$B2WR USING REC LENG          * 2ND REWRITE INTENTIONAL?
```

If the internal "saved file address" was invalidated after the first R$B2WR call then the subsequent R$B2WR call, without a previous R$B-SA, would be detected (by a STOP WITH 43).

4. Examples

[EXAMPLE REQUIRED]

5. Copy-Books

None.

6. See Also

B$D-2R	Direct File Access Simple Read Next Routine (B$D-2RNXT)
B$D-2W	Direct File Access Simple Write Next Routine (B$D-2WRITE)
B$D-CH	Direct File Access Check and Close Routine (B$D-CHK)
B$D-CL	Direct File Access Close Routine (B$D-CLOSE)
B$D-DE	Direct File Access Delete Routine (B$D-DELETE)
B$D-IN	Direct File Access Return File Information Routine (B$D-INFO)
B$D-OL	Direct File Access Open Old Routine (B$D-OLD)
B$D-OP	Direct File Access Open New Routine (B$D-OPEN)
B$D-PO	Direct File Access Set Current File Position Routine (B$D-POSITION)
B$D-UN	Direct File Access Set Unit Routine (B$D-UNIT)
B$D-WR	Direct File Access Write Speedbase Channel (B$D-WRITE)
B$E-CL	Direct File Access Extra Close Routine (B$E-CLOSE)
B$E-GE	Direct File Access Get Record Length Routine (B$E-GET-RLEN)
B$E-RN	Direct File Access Simple Read Block Routine (B$E-RNXT)
B$E-SE	Direct File Access Set Record Length Routine (B$E-SET-RLEN)
B$E-ZE	Direct File Access Set File Address to Zero Routine (B$E-ZERO)
R$B-SA	Direct File Access Save Address of Current Record (R$B-SAVE)
R$B-WR	Direct File Access Rewrite from Saved File Address (R$B-WRITE)
R$PFSC	Direct File Access Check For Free Space (R$PFSCHK)