

MVBFS Functions

OpenInsight Record Locking for Multivalued Developers

Unlike most multivalued databases, OpenInsight basic+ has explicitly separated the locking and reading/writing of records. In OpenInsight, programmers call the [LOCK](#) statement before reading a record; they then modify the record (which does `_not_` change the state of the lock), and then use the [UNLOCK](#) statement when it is appropriate to release the record lock.

However, when accessing a multivalued database through the MVBFS interface, it is important to use the remote database's locking to maintain database integrity. While this can be accomplished using the OpenInsight [LOCK](#) and [UNLOCK](#) statements, as a convenience the functions [RTI_READU](#), [RTI_WRITEU](#), and [RTI_WRITERELEASE](#) have been implemented.

[RTI_READU](#) mimics the [READU](#) functionality in other multivalued systems; it uses the host [READU](#) command (or comparable functionality) to lock and read the record, returning a status that indicates if the record was already locked, does not exist, or has been successfully read.

[RTI_WRITEU](#) mimics the [WRITEU](#) functionality in other multivalued systems; it writes the record and leaves the status of any locks unchanged. Note that this is comparable to normal OpenInsight [WRITE](#) functionality, and thus [WRITE](#) can be used instead of [RTI_WRITEU](#) if desired.

[RTI_WRITERELEASE](#) mimics the [WRITE](#) functionality in other multivalued systems; it writes, and then releases any locks on, the record.

If a lock on a record has been acquired with the [RTI_READU](#) function, it should normally be released by using the [RTI_WRITERELEASE](#) function. However, if the lock needs to be released without writing the record, you can use the [UNLOCK](#) statement and the lock will be released (either locally, if on an OpenInsight table, or on the host if on an MVBFS file).

[MVBFS_NATIVE_TABLE](#) is a function added in OpenInsight 9.4 to allow developers to read or write records to server-side tables even if they are not mapped. The primary purpose is to let a developer read or write values to a non-standard dictionary row, for instance a sequence counter.

See Also

[RTI_READU_EQUATES](#)