BR_EntityLink – Entity Link Services

Robin Lamacraft 2017-03-24

SCOPE

This module must be used to allocate, disable and retrieve the persistent (non-reusable) unique HRE_ID values that control the type and inter-relationship of all records in all HRE databases.
NOTE: All records in all HRE tables have an HRE-ID identity, whether they represent an Entity or a Link between 2 Entities.

This module also controls the access to a database record (or its equivalent memory-based object). It compares the rights of the current user with the rights associated with that data base record to confirm or deny an action on that record.

This module also assigns the ownership and rights for each new data base record created.

ACTIONS

The fundamental operations are:

1. Create unique persistent HRE-ID. This construction is dependent on the HRE data type that is being requested.

   PROPOSAL: An HRE_ID is a 64bit integer value that has 4 parts with following suggested partitions:
   - The HRE-Entity Group (maximum 254 groups = 8 bits as 127 for user data and 127 for application data)
   - The HRE-Entity Pattern Index within a Group (maximum 65,536 = 16 bits)
   - The persistent index of this instance within that HRE- Entity Group Pattern Type (maximum 4,294,967,295 = 32 bits). This becomes the maximum number of records that have been created in a table (including the disabled and purged records over time)
   - Whether this is the most current record or not – a negative HRE-ID is a disabled version of the positive value. There can be only one positive but there may be several negative values existing the database at any one time.

   The encoding and the decoding of the HRE_ID must be extremely simple. The proposed encoding is very quick to decode for various purposes. When a record is superseded by a new update, reverse the sign on the disabled copy. Using the absolute value function ABS() in SQL will enable recovery of all transactions on that record that are still remaining in that table. Note that database transactions can simultaneously update several tables. The database would become corrupted if not all changes were recognized for any sequence of undo actions.
   a. Each HRE database record must have its own unique HRE-ID. Each record must have 3 Transaction ID fields - This Transaction, Previous Transaction and Next Transaction
   b. At the creation of a record, This Transaction ID is set, both Previous Transaction ID and Next Transaction ID are zero
   c. On subsequent update, the replacement record is created with a copy of the old record's HRE_ID, the original This Transaction ID is copied to the new record's Previous Transaction ID field
   d. On the disabled record, rewrite it with the new record's Transaction ID copied into the Next Transaction ID field and rewrite the HRE-ID as its negative value.

The above implies that this module must maintain a database of HRE table of highest used HRE-ID indexes for each HRE database table type in order that it can allocate unique values to a request to create a new table record.

2. Allocate new HRE-ID
3. Decode an HRE ID
4. Retrieve the identity, ownership , access rights, timestamp and transaction ID of a database record
5. Perform searches for a change log of transactions
6. Perform database change undo
7. Perform purge of transaction records before a particular date/time
8. For a given HRE-ID, provide a list of the non-identity fields in all HRE tables that could hold a reference to that HRE-ID
9. Needs other details.

USED BY:

Any data management process.

DATA UNDER CONTROL OF THIS MODULE

1. The database Field Definition table
2. HRE_ID last used ID database
3. HRE-ID where referenced database.

REQUIRED EXTERNAL DATA UNDER CONTROL OF ANOTHER MODULE

1. Transaction Data Base.

REQUIRED SERVICES

1. Needs details

APPLICATION PROGRAMMING INTERFACE (API)

1. Needs details

EVENT ACTIONS

1. Need details of event.

WARNING CONDITIONS

1. Need details of the condition that raised the warning, example message and possible next steps.

ERROR CONDITIONS

1. Need to record the condition that raised the error, example message and possible next steps.