

1. Obtain an entity/attribute list object

Responsible Team:

Priority:

Characteristic Information: Calling Application can parse a metadata document for entity and attribute information. The returned object contains a list of the entity and attribute information.

Goal in Context:	Obtain an entity/attribute list object.
Scope:	
Level:	
Pre-Condition:	Metadata package exists with a unique ID. Appropriate parsing routine exists for the specific metadata package schema.
Success End Condition:	A correct entity/attribute list object is returned to the calling application.
Failed End Condition:	A metadata package is not located. An entity/attribute list object is not returned or is returned with incorrect information.
Primary Actor:	Calling Application (CA)
Trigger Event:	Calling Application requests an entity/attribute list object from the Data Manager controller.

Main Success Scenario:

Step	Actor	Action Description
1.		
2.		
3.		
4.		
5.		

Scenario Extensions: None

Step	Condition	Action Description

Scenario Variations

Step	Actor	Action Description

2. Download data

Responsible Team:

Priority:

Characteristic Information: Calling Application can download data from the source as defined in the entity/attribute list object and have it stored in the Calling Application's data store.

Goal in Context:	Download data from source.
Scope:	
Level:	
Pre-Condition:	Use Case #1. Data access information is available. A data store is defined.
Success End Condition:	Data is downloaded into the Calling Application's data store.
Failed End Condition:	Data is not downloaded into the Calling Application's data store. Incorrect data is downloaded into the Calling Application's data store.
Primary Actor:	Calling Application (CA)
Trigger Event:	Calling Application requests that data is download from the source by the Data Manager controller.

Main Success Scenario:

Step	Actor	Action Description
1.		
2.		
3.		
4.		
5.		

Scenario Extensions: None

Step	Condition	Action Description

Scenario Variations

Step	Actor	Action Description

3. Load data into a relational database table

Responsible Team:

Priority:

Characteristic Information: Calling Application requires data to be loaded into a database table using a RDBMS of choice.

Goal in Context:	Load data into a relational database table.
Scope:	
Level:	
Pre-Condition:	Use Case #1 and #2.
Success End Condition:	A relational database table is created according to the DDL generated by the entity/attribute list object and associated data is loaded into the table.
Failed End Condition:	A relational database table is not created. The wrong data is loaded into the relational database table.
Primary Actor:	Calling Application
Trigger Event:	Calling Application requests that data be loaded into a relational database table by the Data Manager controller.

Main Success Scenario:

Step	Actor	Action Description
1.		
2.		
3.		
4.		
5.		

Scenario Extensions: None

Step	Condition	Action Description

Scenario Variations

Step	Actor	Action Description

4. Select data from a database table

Responsible Team:

Priority:

Characteristic Information: Calling Application requests data from a relational database table based on an ANSI SQL select statement.

Goal in Context:	Return data to the Calling Application.
Scope:	
Level:	
Pre-Condition:	Use Case #1, #2, and #3.
Success End Condition:	Correct data is returned to the Calling Application (including a NULL return, which would indicate that the query requirements for data were not met).
Failed End Condition:	Incorrect data is returned to the Calling Application.
Primary Actor:	Calling Application
Trigger Event:	Calling Application requests data from the Data Manager controller.

Main Success Scenario:

Step	Actor	Action Description
1.		
2.		
3.		
4.		
5.		

Scenario Extensions: None

Step	Condition	Action Description

Scenario Variations

Step	Actor	Action Description

5. Set upper-limit database size

Responsible Team:

Priority:

Characteristic Information: Calling Application can set an upper-limit on the database size (to prevent over loading the RDBMS).

Goal in Context:	Set an upper-limit for the size of the database.
Scope:	
Level:	
Pre-Condition:	Database table management must abide by the upper-limit size constraint. Table management must include routines to drop tables when size constraints are met.
Success End Condition:	An upper-limit database size is set.
Failed End Condition:	An upper-limit database size is not set. An upper-limit database size is set incorrectly.
Primary Actor:	Calling Application
Trigger Event:	Calling Application requests that an upper-limit database size be set by the Data Manager controller.

Main Success Scenario:

Step	Actor	Action Description
1.		
2.		
3.		
4.		
5.		

Scenario Extensions: None

Step	Condition	Action Description

Scenario Variations

Step	Actor	Action Description

6. Set database table life-span priority

Responsible Team:

Priority:

Characteristic Information: Calling Application can set a table life-span priority on specific database tables for use during “least used” removal algorithm during database table memory garbage collection.

Goal in Context:	Set a table life-span priority for a specific table.
Scope:	
Level:	
Pre-Condition:	A data table ID exists in the Table Manager table.
Success End Condition:	A table life-span priority is set in the Table Manager table.
Failed End Condition:	A table life-span priority is not set in the Table Manager table. An incorrect table life-span priority is set in the Table Manager table.
Primary Actor:	Calling Application
Trigger Event:	Calling Application requests that a table life-span priority is set by the Data Manager controller.

Main Success Scenario:

Step	Actor	Action Description
1.		
2.		
3.		
4.		
5.		

Scenario Extensions: None

Step	Condition	Action Description

Scenario Variations

Step	Actor	Action Description