Macintosh HD:Users:jon:Documents:DLM Forum:DLM Foundation:Branding:Logos:MoReq logos:MoReq2010:MoReq2010_Reg:MoReq2010_Reg_pms.eps

MoReq2010®

Modular Requirements for Records Systems

Test Framework

Test Module 3: Records

Version 0.1 BETA

Copyright © 2011 & 2012 DLM Forum Foundation, all rights reserved

# 3.0 Scope

This test module tests requirements from the following sections of the MoReq2010® specification:

* **2. System Services**
* **5. Classification Service**
* **6. Record Service**
* **10. Searching and Reporting Service**
* **11. Export Service**
* **101. Graphical User Interface**
* **102. Application Programming Interface**

Of these the major emphasis of the test module is placed on **6. Record Service**.

This test module should be used whenever the core services of MoReq2010® are being tested and must be tested after **Test Module 0: Familiarisation and training**.

Additional notes are provided to test centres for clarity throughout. From time-to-time the test scripts will be updated; be sure to check that you are using the most up-to-date version of the testing materials.

Macintosh HD:Users:jon:Documents:DLM Forum:DLM Foundation:Branding:Logos:MoReq logos:MoReq2010:MoReq2010_Reg:MoReq2010_Reg_pms.eps

# TEST MODULE 3: RECORDS

(To be completed by the test centre)

# 3.1 Observation record

Supplier name

Product name and version

Test centre name

Date of test

Test location

Supplier representative(s) present during testing

Test centre representative(s) present during testing

DLM Forum observer(s) present during testing

Test system technical architecture, hardware, software and operating system(s)

Test outcomes

Fill in with 🗹 or 🗷 to ensure all test outcomes are met

🞏 The test system was correctly configured prior to testing

🞏 The test system was not reconfigured during testing

🞏 The test module was completed in a single session

🞏 All test cases in the test module were completed, in order

🞏 All test cases in the test module were completed successfully

Refer to the individual test results for all test cases in this test module

Additional remarks or observations

# 3.2 Test script – Records

All test cases in the test script for this test module must be completed in order before moving onto the next test case.

## T3.2.1 – Test case

### 3.2.1.1 Test case description

**Initial error check**

### 3.2.1.2 Test case preconditions

* Test system is operational
* Test data (see appendix **3.A Test data**) are loaded onto test system
* Where the test system uses an API then a suitable test harness has been provided (see the **Test Framework: Overview and Instructions, 4.1 Providing a test harness for API interfaces**)
* The test system has been configured, under **R2.4.13** such that performing any function system wide will generate an event under **R2.4.15** and **R2.4.16** – in other words, all function definitions must have their Generate Event Flag (**M14.4.34**) set
* The test system has been configured, under **R2.4.20** such that the events generated by all functions system wide will be retained by residual entities – in other words, all function definitions must have their Retain On Destruction Flag (**M14.4.88**) set
* If the model role service is implemented then the test system has been configured, under **R4.5.8** such that every access control list is set to inherit all roles, not just administrative roles – in other words, the Include Inherited Roles Flag (**M14.4.43**) for all access control lists for all entities is set by default
* If a native role service is implemented then the test system must deliver an equivalent level of functionality to that provided by the roles listed in appendix **3.A.1.3 Role entities**

### 3.2.1.3 Test case steps (instructions)

* Before logging into the test system take a copy of the external error log, under **R2.4.7**

### 3.2.1.4 Test case post-conditions (expected result)

* Errors in the external error log must contain at least the information listed under **R2.4.7**

### 3.2.1.5 Notes to test centre

* Check whether there are any errors generated by functions failing to perform under **R2.4.8**
* Note that **R2.4.8** is not tested directly unless an error occurs during testing, instead it is tested indirectly through this test case

### 3.2.1.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.2 – Test case

### 3.2.2.1 Test case description

**Initial compliance check**

### 3.2.2.2 Test case preconditions

* **T3.2.1** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.1** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.2.3 Test case steps (instructions)

* Log into the test system as user **Usr.3.1** under **R3.4.1**
* Generate the compliance report for the test system under **R2.4.5**

### 3.2.2.4 Test case post-conditions (expected result)

* User **Usr.3.1** logs in successfully
* The compliance report contains all of the information and metadata required by **R2.4.5** and **R2.4.2**
* Specifically, the compliance report must show compliance with, at least:
  + **User and Group Service v1.x** (**cd532472-85b0-4c1c-82b4-5c8370b7d0e6**)
  + **Model Role Service v1.x (2f6d05c6-51e6-4a32-a7fc-c0a6883eb85b)**

**OR Native Role Service v1.x (d945dcd9-dc2d-491d-965a-11ce936d044b)**

* + **Classification Service v1.x** (**10fea10e-9c2f-4760-9095-f4f9295f4b19**)
  + **Record Service v1.x** (**ced3d0df-3f9f-4807-9e96-b5b790adad4a**)
  + **Model Metadata Service v1.x** (**a600f8d0-2d58-418e-bb41-211d1fd42350**)

**OR Native Metadata Service v1.x** (**66bf4419-d92f-4358-8506-7ee9c06abdcd**)

* + **Disposal Scheduling Service (fd05e284-181f-4f5d-bd8c-4bed835c8931)**
  + **Searching and Reporting v1.x** (**f09984a5-dd31-44d8-9607-22521667c78a**)
  + **Export Service v1.x** (**2777ab81-057e-4aa4-9595-69459ec2dc1e**)
  + **Graphical User Interface v1.x (0f9584e5-552a-4a79-a8ea-3c2801765255)**

**AND/OR Application Programming Interface (654633ec-8b17-4a3c-a483-436ee2bd506a)**

* The compliance report must use the UUIDs listed above, under **R2.4.23**

### 3.2.2.5 Notes to test centre

Describe in the remarks how the user performs a compliance check.

### 3.2.2.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.3 – Test case

### 3.2.3.1 Test case description

**Browse available services (or bundles of services) and inspect the metadata of the record service**

### 3.2.3.2 Test case preconditions

* **T3.2.2** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.2** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.3.3 Test case steps (instructions)

* Browse the services (**E14.2.14**) of the test system, under **R2.4.3**
* Inspect the metadata of the record service or the bundle of services that contains the record service, under **R2.4.3**

### 3.2.3.4 Test case post-conditions (expected result)

* The test system must allow browsing of all services listed under **R2.4.1**, either as individual services or within one or more bundles of services
* The test system must allow inspection of the metadata of services or bundles of services where each service or bundle of services has the metadata listed under Inspect the metadata of the record service or the bundle of services that contains the record service, under **R2.4.3**
* The metadata of the record service or bundle of services must match the compliance report generated previously under **T3.2.2**

### 3.2.3.5 Notes to test centre

It is possible that the test system implements all services as part of a single bundle. In this circumstance the metadata will relate to the test system as a whole.

### 3.2.3.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.4 – Test Case

### 3.2.4.1 Test case description

**Modify the metadata of the record service**

### 3.2.4.2 Test case preconditions

* **T3.2.3** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.3** was completed
* The test operator is logged in as user **Usr.3.1**
* The Title (**M14.4.104**), Description (**M14.4.16**) and Owner Information (**M14.4.62**) for the record service or bundle of services containing the record service must NOT have been set to the values below prior to this test case

### 3.2.4.3 Test case steps (instructions)

* Modify the metadata of the record service or bundle of services containing the record service, under **R2.4.4**, as follows:
  + Change Title (**M14.4.104**) to “MoReq2010® Test System”
  + Change Description (**M14.4.16**) to “Test Module 3: Records”
  + Change Owner Information (**M14.4.62**) to the name and address of the test centre
* Add a comment, “Changed during testing” under **R2.4.18**

### 3.2.4.4 Test case post-conditions (expected result)

* The metadata is modified correctly to the input values
* The comment has been accepted as part of the event generated by the changes to the metadata

### 3.2.4.5 Notes to test centre

Ensure that you can include the Unicode “®” character (U+00AE) in accordance with **R2.4.28**

### 3.2.4.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.5 – Test Case

### 3.2.5.1 Test case description

**Create root aggregations**

### 3.2.5.2 Test case preconditions

* **T3.2.4** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.4** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.5.3 Test case steps (instructions)

* Attempt to create an open root aggregation without a class in the record service under **R6.5.1** and **R6.5.2** with the following metadata:
  + Set Title (**M14.4.104**) to “Agg.3.0”
  + Set Description (**M14.4.16**) to “Created as an open aggregation”
* Attempt to create an open root aggregation without an active class in the record service under **R6.5.1** and **R6.5.2** with the following metadata:
  + Set Class Identifier (**M14.4.4**) to “Cls.3.0”
  + Set Title (**M14.4.104**) to “Agg.3.0”
  + Set Description (**M14.4.16**) to “Created as an open aggregation”
* Create an open root aggregation in the record service under **R6.5.1** and **R6.5.2** with the following metadata:
  + Set Class Identifier (**M14.4.4**) to “Cls.3.1”
  + Set Title (**M14.4.104**) to “Agg.3.1”
  + Set Description (**M14.4.16**) to “Created as an open aggregation”
  + Set Scope Notes (**M14.4.97**) to “Use for records that are to be retained permanently”
  + Set Max Levels Of Aggregation (**M14.4.52**) to “0” under **R6.5.5**
* When creating aggregation **Agg.3.1** include a comment under **R2.4.18**, “First aggregation to be created during testing”
* Create a closed root aggregation in the record service under **R6.5.1** and **R6.5.2** with the following metadata:
  + Set Class Identifier (**M14.4.4**) to “Cls.3.2”
  + Set Title (**M14.4.104**) to “Agg.3.2”
  + Set Description (**M14.4.16**) to “Created as a closed aggregation”
  + Set Scope Notes (**M14.4.97**) to “Use for records that are to be destroyed after 7 years”
  + Do not set Max Levels Of Aggregation (**M14.4.52**)
* Browse the aggregations in the record service under **R6.5.9** and inspect aggregation **Agg.3.1**
* Inspect the event history of aggregation **Agg.3.1** under **R2.4.19**
* Browse the aggregations in the record service under **R6.5.9** and inspect aggregation **Agg.3.2**

### 3.2.5.4 Test case post-conditions (expected result)

* The record service must not allow a root aggregation to be created without any class under **R6.5.2**
* The record service must not allow a root aggregation to be created with residual class **Cls.3.0** under **R6.5.2** (the class must be active)
* No aggregation with the title “Agg.3.0” is created in this test case
* Aggregation **Agg.3.1** is created with the metadata shown
* Aggregation **Agg.3.2** is created with the metadata shown
* The user is able to browse the aggregations in the record service and inspect aggregations under **R6.5.9**
* Upon inspection, aggregation **Agg.3.1** has values all of the following metadata elements, set either automatically by the test system (check that timestamps have appropriate values) or to the values given above:
  + System Identifier (**M14.4.100**)
  + Created Timestamp (**M14.4.9**)
  + Originated Date/Time (**M14.4.61**)
  + Class Identifier (**M14.4.4**)
  + Title (**M14.4.104**)
  + Description (**M14.4.16**)
  + Scope Notes (**M14.4.97**)
  + Max Levels Of Aggregation (**M14.4.52**)
* Upon inspection, aggregation **Agg.3.1** does not have values for any of the following metadata elements:
  + First Used Timestamp (**M14.4.32**)
  + Last Addition Timestamp (**M14.4.48**)
  + Closed Timestamp (**M14.4.5**)
  + Destroyed Timestamp (**M14.4.17**)
  + Parent Aggregation Identifier (**M14.4.63**)
  + Aggregated Timestamp (**M14.4.1**)
* Upon inspection of the event history of aggregation **Agg.3.1** a “Created” event, generated under **R2.4.16** and **R2.4.17** should be present with the following metadata:
  + Event Timestamp (**M14.4.27**) that correlates with the Created Timestamp (**M14.4.9**) of **Agg.3.1**
  + Event Function Identifier (**M14.4.26**) that is set to function Aggregation – Create (**F14.5.5**)
  + Performed By User Identifier (**M14.4.83**) that is set to user **Usr.3.1**
  + Event Comment (**M14.4.25**) created under **R2.4.18** that contains the appropriate comment given above
  + Participating Aggregation Identifier (**M14.4.65**) that is set to the System Identifier (**M14.4.100**) of aggregation **Agg.3.1**
  + Metadata Change Entries (**D14.3.3**) for each of the metadata elements given above
* Upon inspection of the event history of aggregation **Agg.3.1** an “Inspected” event, generated under **R2.4.16** and **R2.4.17** should be present with the following metadata:
  + Event Timestamp (**M14.4.27**) that has occurred since the creation of **Agg.3.1**
  + Event Function Identifier (**M14.4.26**) that is set to function Aggregation – Inspect (**F14.5.12**)
  + Performed By User Identifier (**M14.4.83**) that is set to user **Usr.3.1**
  + Participating Aggregation Identifier (**M14.4.65**) that is set to the System Identifier (**M14.4.100**) of aggregation **Agg.3.1**
  + There should be no Event Comment (**M14.4.25**) and no Metadata Change Entries (**D14.3.3**) in this event
* Upon inspection, aggregation **Agg.3.2** has equivalent values to those given for aggregation **Agg.3.1** for all metadata elements, except:
  + **Agg.3.2** also has Closed Timestamp (**M14.4.5**) – because it was created as a closed aggregation
  + **Agg.3.2** does not have Max Levels Of Aggregation (**M14.4.52**) – because no limit was placed on the number of levels of aggregation upon creation

### 3.2.5.5 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.6 – Test Case

### 3.2.6.1 Test case description

**Create child aggregations**

### 3.2.6.2 Test case preconditions

* **T3.2.5** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.5** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.6.3 Test case steps (instructions)

* Attempt to create an open child aggregation under **R6.5.1** and **R6.5.2** with the following metadata:
  + Do not set Class Identifier (**M14.4.4**) – class is inherited from the parent aggregation
  + Set Title (**M14.4.104**) to “Agg.3.3”
  + Set Description (**M14.4.16**) to “Created as a child aggregation”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.1”
* Attempt to create an open child aggregation under **R6.5.1** and **R6.5.2** with the following metadata:
  + Set Class Identifier (**M14.4.4**) to **Cls.3.3** – overriding class inherited from the parent aggregation
  + Set Title (**M14.4.104**) to “Agg.3.4”
  + Set Description (**M14.4.16**) to “Created as a child aggregation”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.2”
* Find or browse to aggregation **Agg.3.1** and inspect it under **R6.5.9**
* Modify the Max Levels Of Aggregation (**M14.4.52**) for aggregation **Agg.3.1** under **R6.5.5** to “1”
* Find or browse to aggregation **Agg.3.2** and inspect it under **R6.5.9**
* Open aggregation **Agg.3.1** under **R6.5.6**
* Create an open child aggregation under **R6.5.1** and **R6.5.2** with the following metadata:
  + Do not set Class Identifier (**M14.4.4**) – class is inherited from the parent aggregation
  + Set Title (**M14.4.104**) to “Agg.3.3”
  + Set Description (**M14.4.16**) to “Created as a child aggregation”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.1”
* Create an open child aggregation under **R6.5.1** and **R6.5.2** with the following metadata:
  + Set Class Identifier (**M14.4.4**) to **Cls.3.3** – overriding class inherited from the parent aggregation
  + Set Title (**M14.4.104**) to “Agg.3.4”
  + Set Description (**M14.4.16**) to “Created as a child aggregation”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.2”
* Attempt to create an closed child aggregation under **R6.5.1** and **R6.5.2** with the following metadata:
  + Set Class Identifier (**M14.4.4**) to **Cls.3.3** – overriding class inherited from the parent aggregation
  + Set Title (**M14.4.104**) to “Agg.3.5”
  + Set Description (**M14.4.16**) to “Created as a child of a child”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.3”
* Create a closed child aggregation under **R6.5.1** and **R6.5.2** with the following metadata:
  + Set Class Identifier (**M14.4.4**) to **Cls.3.3** – overriding class inherited from the parent aggregation
  + Set Title (**M14.4.104**) to “Agg.3.5”
  + Set Description (**M14.4.16**) to “Created as a child of a child”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.1”
* Create an open child aggregation under **R6.5.1** and **R6.5.2** with the following metadata:
  + Do not set Class Identifier (**M14.4.4**) – class is inherited from the parent aggregation
  + Set Title (**M14.4.104**) to “Agg.3.6”
  + Set Description (**M14.4.16**) to “Created as a child of a child”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.4”
* Browse the aggregations in the record service under **R6.5.9** and inspect aggregation **Agg.3.1**
* Browse from aggregation **Agg.3.1** to its child aggregation **Agg.3.3** and inspect its metadata under **R6.5.9**
* Browse from aggregation **Agg.3.3** to its sibling aggregation **Agg.3.5** and inspect its metadata under **R6.5.9**
* Browse from aggregation **Agg.3.5** to its parent aggregation **Agg.3.1** and inspect its metadata under **R6.5.9**
* Attempt to modify the Max Levels Of Aggregation (**M14.4.52**) for aggregation **Agg.3.1** under **R6.5.5** back to “0”
* Browse from aggregation **Agg.3.1** to its sibling aggregation **Agg.3.2** in the record service and inspect its metadata under **R6.5.9**
* Attempt to add a Max Levels Of Aggregation (**M14.4.52**) to aggregation **Agg.3.2** under **R6.5.5** setting it to “1”
* Add Max Levels Of Aggregation (**M14.4.52**) to aggregation **Agg.3.2** under **R6.5.5** setting it to “2”
* Browse from aggregation **Agg.3.2** to its child aggregation **Agg.3.4** and inspect its metadata under **R6.5.9**
* Browse from aggregation **Agg.3.4** to its child aggregation **Agg.3.5** and inspect its metadata under **R6.5.9**
* Browse from aggregation **Agg.3.6** to its class **Cls.3.3** and inspect its metadata under **R6.5.9** and **R5.4.7**

### 3.2.6.4 Test case post-conditions (expected result)

* Under **R6.5.5** a child aggregation cannot be created with parent aggregation **Agg.3.1** because the maximum number of levels of aggregation allowed under aggregation **Agg.3.1** was set to zero in test case **T3.2.5**
* Under **R6.5.6** a child aggregation cannot be created with parent aggregation **Agg.3.2** because aggregation **Agg.3.2** was created closed in test case **T3.2.5**
* The Max Levels Of Aggregation (**M14.4.52**) allowed for aggregation **Agg.3.1** is successfully increased to “1”
* Aggregation **Agg.3.2** is successfully opened
* Aggregation **Agg.3.3** is successfully created as a child of aggregation **Agg.3.1**
* Aggregation **Agg.3.4** is successfully created as a child of aggregation **Agg.3.2**
* Under **R6.5.5** a child aggregation cannot be created with parent aggregation **Agg.3.3** because aggregation **Agg.3.3** is a child of root aggregation **Agg.3.1** and the maximum number of levels of aggregation allowed under aggregation **Agg.3.1** was set to one earlier in the test case
* Aggregation **Agg.3.5** is successfully created as a child of aggregation **Agg.3.1** (instead)
* Aggregation **Agg.3.6** is successfully created as a child of aggregation **Agg.3.4** (because there is no limit to the maximum levels of aggregation allowed under its root aggregation **Agg.3.2**)
* Under **R6.5.9** the user is able to browse and inspect:
  + The root aggregations in the record service
  + The child aggregations of a parent aggregation
  + From a child aggregation to its parent aggregation
  + From an aggregation to its class
* Upon inspection, the aggregations **Agg.3.1**, **Agg.3.2** and **Agg.3.4** all have the following additional metadata values (in addition to those listed under test case **T3.2.5**) set when a child record or aggregation is added to an aggregation:
  + First Used Timestamp (**M14.4.32**)
  + Last Addition Timestamp (**M14.4.48**)
* Upon inspection, the aggregations **Agg.3.3**, **Agg.3.4**, **Agg.3.5** and **Agg.3.6** have metadata values similar to those listed under test case **T3.2.5** for root aggregation **Agg.3.1**, however as child aggregations they do not have the following under **R6.5.5**:
  + Max Levels Of Aggregation (**M14.4.52**)
* Upon inspection, the child aggregations **Agg.3.3**, **Agg.3.4**, **Agg.3.5** and **Agg.3.6** have the following additional metadata values not shared by root aggregations **Agg.3.1** and **Agg.3.2** under **R6.5.1**:
  + Parent Aggregation Identifier (**M14.4.63**) – identifies the child’s parent aggregation
  + Aggregated Timestamp (**M14.4.1**) – identifies the precise time at which the child aggregation was added to its parent
* Upon inspection, the child aggregation **Agg.3.5** is created closed and has the metadata value Closed Timestamp (**M14.4.5**) under **R6.5.1** (all other child aggregations are created open)
* Under **R6.5.5**, it is not possible to set the Max Levels Of Aggregation (**M14.4.52**) for a root aggregation to a value smaller than the number of levels of aggregation it already contains, hence:
  + It is not possible to set the Max Levels Of Aggregation for **Agg.3.1** back to “0”
  + It is not possible to set the Max Levels Of Aggregation for **Agg.3.2** to “1”
  + It is possible to set the Max Levels Of Aggregation for **Agg.3.2** to “2”

### 3.2.6.5 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.7 – Test Case

### 3.2.7.1 Test case description

**Modify the metadata of an aggregation**

### 3.2.7.2 Test case preconditions

* **T3.2.6** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.6** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.7.3 Test case steps (instructions)

* Find or browse to aggregation **Agg.3.5** and inspect it under **R6.5.9**
* Modify the metadata of aggregation **Agg.3.5** under **R6.5.3**, as follows:
  + Set Title (**M14.4.104**) to “Agg.3.5 (modified)”
  + Set Description (**M14.4.16**) to “Modified as part of test case T3.2.7”
  + Set Scope Notes (**M14.4.97**) to “This aggregation is closed”
* Inspect the metadata of aggregation **Agg.3.5** under **R6.5.9**

### 3.2.7.4 Test case post-conditions (expected result)

* Aggregation **Agg.3.5** is found
* The metadata of aggregation **Agg.3.5** is modified successfully to the values given above, as shown when the aggregation’s metadata is inspected

### 3.2.7.5 Notes to test centre

Note that aggregation **Agg.3.5** is closed – this should make no difference to this test case

### 3.2.7.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.8 – Test Case

### 3.2.8.1 Test case description

**Delete an aggregation before it has been used**

### 3.2.8.2 Test case preconditions

* **T3.2.7** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.7** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.8.3 Test case steps (instructions)

* Find or browse to aggregation **Agg.3.6** and inspect it under **R6.5.9**
* Delete aggregation **Agg.3.6** under **R6.5.7**
* Browse to aggregation **Agg.3.4** and inspect it under **R6.5.9**, confirming that it has no children
* Attempt to delete aggregation **Agg.3.4** under **R6.5.7**

### 3.2.8.4 Test case post-conditions (expected result)

* Aggregation **Agg.3.6** is found and upon inspection has neither of the following metadata elements:
  + First Used Timestamp (**M14.4.32**)
  + Last Addition Timestamp (**M14.4.48**)
* Aggregation **Agg.3.6** is successfully deleted under **R6.5.7**
* Aggregation **Agg.3.4** is found and upon inspection has the following metadata elements indicating that it has been used:
  + First Used Timestamp (**M14.4.32**)
  + Last Addition Timestamp (**M14.4.48**)
* Aggregation **Agg.3.4** cannot be deleted because, although it no longer has any child entities, it has previously been used for aggregating entities under **R6.5.7**

### 3.2.8.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.9 – Test Case

### 3.2.9.1 Test case description

**Create records (Part 1)**

### 3.2.9.2 Test case preconditions

* **T3.2.8** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.8** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.9.3 Test case steps (instructions)

* Create a record under **R6.5.10** and **R6.5.12** with the following metadata:
  + Set Title (**M14.4.104**) to “Rec.3.1”
  + Set Description (**M14.4.16**) to “Created in Agg.3.3”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.3”
  + Do not set Class Identifier (**M14.4.4**) – class is inherited from the parent aggregation
* Create a record under **R6.5.10** and **R6.5.12** with the following metadata:
  + Set Title (**M14.4.104**) to “Rec.3.2”
  + Set Description (**M14.4.16**) to “Created in Agg.3.3”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.3”
  + Set Class Identifier (**M14.4.4**) to **Cls.3.3** – overriding class inherited from the parent aggregation
* Browse to and inspect the metadata of newly created **Rec.3.1** under **R2.4.22**
* Browse to and inspect the metadata of newly created **Rec.3.2** under **R2.4.22**

### 3.2.9.4 Test case post-conditions (expected result)

* Record **Rec.3.1** is created successfully in parent aggregation **Agg.3.3** and can be found and inspected
* Record **Rec.3.2** is created successfully in parent aggregation **Agg.3.3** and can be found and inspected
* Upon inspection both aggregations have the following metadata values set (from **R6.5.10**) in accordance with the data above:
  + System Identifier (**M14.4.100**)
  + Created Timestamp (**M14.4.9**)
  + Originated Date/Time (**M14.4.61**)
  + Title (**M14.4.104**)
  + Description (**M14.4.16**)
  + Parent Aggregation Identifier (**M14.4.63**)
* Aggregation **Agg.3.2**, but not aggregation **Agg.3.1**, has a value for the following metadata element as given above:
  + Class Identifier (**M14.4.4**)

### 3.2.9.5 Notes to test centre

* The test operator may create any type of record supported by the test system
* However, if the test system supports records with multiple components then at least record **Rec.3.1** must be created with more than one component (see test case **T3.2.21**)
* (The supplier must stipulate what different types of record are supported by the test system when answering the pre-qualification questionnaire)

### 3.2.9.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.10 – Test Case

### 3.2.10.1 Test case description

**Close and open aggregations**

### 3.2.10.2 Test case preconditions

* **T3.2.9** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.9** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.10.3 Test case steps (instructions)

* Browse from record **Rec.3.2** to its parent aggregation **Agg.3.3** and inspect it under **R6.5.17**
* Browse from record **Agg.3.3** to its parent aggregation **Agg.3.1** and inspect it under **R6.5.17**
* Attempt to create a record (see the previous test case **T3.2.9**) in root aggregation **Agg.3.1** under **R6.5.10**
* Attempt to close aggregation **Agg.3.1** under **R6.5.6**
* Close aggregation **Agg.3.3** followed by (or with) **Agg.3.1** under **R6.5.6**
* Attempt to create a record (see the previous test case **T3.2.9**) in closed aggregation **Agg.3.3** under **R6.5.10**
* Attempt to open closed aggregation **Agg.3.5** under **R6.5.6**
* Open root aggregation **Agg.3.1** under **R6.5.6 (**leaving child aggregations **Agg.3.3** and **Agg.3.5** closed**)**
* Attempt to create a record (see the previous test case **T3.2.9**) in closed aggregation **Agg.3.5** under **R6.5.10**
* Open aggregation **Agg.3.5** under **R6.5.6**

### 3.2.10.4 Test case post-conditions (expected result)

* Aggregation **Agg.3.3** is found, is open, and is a child of aggregation **Agg.3.1**
* Aggregation **Agg.3.1** is found, is open, and is a root aggregation containing child aggregations **Agg.3.3** and **Agg.3.5**
* A record cannot be created in aggregation **Agg.3.1** under **R6.5.10** because it contains child aggregations (see the rationale to **R6.5.10**)
* Aggregation **Agg.3.1** cannot be closed under **R6.5.6** because it contains an open child aggregation **Agg.3.3** (the other child aggregation **Agg.3.5** is closed)
* Only when aggregation **Agg.3.3** is closed may its parent aggregation **Agg.3.1** be closed (the test system may or may not include functionality to close both parent and all descendant aggregations simultaneously in a single operation)
* Aggregations **Agg.3.3** and **Agg.3.1** are successfully closed
* Once aggregation **Agg.3.3** is closed a record cannot be created in this parent aggregation
* Closed aggregation **Agg.3.5** cannot be opened while its parent aggregation **Agg.3.1** is closed
* Only when aggregation **Agg.3.1** is opened may its child aggregation **Agg.3.5** be opened
* Aggregation **Agg.3.1** is successfully opened (its child aggregations **Agg.3.3** and **Agg.3.5** remain closed)
* A record still cannot be created in closed aggregation **Agg.3.5** under **R6.5.10**
* Aggregation **Agg.3.5** is successfully opened (its sibling aggregation **Agg.3.3** remains closed)

### 3.2.10.5 Notes to test centre

Include in the observations whether each child of a parent aggregation must be closed individually before its parent aggregation can be, under **R6.5.6**, or whether the test system supports closing of an aggregation and its descendant classes simultaneously as a single operation (see the rationale to **R6.5.6**)

### 3.2.10.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.11 – Test Case

### 3.2.11.1 Test case description

**Create records (Part 2)**

### 3.2.11.2 Test case preconditions

* **T3.2.10** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.10** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.11.3 Test case steps (instructions)

* Create a record under **R6.5.10** and **R6.5.12** with the following metadata:
  + Set Title (**M14.4.104**) to “Rec.3.3”
  + Set Description (**M14.4.16**) to “Created in Agg.3.5”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.5”
  + Do not set Class Identifier (**M14.4.4**) – class is inherited from the parent aggregation
* Create a record under **R6.5.10** and **R6.5.12** with the following metadata:
  + Set Title (**M14.4.104**) to “Rec.3.4”
  + Set Description (**M14.4.16**) to “Created in Agg.3.5”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.5”
  + Set Class Identifier (**M14.4.4**) to **Cls.3.2** – overriding class inherited from the parent aggregation
* Create a record under **R6.5.10** and **R6.5.12** with the following metadata:
  + Set Title (**M14.4.104**) to “Rec.3.5”
  + Set Description (**M14.4.16**) to “Created in Agg.3.4”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.4”
  + Do not set Class Identifier (**M14.4.4**) – class is inherited from the parent aggregation
* Create a record under **R6.5.10** and **R6.5.12** with the following metadata:
  + Set Title (**M14.4.104**) to “Rec.3.6”
  + Set Description (**M14.4.16**) to “Created in Agg.3.4”
  + Set Parent Aggregation Identifier (**M14.4.63**) to “Agg.3.4”
  + Set Class Identifier (**M14.4.4**) to **Cls.3.1** – overriding class inherited from the parent aggregation
* Browse to and inspect the metadata of newly created **Rec.3.3** under **R2.4.22**
* From **Rec.3.3** browse to and inspect its class under **R6.5.17**
* From **Rec.3.3** browse to and inspect its disposal schedule under **R6.5.17**
* Browse to and inspect the metadata of newly created **Rec.3.4** under **R2.4.22**
* From **Rec.3.4** browse to and inspect its class under **R6.5.17**
* From **Rec.3.4** browse to and inspect its disposal schedule under **R6.5.17**
* Browse to and inspect the metadata of newly created **Rec.3.5** under **R2.4.22**
* From **Rec.3.5** browse to and inspect its class under **R6.5.17**
* From **Rec.3.5** browse to and inspect its disposal schedule under **R6.5.17**
* Browse to and inspect the metadata of newly created **Rec.3.6** under **R2.4.22**
* From **Rec.3.6** browse to and inspect its class under **R6.5.17**
* From **Rec.3.6** browse to and inspect its disposal schedule under **R6.5.17**

### 3.2.11.4 Test case post-conditions (expected result)

* Record **Rec.3.3** is successfully created and may be found and inspected with the following class and disposal schedule:
  + Class is **Cls.3.3** (inherited from parent aggregation)
  + Disposal schedule is **DiS.3.3** (inherited from class under **R6.5.14**)
* Record **Rec.3.4** is successfully created and may be found and inspected with the following class and disposal schedule:
  + Class is **Cls.3.2** (overriding the inherited class)
  + Disposal schedule is **DiS.3.2** (inherited from class under **R6.5.14**)
* Record **Rec.3.5** is successfully created and may be found and inspected with the following class and disposal schedule:
  + Class is **Cls.3.3** (inherited from parent aggregation)
  + Disposal schedule is **DiS.3.3** (inherited from class under **R6.5.14**)
* Record **Rec.3.6** is successfully created and may be found and inspected with the following class and disposal schedule:
  + Class is **Cls.3.1** (overriding the inherited class)
  + Disposal schedule is **DiS.3.1** (inherited from class under **R6.5.14**)

### 3.2.11.5 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.12 – Test Case

### 3.2.12.1 Test case description

**Override the inherited disposal schedule**

### 3.2.12.2 Test case preconditions

* **T3.2.11** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.11** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.12.3 Test case steps (instructions)

* Search under **R6.5.18** for all aggregations and records with class **Cls.3.1**
* From the search results inspect record **Rec.3.6**
* Under **R6.5.17** browse from record **Rec.3.6** to its disposal schedule and confirm that it is **DiS.3.1** and is inherited from class **Cls.3.1**
* Under **R6.5.15** apply disposal schedule **DiS.3.3** directly to record **Rec.3.6** overriding its inherited disposal schedule
* Search under **R6.5.18** for all records with disposal schedule **DiS.3.3**
* From the search results inspect record **Rec.3.5**
* Under **R6.5.15** apply disposal schedule **DiS.3.4** directly to record **Rec.3.5** overriding its inherited disposal schedule

### 3.2.12.4 Test case post-conditions (expected result)

* The search for aggregations and records with class **Cls.3.1** must produce the following search results:
  + Aggregation **Agg.3.1** (class is applied directly)
  + Aggregation **Agg.3.3** (class is inherited)
  + Record **Rec.3.1** (class is inherited)
  + Record **Rec.3.6** (class is applied directly)
* Record **Rec.3.6** is found to inherit disposal schedule **DiS.3.1** from its class
* The disposal schedule for record **Rec.3.6** is successfully changed to **DiS.3.3**, overriding its inherited disposal schedule under **R6.5.15**
* The search for records with disposal schedule **DiS.3.3** must produce the following search results:
  + Record **Rec.3.2** (inherited from its class)
  + Record **Rec.3.3** (inherited from its class)
  + Record **Rec.3.5** (inherited from its class)
  + Record **Rec.3.6** (overriding the disposal schedule of its class)
* The disposal schedule for record **Rec.3.5** is successfully changed to **DiS.3.4**, overriding its inherited disposal schedule (previously **DiS.3.3**) under **R6.5.15**

### 3.2.12.5 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.13 – Test Case

### 3.2.13.1 Test case description

**Change default disposal schedule for a class**

### 3.2.13.2 Test case preconditions

* **T3.2.12** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.12** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.13.3 Test case steps (instructions)

* Search under **R6.5.18** for all records with disposal schedule **DiS.3.4**
* In the classification service, under **R5.4.4**, change the default disposal schedule for class **Cls.3.1** to disposal schedule **DiS.3.4**
* Search under **R6.5.18** for all records with disposal schedule **DiS.3.4**
* From the search results inspect record **Rec.3.1**
* Under **R6.5.15** apply disposal schedule **DiS.3.1** directly to record **Rec.3.1** overriding its inherited disposal schedule

### 3.2.13.4 Test case post-conditions (expected result)

* The search for records with disposal schedule **DiS.3.4** must produce the following search results:
  + Record **Rec.3.5** (overriding the disposal schedule of its class)
* The default disposal schedule for class **Cls.3.1** is successfully changed to **DiS.3.4**
* The search for records with disposal schedule **DiS.3.4** must produce the following search results:
  + Record **Rec.3.1** (inherited from its class)
  + Record **Rec.3.5** (overriding the disposal schedule of its class)
* The disposal schedule for record **Rec.3.1** is successfully changed to **DiS.3.1**, overriding its inherited disposal schedule (previously **DiS.3.4**) under **R6.5.15**

### 3.2.13.5 Notes to test centre

This test case provides an additional test for classification requirement **R5.4.4**

### 3.2.13.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.14 – Test Case

### 3.2.14.1 Test case description

**Replace one class with another for all classified entities**

### 3.2.14.2 Test case preconditions

* **T3.2.13** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.13** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.14.3 Test case steps (instructions)

* Search under **R6.5.18** for all aggregations and records with class **Cls.3.2**
* In the classification service, under **R5.4.8**, replace class **Cls.3.1** with class **Cls.3.2** for all aggregations and records classified by class **Cls.3.1**
* Search under **R6.5.18** for all aggregations and records with class **Cls.3.2**
* Search under **R6.5.18** for all records with disposal schedule **DiS.3.2**

### 3.2.14.4 Test case post-conditions (expected result)

* The first search for aggregations and records with class **Cls.3.2** must produce the following search results:
  + Aggregation **Agg.3.2** (class is applied directly)
  + Record **Rec.3.4** (class is applied directly)
* Class **Cls.3.1** is successfully replaced with class **Cls.3.2** for all aggregations and records under **R5.4.8**
* The second search for aggregations and records with class **Cls.3.2** must produce the following search results:
  + Aggregation **Agg.3.1** (class is applied directly)
  + Aggregation **Agg.3.2** (class is applied directly)
  + Aggregation **Agg.3.3** (class is inherited)
  + Record **Rec.3.1** (class is inherited)
  + Record **Rec.3.4** (class is applied directly)
  + Record **Rec.3.6** (class is applied directly)
* The third search for records with disposal schedule **DiS.3.2** must produce the following search results:
  + Record **Rec.3.4** (inherited from its class)

### 3.2.14.5 Notes to test centre

This test case provides an additional test for classification requirement **R5.4.8**

### 3.2.14.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.15 – Test Case

### 3.2.15.1 Test case description

**Reclassify records**

### 3.2.15.2 Test case preconditions

* **T3.2.14** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.14** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.15.3 Test case steps (instructions)

* Search or browse to record **Rec.3.1** and inspect it under **R6.5.17**
* Browse from record **Rec.3.1** to its class and its disposal schedule under **R6.5.17**
* Under **R6.5.12** reclassify record **Rec.3.1** by directly assigning it class **Cls.3.1** and thereby overriding the class it inherits by default from its parent aggregation (**Cls.3.2** from root aggregation **Agg.3.1**)
* Ensure that it is possible to browse from record **Rec.3.1** to its new class and its disposal schedule under **R6.5.17**
* Search or browse to record **Rec.3.2** and inspect it under **R6.5.17**
* Browse from record **Rec.3.2** to its class and its disposal schedule under **R6.5.17**
* Under **R6.5.12** reclassify record **Rec.3.2** by removing its assigned class (**Cls.3.3**) and thereby allowing it to inherit class **Cls.3.2** from its root aggregation (**Agg.3.1**) via its parent aggregation (**Agg.3.3**)
* Ensure that it is possible to browse from record **Rec.3.2** to its new class and its disposal schedule under **R6.5.17**
* Search or browse to record **Rec.3.4** and inspect it under **R6.5.17**
* Browse from record **Rec.3.4** to its class and its disposal schedule under **R6.5.17**
* Under **R6.5.12** reclassify record **Rec.3.4** by directly assigning to it class **Cls.3.1** and replacing the class that was previously directly assigned to it (**Cls.3.2**)
* Ensure that it is possible to browse from record **Rec.3.4** to its new class and its disposal schedule under **R6.5.17**

### 3.2.15.4 Test case post-conditions (expected result)

* Record **Rec.3.1** is found and has the following class and disposal schedule:
  + Class **Cls.3.2** – inherited
  + Disposal schedule **DiS.3.1** – assigned
* Record **Rec.3.1** is reclassified by directly assigning class **Cls.3.1** and now has:
  + Class **Cls.3.1** – assigned
  + Disposal schedule **DiS.3.1** – assigned
* Record **Rec.3.2** is found and has the following class and disposal schedule:
  + Class **Cls.3.3** – assigned
  + Disposal schedule **DiS.3.3** – inherited
* Record **Rec.3.2** is reclassified by removing the directly assigned class (**Cls.3.3**) and inheriting its parent class, it now has:
  + Class **Cls.3.2** – inherited
  + Disposal schedule **DiS.3.2** – inherited
* Record **Rec.3.4** is found and has the following class and disposal schedule:
  + Class **Cls.3.2** – assigned
  + Disposal schedule **DiS.3.2** – inherited
* Record **Rec.3.4** is reclassified by removing the directly assigned class (**Cls.3.2**) and replacing it with a new directly assigned class **Cls.3.1**, it now has:
  + Class **Cls.3.1** – assigned
  + Disposal schedule **DiS.3.4** – inherited

### 3.2.15.5 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.16 – Test Case

### 3.2.16.1 Test case description

**Reclassify aggregations**

### 3.2.16.2 Test case preconditions

* **T3.2.15** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.15** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.16.3 Test case steps (instructions)

* Search or browse to aggregation **Agg.3.3** and inspect it under **R6.5.9**
* Under **R6.5.4** reclassify aggregation **Agg.3.3** by directly assigning class **Cls.3.3** and overriding the class (**Cls.3.2**) it previously inherited from its parent aggregation (**Agg.3.1**)
* Browse from parent aggregation **Agg.3.3** to the records it contains and inspect them under **R6.5.17**
* For each child record of aggregation **Agg.3.3** browse under **R6.5.17** to the record’s class and then to the record’s disposal schedule
* Search or browse to aggregation **Agg.3.4** and inspect it under **R6.5.9**
* Under **R6.5.4** reclassify aggregation **Agg.3.4** by removing its directly assigned class (**Cls.3.3**) and allowing it to inherit its parent aggregation’s class (**Cls.3.2**)
* Browse from parent aggregation **Agg.3.4** to its records and inspect them under **R6.5.17**
* For each child record of aggregation **Agg.3.4** browse under **R6.5.17** to the record’s class and then to the record’s disposal schedule
* Search or browse to root aggregation **Agg.3.2** and inspect it under **R6.5.9**
* Under **R6.5.4** reclassify aggregation **Agg.3.2** by replacing its previously directly assigned class (**Cls.3.2**) with class **Cls.3.1** as a new directly assigned class
* Browse from root aggregation **Agg.3.2** to its child aggregation **Agg.3.4** and then to the records in aggregation **Agg.3.4** and inspect them under **R6.5.17**
* For each child record of aggregation **Agg.3.4** browse under **R6.5.17** to the record’s class and then to the record’s disposal schedule
* Under **R6.5.14** and **R6.5.15** remove the disposal schedule (**DiS.3.3**) directly assigned to record **Rec.3.6** and allow it to inherit disposal schedule **DiS.3.2** from its class

### 3.2.16.4 Test case post-conditions (expected result)

* Aggregation **Agg.3.3** is found and its class is successfully changed to **Cls.3.3**
* Aggregation **Agg.3.3** contains the following records:
  + Record **Rec.3.1**
    - Class **Cls.3.1** – assigned
    - Disposal schedule **DiS.3.1** – assigned
  + Record **Rec.3.2**
    - Class **Cls.3.3** – inherited
    - Disposal schedule **DiS.3.3** – inherited
* Aggregation **Agg.3.4** is found and its directly assigned class is successfully removed allowing it to inherit **Cls.3.2**
* Aggregation **Agg.3.4** contains the following records:
  + Record **Rec.3.5**
    - Class **Cls.3.2** – inherited
    - Disposal schedule **DiS.3.4** – assigned
  + Record **Rec.3.6**
    - Class **Cls.3.2** – assigned
    - Disposal schedule **DiS.3.3** – assigned
* Root aggregation **Agg.3.2** is found and its class is successfully changed to **Cls.3.1**
* Aggregation **Agg.3.4** (the child aggregation of **Agg.3.2**) now contains the following records:
  + Record **Rec.3.5**
    - Class **Cls.3.1** – inherited
    - Disposal schedule **DiS.3.4** – assigned
  + Record **Rec.3.6**
    - Class **Cls.3.2** – assigned
    - Disposal schedule **DiS.3.3** – assigned
* Disposal schedule **DiS.3.3** is removed from its direct association with record **Rec.3.6**, allowing record **Rec.3.6** to inherit disposal schedule **DiS.3.2** from its class (**Cls.3.2**) instead

### 3.2.16.5 Notes to test centre

Note the difference between record **Rec.3.5** and **Rec.3.6** in aggregation **Agg.3.4**. At one point in the test case both records seem to share the same class (**Cls.3.2**). However, the inherited class of record **Rec.3.5** changes with changes to its parent aggregation. Because record **Rec.3.6** has its class directly assigned it does not change when the class of its aggregation changes – even when it changes to the same class as itself.

### 3.2.16.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.17 – Test Case

### 3.2.17.1 Test case description

**Move records**

### 3.2.17.2 Test case preconditions

* **T3.2.16** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.16** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.17.3 Test case steps (instructions)

* Search or browse to record **Rec.3.5** and inspect it under **R6.5.17**
* Attempt to move record **Rec.3.5** to aggregation **Agg.3.1** under **R6.5.13**
* Attempt to move record **Rec.3.5** to aggregation **Agg.3.3** under **R6.5.13**
* Move record **Rec.3.5** to aggregation **Agg.3.5** while retaining its previous classification (**Cls.3.1**) and applying it directly to the record under **R6.5.13**
* Search or browse to record **Rec.3.6** and inspect it under **R6.5.17**
* Move record **Rec.3.6** to aggregation **Agg.3.5** while retaining its previous classification (**Cls.3.2**) under **R6.5.13**
* Close aggregation **Agg.3.4** under **R6.5.6**
* Search or browse to record **Rec.3.1** and inspect it under **R6.5.17**
* Move record **Rec.3.1** to aggregation **Agg.3.5** electing not to retain its previous class (**Cls.3.1**) but instead to inherit the class (**Cls.3.2**) of its new parent aggregation under **R6.5.13**
* Browse and inspect the records in aggregation **Agg.3.5** under **R6.5.17** noting for each record its class, disposal schedule and Aggregated Timestamp (**M14.4.4**)
* Inspect aggregation **Agg.3.5** and note the following metadata values:
  + First Used Timestamp (**M14.4.32**)
  + Last Addition Timestamp (**M14.4.48**)

### 3.2.17.4 Test case post-conditions (expected result)

* Record **Rec.3.5** is found in parent aggregation **Agg.3.4**
* Record **Rec.3.5** cannot be moved to root aggregation **Agg.3.1** under **R6.5.13** because it contains aggregations
* Record **Rec.3.5** cannot be moved to aggregation **Agg.3.3** under **R6.5.13** because it is a closed aggregation
* Record **Rec.3.5** is moved successfully to aggregation **Agg.3.5** with class **Cls.3.1** directly applied
* Record **Rec.3.6** is found in parent aggregation **Agg.3.4**
* Record **Rec.3.6** is moved successfully to aggregation **Agg.3.5** with class **Cls.3.2** directly applied
* Aggregation **Agg.3.4** is successfully closed
* Record **Rec.3.1** is found in parent aggregation **Agg.3.3**
* Record **Rec.3.1** is moved successfully to aggregation **Agg.3.5** and inherits class **Cls.3.2** from its new parent aggregation
* On inspection, the records in aggregation **Agg.3.5** are:
  + Record **Rec.3.1**
    - Class **Cls.3.3** – inherited
    - Disposal schedule **DiS.3.1** – assigned
  + Record **Rec.3.3**
    - Class **Cls.3.3** – inherited
    - Disposal schedule **DiS.3.3** – inherited
  + Record **Rec.3.4**
    - Class **Cls.3.1** – assigned
    - Disposal schedule **DiS.3.4** – inherited
  + Record **Rec.3.5**
    - Class **Cls.3.1** – assigned
    - Disposal schedule **DiS.3.4** – assigned
  + Record **Rec.3.6**
    - Class **Cls.3.2** – assigned
    - Disposal schedule **DiS.3.2** – inherited
* In order of Aggregated Timestamp, the records aggregation **Agg.3.5** should be:
  + Record **Rec.3.3** (Aggregated Timestamp for this record must correlate to the First Used Timestamp for the aggregation)
  + Record **Rec.3.4**
  + Record **Rec.3.5**
  + Record **Rec.3.6**
  + Record **Rec.3.1** (Aggregated Timestamp for this record must correlate to the Last Addition Timestamp for the aggregation)

### 3.2.17.5 Notes to test centre

Note that although the user is not allowed to move records into a closed aggregation (in this example attempting to move record **Rec.3.5** out of its parent **Agg.3.4** and into the closed aggregation **Agg.3.3**) the user is able to move records out of a closed aggregation (in this example moving record **Rec.3.1** out of **Agg.3.3** and into the new parent **Agg.3.5**).

### 3.2.17.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.18 – Test Case

### 3.2.18.1 Test case description

**Browse records by Originated Date/Time**

### 3.2.18.2 Test case preconditions

* **T3.2.17** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.17** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.18.3 Test case steps (instructions)

* Browse and inspect the records in aggregation **Agg.3.5** in order of their Originated Date/Time under **R6.5.17** noting for each record its Originated Date/Time (**M14.4.61**)
* Under **R2.4.26** attempt to set the Originated Date/Time for record **Rec.3.1** so that it is after the Originated Date/Time of record **Rec.3.3** and before the Originated Date/Time of record **Rec.3.4**
* Under **R2.4.26** set the Originated Date/Time for record **Rec.3.5** so that it is after the Originated Date/Time of record **Rec.3.3** and before the Originated Date/Time of record **Rec.3.4**
* Under **R2.4.26** set the Originated Date/Time for record **Rec.3.6** so that it is before the Originated Date/Time of record **Rec.3.1**
* Browse and inspect the records in aggregation **Agg.3.5** in order of their Originated Date/Time under **R6.5.17**

### 3.2.18.4 Test case post-conditions (expected result)

* The order of records in aggregation **Agg.3.5** by Originated Date/Time is:
  + Record **Rec.3.1**
  + Record **Rec.3.3**
  + Record **Rec.3.4**
  + Record **Rec.3.5**
  + Record **Rec.3.6**
* The Originated Date/Time for record **Rec.3.1** cannot be changed to after that of record **Rec.3.3** because in accordance with **R2.4.26** the Originated Date/Time of an entity cannot be later than its Created Timestamp in the MCRS (record **Rec.3.3** is created in test case **T3.2.11** after record **Rec.3.1** has already been created in test case **T3.2.9** and by default the Originated Date/Time of record **Rec.3.3** will reflect its Created Timestamp under **R2.4.25**)
* The Originated Date/Time for record **Rec.3.5** is successfully set earlier than record **Rec.3.4** and later than record **Rec.3.3**
* The Originated Date/Time for record **Rec.3.6** is successfully set earlier than record **Rec.3.1**
* The order of records in aggregation **Agg.3.5** when browsed in order of Originated Date/Time is (now):
  + Record **Rec.3.6**
  + Record **Rec.3.1**
  + Record **Rec.3.3**
  + Record **Rec.3.5**
  + Record **Rec.3.4**

### 3.2.18.5 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.19 – Test Case

### 3.2.19.1 Test case description

**Move aggregations**

### 3.2.19.2 Test case preconditions

* **T3.2.18** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.18** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.19.3 Test case steps (instructions)

* Search or browse to aggregation **Agg.3.3** and inspect it under **R6.5.9**
* Attempt to move aggregation **Agg.3.3** so that it becomes a child of aggregation **Agg.3.4** under **R6.5.8**
* Open aggregation **Agg.3.4** under **R6.5.6**
* Move aggregation **Agg.3.3** to become a child of aggregation **Agg.3.4** under **R6.5.8** so that it inherits the class of its new parent (**Cls.3.1**) and no longer retains its own directly applied class (**Cls.3.3**)
* Search or browse to record **Rec.3.2** and inspect its class and disposal schedule under **R6.5.17**
* Search or browse to aggregation **Agg.3.1** and inspect it under **R6.5.9**
* Attempt to move aggregation **Agg.3.1** so that it becomes a child of aggregation **Agg.3.4** under **R6.5.8**
* Move aggregation **Agg.3.1** to become a child of aggregation **Agg.3.2** under **R6.5.8** while retaining its original directly applied class (**Cls.3.2**)
* Search or browse to aggregation **Agg.3.4** and inspect it under **R6.5.9**
* Move aggregation **Agg.3.4** to become a root aggregation under **R6.5.8**
* Do not set Max Levels Of Aggregation for the new root aggregation **Agg.3.4**
* Search or browse to aggregation **Agg.3.3** and inspect it under **R6.5.9**
* Open aggregation **Agg.3.3** under **R6.5.6**
* Search or browse to aggregation **Agg.3.5** and inspect it under **R6.5.9**
* Attempt to move aggregation **Agg.3.5** so that it becomes a child of aggregation **Agg.3.3** under **R6.5.8**
* Move aggregation **Agg.3.5** to become a root aggregation under **R6.5.8**

### 3.2.19.4 Test case post-conditions (expected result)

* Aggregation **Agg.3.3** is found and is closed
* Aggregation **Agg.3.3** cannot be moved to become a child of aggregation **Agg.3.4** because **Agg.3.4** is closed
* Aggregation **Agg.3.4** is successfully opened
* Aggregation **Agg.3.3** is successfully moved to become a child of aggregation **Agg.3.4**, inheriting class **Cls.3.1**
* Record **Rec.3.2** is found and has the following:
  + Class **Cls.3.1** – inherited
  + Disposal schedule **DiS.3.4** – inherited
* Aggregation **Agg.3.1** is found and is a root aggregation
* Aggregation **Agg.3.1** cannot be moved to become a child aggregation of **Agg.3.4** because this would exceed the Max Levels Of Aggregation for root aggregation **Agg.3.2** (set to “2” in test case **T3.2.6**)
* Aggregation **Agg.3.1** is successfully moved to become a child of aggregation **Agg.3.4**, while retaining class **Cls.3.2**
* Aggregation **Agg.3.4** is found and inherits class **Cls.3.1** from its parent aggregation **Agg.3.2**
* Aggregation **Agg.3.4** is successfully moved to become a root aggregation under **R6.5.8**:
  + Its previous class **Cls.3.1** is directly applied (automatically)
  + Max Levels Of Aggregation is not set (meaning unlimited levels below the root)
* Aggregation **Agg.3.3** is found and is (still) closed
* Aggregation **Agg.3.3** is successfully opened
* Aggregation **Agg.3.5** is found, is open, and contains 5 records
* Aggregation **Agg.3.5** cannot be moved to become a child of (open and active) aggregation **Agg.3.3** because **Agg.3.3** contains a record (**Rec.3.2**)
* Aggregation **Agg.3.5** is successfully moved to become a root aggregation under **R6.5.8** with the same directly applied class (**Cls.3.3**)

### 3.2.19.5 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.20 – Test Case

### 3.2.20.1 Test case description

**Modify record metadata**

### 3.2.20.2 Test case preconditions

* **T3.2.19** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.19** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.20.3 Test case steps (instructions)

* Search or browse to record **Rec.3.1** and inspect it under **R6.5.17**
* Under **R6.5.11** modify the metadata of record **Rec.3.1**:
  + Set Title (**M14.4.104**) to “Rec.3.1 (modified)”
  + Set Description (**M14.4.16**) to “Metadata modified in test case T3.2.20”

### 3.2.20.4 Test case post-conditions (expected result)

* The Title and Description of record **Rec.3.1** are successfully updated to the values given above

### 3.2.20.5 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.21 – Test Case

### 3.2.21.1 Test case description

**Modify component metadata**

### 3.2.21.2 Test case preconditions

* **T3.2.20** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.20** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.21.3 Test case steps (instructions)

* From record **Rec.3.1** browse its component(s) under **R6.5.17**
* Inspect a component of record **Rec.3.1** under **R6.5.17** and ensure that its metadata elements and values match those listed in **R6.5.19**, in particular:
  + The Created Timestamp (**M14.4.9**) should match closely the Created Timestamp of the record **Rec.3.1**
  + The Originated Date/Time (**M14.4.61**) should match the Originated Date/Time of the record **Rec.3.1**
  + The Title (**M14.4.104**) and Description (**M14.4.16**) may:
    - Match the Title and Description of record **Rec.3.1**
    - Have been generated automatically on record creation
    - Have been specified separately when the record was created in test case **T3.2.9**
* Under **R6.5.20** modify the metadata of the component:
  + Set Title (**M14.4.104**) to “Component of record Rec.3.1”
  + Set Description (**M14.4.16**) to “Metadata modified in MoReq2010® Test Module 3: Records”
* When modifying the metadata of the component include the comment “Modifying component metadata in accordance with test case T3.2.21”, under **R2.4.18**

### 3.2.21.4 Test case post-conditions (expected result)

* The user can browse from record **Rec.3.1** to its components, browse the components and browse from the components of a record to their record entity as described in **R6.5.17**
* The metadata of the inspected component match those of record **Rec.3.1**
* The Title and Description of the record component are successfully updated to the values given above
* A comment is provided when the Title and Description are updated under **R2.4.18**

### 3.2.21.5 Notes to test centre

* Record **Rec.3.1** is created in test case **T3.2.9** and should have more than one component if the test system is capable of supporting multiple components (see **3.2.9.5 Notes to test centre**)
* Some specialised types of MCRS may only support records with one component
* The number of components contained in record **Rec.3.1** should be noted in the test centre observations
* The way in which component metadata is generated and modified should be noted in the test centre observations

### 3.2.21.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.22 – Test Case

### 3.2.22.1 Test case description

**Inspect the event history of a component and a record**

### 3.2.22.2 Test case preconditions

* **T3.2.21** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.21** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.22.3 Test case steps (instructions)

* Under **R2.4.19**, inspect the event history of the same component of record **Rec.3.1** that was inspected (and its metadata modified) in the previous test case **T3.2.21**
* Ensure that the component event history contains appropriate events for each of the following functions:
  + Component – Create (**F14.5.38**) from test case **T3.2.9**
  + Component – Inspect (**F14.5.44**) from test case **T3.2.21**
  + Component – Modify Metadata (**F14.5.46**) from test case **T3.2.21**
* Browse from the component to its record **Rec.3.1** under **R6.5.17**
* Under **R2.4.19**, inspect the event history of record **Rec.3.1** and ensure that the same events listed in the event history of the component also appear in the event history of the record under **R6.5.21**

### 3.2.22.4 Test case post-conditions (expected result)

* The component event history contains the events listed above
* The Component Create (**F14.5.38**) and Component Modify Metadata (**F14.5.46**) events contain appropriate Metadata Change Entries (**D14.3.3**) under **R2.4.17**
* The comment provided under **R2.4.18** in the previous test case **T3.2.21** is included in the Component Modify Metadata (**F14.5.46**) event
* The user can browse between the record, its components, and their respective event histories under **R6.5.17**
* The same events listed in the component event history are also listed in the event history for record **Rec.3.1**

### 3.2.22.5 Notes to test centre

* If necessary the test operator can ensure that the events in the event history of the component are the same (identical) events as those listed in the event history of the record by comparing the unique System Identifier (**M14.4.100**) for each event
* The same events should appear in each event history – not identical copies – but the the same events

### 3.2.22.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.23 – Test Case

### 3.2.23.1 Test case description

**Duplicate records**

### 3.2.3.2 Test case preconditions

* **T3.2.22** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.22** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.23.3 Test case steps (instructions)

* Make a duplicate of record **Rec.3.1** under **R6.5.16**
* Modify the metadata of one of the duplicates of record **Rec.3.1** and set its Title (**M14.4.104**) to “Rec.3.1a”
* Modify the metadata of the other duplicate and set its Title (**M14.4.104**) to “Rec.3.1b”
* Inspect record **Rec.3.1a** and **Rec.3.1b** and compare their respective metadata elements and values
* Ensure that record **Rec.3.1a** has a Duplicate Identifier (**M14.4.23**) that is set to record **Rec.3.1b**
* Ensure that record **Rec.3.1b** has a Duplicate Identifier (**M14.4.23**) that is set to record **Rec.3.1a**
* Inspect the event histories of record **Rec.3.1a** and **Rec.3.1b** under **R2.4.19** and compare the corresponding events from each event history
* Under **R6.5.17** browse and inspect each of the components of records **Rec.3.1a** and **Rec.3.1b** and compare their respective metadata elements and values
* Make a duplicate of record **Rec.3.1b** under **R6.5.16**
* Modify the metadata of one of the duplicates of record **Rec.3.1b** and set its Title (**M14.4.104**) to “Rec.3.1b(i)”
* Modify the metadata of the other duplicate and set its Title (**M14.4.104**) to “Rec.3.1b(ii)”
* Inspect each of the records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** and compare their respective metadata elements and values
* Ensure that record **Rec.3.1a** has a Duplicate Identifier (**M14.4.23**) that is set to record **Rec.3.1b(i)** and another Duplicate Identifier that is set to record **Rec.3.1b(ii)**
* Ensure that record **Rec.3.1b(i)** has a Duplicate Identifier (**M14.4.23**) that is set to record **Rec.3.1a** and another Duplicate Identifier that is set to record **Rec.3.1b(ii)**
* Ensure that record **Rec.3.1b(ii)** has a Duplicate Identifier (**M14.4.23**) that is set to record **Rec.3.1a** and another Duplicate Identifier that is set to record **Rec.3.1b(i)**
* Inspect the event histories of the records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** under **R2.4.19** and compare the corresponding events from each event history
* Under **R6.5.17** browse and inspect each of the components of the records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** and compare their respective metadata elements and values

### 3.2.23.4 Test case post-conditions (expected result)

* Record **Rec.3.1** is successfully duplicated and both duplicate records have the same Title (previously set to “Rec.3.1 (modified)” in test case **T3.2.20**)
* The metadata of the two duplicate records is successfully modified so that they can be distinguished from one another based on Title
* On inspection both records have exactly the same metadata elements with exactly the same values, except:
  + The System Identifier (**M14.4.100**) for each duplicate record is different, and
  + The Duplicate Identifier (**M14.4.23**) for each record is set to indicate the other duplicate record
* On inspection of their event histories, both record **Rec.3.1a** and record **Rec.3.1b** must have exactly the same events that occurred with exactly the same timestamp and the same metadata elements and values, except:
  + The System Identifier (**M14.4.100**) for each duplicate event is different,
  + The Duplicate Identifier (**M14.4.23**) for each event is set to indicate the other duplicate event,
  + The Participating Record Identifier (**M14.4.77**) refers either to record **Rec.3.1a** or **Rec.3.1b**, and
  + Where the event refers to a component, each duplicate record’s events will refer to its own components in the Participating Component Identifier (**M14.4.66**)
* Note that the event histories for each duplicate record only contain duplicate events up to and including the Record – Duplicate (**F14.5.126**) event itself – all events after that (such as modifying the Title of the record following duplication) belong only to one record or the other
* On inspection of their components, both record **Rec.3.1a** and **Rec.3.1b** must each have exactly the same components with exactly the same metadata elements and the same values, except:
  + The System Identifier (**M14.4.100**) for each duplicate component is different,
  + The Record Identifier (**M14.4.86**) refers to either record **Rec.3.1a** or **Rec.3.1b**, and
  + The Duplicate Identifier (**M14.4.23**) for each component is set to indicate the other duplicate component
* Record **Rec.3.1b** is successfully duplicated (and becomes two records both initially with the Title “Rec.3.1b”)
* The metadata of the two duplicate records of **Rec.3.1.b** is successfully modified so that they can be distinguished from one another based on Title
* On inspection, all three records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** have exactly the same metadata elements with exactly the same values, except that:
  + The System Identifier (**M14.4.100**) for each duplicate record is different, and
  + Each record has two Duplicate Identifier (**M14.4.23**) elements each of which indicates one of the other two duplicate records
* On inspection of their event histories, all three records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** must have exactly the same previous events with exactly the same timestamp and the same metadata elements and values, except:
  + The System Identifier (**M14.4.100**) for each duplicate event is different,
  + Each event entity for an event leading up to the original duplication of record **Rec.3.1** has two Duplicate Identifier (**M14.4.23**) elements so that the events for records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** all cross-reference each other
  + Each event entity for an event after the original duplication of record **Rec.3.1** leading up to the later duplication of record **Rec.3.1b** has one Duplicate Identifier (**M14.4.23**) elements so that only the events for records **Rec.3.1b(i)** and **Rec.3.1b(ii)** cross-reference each other during this period
  + The Participating Record Identifier (**M14.4.77**) refers to record **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)**, and
  + Where the event refers to a component, each duplicate record’s events will refer to its own components in the Participating Component Identifier (**M14.4.66**)
* On inspection of their components, all three records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** must have exactly the same components each with exactly the same metadata elements and the same values, except:
  + The System Identifier (**M14.4.100**) for each duplicate component is different,
  + The Record Identifier (**M14.4.86**) refers to one of record **Rec.3.1a**, **Rec.3.1b(i)** or **Rec.3.1b(ii)**, and
  + Each component has two Duplicate Identifier (**M14.4.23**) elements so that the components for records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** all cross-reference each other

### 3.2.23.5 Notes to test centre

* When record **Rec.3.1** is initially duplicated, each duplicate record should be indistinguishable from the other in all respects including metadata (apart from the System Identifier element), components and events
* Following duplication, each record is managed separately and independently
* When record **Rec.3.1b** is further duplicated then there will be three duplicate records in the test system, each of which should cross-reference the other two records
* The test system must therefore update record **Rec.3.1a**, its components and events (by adding appropriate Duplicate Identifier elements to these entities), when record **Rec.3.1b** is duplicated even though **Rec.3.1a** is not directly involved in the further duplication of record **Rec.3.1b**
* It is important to note that although the three records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)** are all duplicates of each other, the two records **Rec.3.1b(i)** and **Rec.3.1b(ii)** have a longer shared event history than either of these records has with record **Rec.3.1a**
* Therefore, events prior to the duplication of the original record **Rec.3.1** will be shared by all three records **Rec.3.1a**, **Rec.3.1b(i)** and **Rec.3.1b(ii)**
* Events after the duplication of record **Rec.3.1** but prior to the duplication of record **Rec.3.1b** will be shared only by the two records **Rec.3.1b(i)** and **Rec.3.1b(ii)**
* Events after the duplication of record **Rec.3.1b** will only occur in the event history of a particular record (all records are managed separately and independently from then on)

### 3.2.23.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.24 – Test Case

### 3.2.24.1 Test case description

**Export aggregation and record data**

### 3.2.24.2 Test case preconditions

* **T3.2.23** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.23** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.24.3 Test case steps (instructions)

* Export record **Rec.3.1** under **R11.4.1**
* Export root aggregation **Agg.3.1** under **R11.4.1** (including residual entities under **R11.4.2**)
* Export child aggregation **Agg.3.3** under **R11.4.1** (excluding residual entities under **R11.4.2**)
* Export the record service under **R11.4.1** (excluding residual entities under **R11.4.2**)
* The text comment for each export under **R11.4.5** should be: “Test case T3.2.24: export of *<name>*”

### 3.2.24.4 Test case post-conditions (expected result)

* Each of the four sets of exported data has its own universally unique export identifier, under **R11.4.4**
* Each of the four sets of exported data has the appropriate text comment, under **R11.4.5**
* The exported data in each set matches that expected (\* see notes) under **R11.4.3**, **R11.4.6**, **R11.4.7** and **R11.4.9**
* The export identifier and export comment can be found in the export event generated under **R11.4.10**, and stored in the event history of each entity exported
* The exported XML data is in Unicode, under **R2.4.28**
* All textual metadata elements in the exported XML data are accompanied by a language identifier, under **R2.4.28**
* All timestamps in the exported XML data are in the W3C XML dateTimeStamp format and include time zone information, under **R2.4.27**

### 3.2.24.5 Notes to test centre

* (\* see above) Definitive comparative test data is not yet available, as the test framework is in beta, and export data must currently be checked by hand

### 3.2.24.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.25 – Test Case

### 3.2.25.1 Test case description

**Final compliance check**

### 3.2.25.2 Test case preconditions

* **T3.2.24** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.24** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.25.3 Test case steps (instructions)

* Generate the compliance report for the test system (reference **R2.4.5**)

### 3.2.25.4 Test case post-conditions (expected result)

* The compliance report contains all of the information required by **R2.4.5**
* Specifically, the compliance report must show compliance with, at least:
  + **User and Group Service v1.x** (**cd532472-85b0-4c1c-82b4-5c8370b7d0e6**)
  + **Model Role Service v1.x (2f6d05c6-51e6-4a32-a7fc-c0a6883eb85b)**

**OR Native Role Service v1.x (d945dcd9-dc2d-491d-965a-11ce936d044b)**

* + **Classification Service v1.x** (**10fea10e-9c2f-4760-9095-f4f9295f4b19**)
  + **Record Service v1.x** (**ced3d0df-3f9f-4807-9e96-b5b790adad4a**)
  + **Model Metadata Service v1.x** (**a600f8d0-2d58-418e-bb41-211d1fd42350**)

**OR Native Metadata Service v1.x** (**66bf4419-d92f-4358-8506-7ee9c06abdcd**)

* + **Disposal Scheduling Service (fd05e284-181f-4f5d-bd8c-4bed835c8931)**
  + **Searching and Reporting v1.x** (**f09984a5-dd31-44d8-9607-22521667c78a**)
  + **Export Service v1.x** (**2777ab81-057e-4aa4-9595-69459ec2dc1e**)
  + **Graphical User Interface v1.x (0f9584e5-552a-4a79-a8ea-3c2801765255)**

**AND/OR Application Programming Interface (654633ec-8b17-4a3c-a483-436ee2bd506a)**

* The compliance report must use the UUIDs listed above, under **R2.4.23**

### 3.2.25.5 Notes to test centre

The compliance report should be identical to the compliance report generated by test case **T3.2.2**.

### 3.2.25.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

## T3.2.26 – Test Case

### 3.2.26.1 Test case description

**Final error check**

### 3.2.26.2 Test case preconditions

* **T3.2.25** has previously been completed successfully during the same test session
* The test system has not been reconfigured or restarted since **T3.2.25** was completed
* The test operator is logged in as user **Usr.3.1**

### 3.2.26.3 Test case steps (instructions)

* Log out of the test system
* Take a copy of the external error log, under **R2.4.7**

### 3.2.26.4 Test case post-conditions (expected result)

* Errors in the external error log must contain at least the information listed under **R2.4.7**

### 3.2.26.5 Notes to test centre

* Check which errors have occurred in the external error log since the start of the test module (compare against the external error log copy obtained under **T3.2.1**)
* Check whether there are have been any errors in this period generated by functions failing to perform under **R2.4.7**
* For any each error that has occurred the current user at the time the error occurred must have had access to the extended error information under **R2.4.8**
* Note that **R2.4.8** is not tested directly unless an error occurs during testing

### 3.2.26.6 Results

Test system operator for this test case

Result

🞏 Correct – result as expected

🞏 Incorrect – unexpected result or outcome

🞏 Incorrect – operation could not be initiated

🞏 Incorrect – crash, timeout or failed to finish for any reason

Observations

One or more of the test case steps…

🞏 …required more than one user action to complete (remark below)

🞏 …provided limited or no feedback to the user (remark below)

🞏 …was dependent on one or more previous user actions (remark below)

Additional remarks on observations

Macintosh HD:Users:jon:Documents:DLM Forum:DLM Foundation:Branding:Logos:MoReq logos:MoReq2010:MoReq2010_Reg:MoReq2010_Reg_pms.eps

# APPENDIX TO TEST MODULE 3

# 3.A Test data

The test system must be pre-loaded with at least the test data given here.

## 3.A.1 User and group service

### 3.A.1.1 User entities

| Title | Groups | Roles | Comment |
| --- | --- | --- | --- |
| **Usr.3.1** | * Grp.3.1 | **For Usr.3.1:**   * Rol.3.1 | Power User, member of group with read/write to record service and classification service |

### 3.A.1.2 Group entities

| Title | Roles | Comment |
| --- | --- | --- |
| **Grp.3.1** | **For User and Group Service:**   * Rol.3.3 * Rol.3.4   **For Role Service:**   * Rol.3.3 * Rol.3.5   **For Classification Service:**   * Rol.3.3 * Rol.3.6   **For Record Service:**   * Rol.3.2 * Rol.3.7 * Rol.3.8   **For Metadata Service:**   * Rol.3.3 * Rol.3.9   **For Disposal Scheduling Service:**   * Rol.3.3 * Rol.3.10   **For Disposal Holding Service:**   * Rol.3.3 * Rol.3.10 | Has read/write to record service and classification service, and read to other services (inherited by all group members) |

## 3.A.2 Role service

### 3.A.1.2 Role entities

| Title | Administrative? | Functions (access control) | Comment |
| --- | --- | --- | --- |
| **Rol.3.1** | No | * User – Browse Records Due For Disposal (**F14.5.178**) * User – Detailed Report (**F14.5.184**) * User – Export (**F14.5.185**) * User – Search (**F14.5.195**) * User – Summary Report (**F14.5.196**) | Power user |
| **Rol.3.2** | Yes | * Entity Type – Inspect (**F14.5.83**) * Entity Type – Inspect ACL (**F14.5.84**) * Entity Type – Inspect Event (**F14.5.85**) * Entity Type – Modify ACL (**F14.5.86**) * Function Definition – Inspect (**F14.5.87**) * Function Definition – Inspect ACL (**F14.5.88**) * Function Definition – Inspect Event (**F14.5.89**) * Function Definition – Modify ACL (**F14.5.91**) * Function Definition – Modify Event Generation (**F14.5.92**) * Function Definition – Modify Retain Event On Destruction (**F14.5.93**) * Service – Inspect (**F14.5.158**) * Service – Inspect ACL (**F14.5.159**) * Service – Inspect Event (**F14.5.160**) * Service – Modify ACL (**F14.5.161**) * Service – Modify Metadata (**F14.5.162**) * Service – Report Compliance (**F14.5.163**) | Read/write services |
| **Rol.3.3** | Yes | * Entity Type – Inspect (**F14.5.83**) * Entity Type – Inspect ACL (**F14.5.84**) * Entity Type – Inspect Event (**F14.5.85**) * Function Definition – Inspect (**F14.5.87**) * Function Definition – Inspect ACL (**F14.5.88**) * Function Definition – Inspect Event (**F14.5.89**) * Service – Inspect (**F14.5.158**) * Service – Inspect ACL (**F14.5.159**) * Service – Inspect Event (**F14.5.160**) * Service – Report Compliance (**F14.5.163**) | Read only services |
| **Rol.3.4** | No | * Group – Inspect (**F14.5.101**) * Group – Inspect ACL (**F14.5.102**) * Group – Inspect Event (**F14.5.103**) * Group – Report User Membership (**F14.5.108**) * User – Inspect (**F14.5.187**) * User – Inspect ACL (**F14.5.188**) * User – Inspect Event (**F14.5.189**) * User – Report Authorisation (**F14.5.193**) * User – Report Group Membership (**F14.5.194**) | Read only users and groups |
| **Rol.3.5** | No | * Role – Inspect (**F14.5.149**) * Role – Inspect ACL (**F14.5.150**) * Role – Inspect Event (**F14.5.151**) * Role – Report Function Definitions (**F14.5.156**) | Read only roles |
| **Rol.3.6** | Yes | * Class – Add Contextual Metadata (**F14.5.23**) * Class – Create (**F14.5.24**) * Class – Delete (**F14.5.25**) * Class – Delete Residual Event (**F14.5.26**) * Class – Delete Residual Metadata (**F14.5.27**) * Class – Destroy (**F14.5.28**) * Class – Inspect (**F14.5.30**) * Class – Inspect ACL (**F14.5.31**) * Class – Inspect Event (**F14.5.32**) * Class – Modify ACL (**F14.5.33**) * Class – Modify Default Disposal Schedule (**F14.5.34**) * Class – Modify Metadata (**F14.5.35**) * Class – Modify Originated Date/Time (**F14.5.36**) | Read/write classes |
| **Rol.3.7** | Yes | * Aggregation – Add Aggregation (**F14.5.1**) * Aggregation – Add Contextual Metadata (**F14.5.2**) * Aggregation – Add Record (**F14.5.3**) * Aggregation – Close (**F14.5.4**) * Aggregation – Create (**F14.5.5**) * Aggregation – Delete (**F14.5.6**) * Aggregation – Delete Residual Event (**F14.5.7**) * Aggregation – Delete Residual Metadata (**F14.5.8**) * Aggregation – Inherit Default Class (**F14.5.11**) * Aggregation – Inspect (**F14.5.12**) * Aggregation – Inspect ACL (**F14.5.13**) * Aggregation – Inspect Event (**F14.5.14**) * Aggregation – Modify ACL (**F14.5.15**) * Aggregation – Modify Max Levels Of Aggregation (**F14.5.16**) * Aggregation – Modify Metadata (**F14.5.17**) * Aggregation – Modify Originated Date/Time (**F14.5.18**) * Aggregation – Open (**F14.5.19**) * Aggregation – Override Class (**F14.5.20**) * Aggregation – Remove Aggregation (**F14.5.21**) * Aggregation – Remove Record (**F14.5.22**) | Read/write aggregations |
| **Rol.3.8** | No | * Record – Add Contextual Metadata (**F14.5.115**) * Record – Cancel Destruction (**F14.5.116**) * Record – Cancel Transfer (**F14.5.117**) * Record – Complete Review (**F14.5.118**) * Record – Confirm Destruction (**F14.5.119**) * Record – Confirm Transfer (**F14.5.120**) * Record – Create (**F14.5.121**) * Record – Delete Residual Event (**F14.5.122**) * Record – Delete Residual Metadata (**F14.5.123**) * Record – Disposal Alert (**F14.5.125**) * Record – Duplicate (**F14.5.126**) * Record – Held (**F14.5.128**) * Record – Inherit Default Class (**F14.5.129**) * Record – Inherit Default Disposal Schedule (**F14.5.130**) * Record – Inspect (**F14.5.131**) * Record – Inspect ACL (**F14.5.132**) * Record – Inspect Event (**F14.5.133**) * Record – Modify ACL (**F14.5.134**) * Record – Modify Metadata (**F14.5.135**) * Record – Modify Originated Date/Time (**F14.5.136**) * Record – Override Class (**F14.5.137**) * Record – Override Disposal Schedule (**F14.5.138**) * Record – Released (**F14.5.139**) * Record – Update Disposal (**F14.5.140**) | Read/write records |
| **Rol.3.9** | No | * Metadata Element Definition – Inspect (**F14.5.109**) * Metadata Element Definition – Inspect ACL (**F14.5.110**) * Metadata Element Definition – Inspect Event (**F14.5.111**) * Template – Inspect (**F14.5.171**) * Template – Inspect ACL (**F14.5.172**) * Template – Inspect Event (**F14.5.173**) | Read only metadata |
| **Rol.3.10** | Yes | * Disposal Hold – Inspect (**F14.5.63**) * Disposal Hold – Inspect ACL (**F14.5.64**) * Disposal Hold – Inspect Event (**F14.5.65**) * Disposal Schedule – Inspect (**F14.5.77**) * Disposal Schedule – Inspect ACL (**F14.5.78**) * Disposal Schedule – Inspect Event (**F14.5.79**) | Read only disposal |

## 3.A.3 Disposal scheduling service

### 3.A.3.1 Disposal Schedule entities

| Title | Disposal Action Code | Retention Trigger | Retention Period | Comment |
| --- | --- | --- | --- | --- |
| **DiS.3.1** | Retain Permanently | *N/A* | *N/A* | Retain permanently |
| **DiS.3.2** | Destroy | From aggregation closed date | 7 years | Destroy 7 years after aggregation closed |
| **DiS.3.3** | Transfer | From aggregation closed date | 2 years | Transfer 2 years after aggregation closed |
| **DiS.3.4** | Review | From aggregation closed date | 5 years | Review 5 years after aggregation closed |

## 3.A.4 Classification service

### 3.A.4.1 Class entities

| Title | Active or residual? | Disposal Schedule | Comment |
| --- | --- | --- | --- |
| **Cls.3.0** | Residual | DiS.3.1 | This class must have been created and destroyed before the start of testing |
| **Cls.3.1** | Active | DiS.3.1 | Default disposal schedule: Retain Permanently |
| **Cls.3.2** | Active | DiS.3.2 | Default disposal schedule: Destroy 7 years after aggregation closed |
| **Cls.3.3** | Active | DiS.3.3 | Default disposal schedule: Transfer 2 years after aggregation closed |