Test Module7

Referencing

Test Cases

Author(s): *imbus AG*
MoReq2 test development team
Date: *15/04/2008*
Version: *1.0*
Status: *Approved*
Customer: *Serco Consulting*

Contents

Document History

Version	Date of Issue	Author	Comment
0.1	16/05/07	Michael Sill	Initial Draft
0.2	31/08/07	Michael Sill	Revision after internal Review
0.3	02/11/07	Esther Kausz	Revision after receiving draft 3 of MoReq2
0.4	25/02/08	Michael Sill	Revision after final draft of MoReq2
1.0	07/04/2008	MoReq2 test development team	Finalising after approval of the test framework

T7.1 Classification Code

REMARKS	TEST DATA:	The test data for the following tests (test module 7, chapter T7.1) can be found in the corresponding test data repository. The reference chapter is called 'on **chapter T7.1 – Classification Code.**
	TEST EXECUTION:	step (1) Please read the chapter **Testframework – Introduction/Chapter 2** before starting the test execution. step (2) The test cases are designed in a way that enables you to skip certain test cases when required. However, the outcome of some test cases might be a precondition of subsequent tests. Where there are dependencies we have inserted test references into the precondition section of the test case. Careful attention should be given to the preconditions of the test cases.

T7.1.1 Configuration Time

Abstract: This chapter deals with general configurations of aspects of the classification code and the fully-qualified classification code.

T7.1.1.1 Format of a (Fully-Qualified) Classification Code

I. Global test case information			
test case id:	T7.1.1.1		
test case priority:	☐ Mandatory	☒ Optional	☐ Not Testable
test case description:	An administrative role checks the options to define the format of a (fully-qualified) Classification Code at configuration time. The administrative role configures a format for three levels of hierarchy. A check is made to confirm that the format of the classification code can be configured.		
Req.-ID:	7.1.5		

II. Test case

a. precondition(s)

- Logged in as user of the role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Check the options to format the (fully-qualified) Classification Code for the first level of the hierarchy	The **CentralAdministrator** is able to configure the following options: • Numeric, alphabetic or alphanumeric; • Presence or absence of leading zeroes; • minimum length (in the case of leading zeroes); • starting value; • increment
2.	Configure the classification code for the first level of the hierarchy as follows: • Numeric • presence of leading zeroes: minimum length is "3" • starting value is "1" • increment is "1"	The format of the classification code of the first hierarchy is configured (001, 002, 003, …)
3.	Check the options to format the (fully-qualified) Classification Code for a second level of the hierarchy	The **CentralAdministrator** is able to configure the following options: • Numeric, alphabetic or alphanumeric; • Presence or absence of leading zeroes; • minimum length (in the case of leading zeroes); • starting value; • increment
4.	Configure the classification code for the second level of the hierarchy as follows: • Numeric • presence of leading zeroes: minimum length is "3" • starting value is "1" • increment is "1"	The format of the classification code of the second level of the hierarchy is configured (001, 002, 003, …)
5.	Check the options to format the (fully-qualified) Classification Code for a third level of the hierarchy	The **CentralAdministrator** is able to configure the following options: • Numeric, alphabetic or alphanumeric; • Presence or absence of leading zeroes;

		• minimum length (in the case of leading zeroes); • starting value; • increment
6.	Configure the classification code for the third level of the hierarchy as follows: • Numeric • presence of leading zeroes: minimum length is "3" • starting value is "1" • increment is "1"	The format of the classification code of the third level of the hierarchy is configured (001, 002, 003, …)

c. postcondition(s)

- The ERMS allows an administrative role to configure the format of the (fully-qualified) Classification Code for three levels of hierarchy at configuration time.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.1.2 Separator Characters of a Fully-Qualified Classification Code

I. Global test case information			
test case id:	T7.1.1.2		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	A check on the documentation is made to confirm that the fully-qualified classification code consists of a concatenation of Classification Codes separated by a separator character.		
Req.-ID:	7.1.6		

II. Test case

a. precondition(s)

- User documentation is available.

b. test steps

step	action/operation	check/ expected result
1.	Read the ERMS's documentation carefully with regard to the consistence of the fully-qualified classification code.	The documentation contains information about the consistence of the fully-qualified classification code.

c. postcondition(s)

- According to the documentation the fully-qualified classification code consists of a concatenation of Classification Codes separated by a separator character.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.1.3 Separator Characters

I. Global test case information			
test case id:	T7.1.1.3		
test case priority:	☐ Mandatory	☒ Optional	☐ Not Testable
test case description:	An administrative role configures a separator character for a fully qualified classification code. A check is made to confirm that a separator character for the fully qualified classification code can be selected at configuration time.		
Req.-ID:	7.1.7		

II. Test case

a. precondition(s)

- Logged in as user of the role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Check the option to configure separator characters for a (fully-qualified) Classification Code	The **CentralAdministrator** is able to configure one of the following separator characters, at a minimum: • " " (space); • "-" (dash); • "/" (forward slash); • "." (dot).
2.	Configure "/" (forward slash) as separator character	The forward slash is configured as separator character (e.g. 001/001/002)

c. postcondition(s)

- The ERMS allows an administrative role to configure a separator character for the fully-qualified Classification Code at configuration time.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.2 Association/ Uniqueness/ Storage

Abstract: This chapter deals with the assigning of the classification code to entities, the aspect of the uniqueness and the storing of system identifiers as metadata elements.

T7.1.2.1 Association of a Classification Code With Newly Created Objects

I. Global test case information			
test case id:	T7.1.2.1		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	An administrative role creates a new class, file, sub-file, volume and a record consisting of components within the classification scheme. A check is made to confirm that the ERMS associates to each object a classification code automatically. Please note: The configuration of the classification code is optional. The classification code in this test case is configured as follows: • Numeric • presence of leading zeroes: minimum length is "3" • starting value is "1" • increment is "1" You will find a fully-qualified classification code such as CS01/001/001/001 in the test data repository. However, other formats may be possible too. Further, MoReq2 requests assigning a classification code to components. The assigning of the classification depends highly on the ERMS. To avoid doubts we do not state a value for the classification code of the components within the test data repository.		
Req.-ID:	7.1.1		

II. Test case

a. precondition(s)

- **TestClassificationScheme1** is created
- **Document1(RD01)** consists of two components (see test data repository)
- Logged in as user of the role **CentralAdministrator** .

b. test steps

step	action/operation	check/ expected result
1.	Create a new class **Product policies**	The class **Product policies (CS01/001)** is created; the classification code **001** is assigned to it.
2.	Create a new class **Innovation Policy** in the class **Product policies (CS01/001)**	The class **Innovation Policy (CS01/001/001)** is created; the classification code **(001)** is assigned to it.
3.	Create a new file **Study1** in the class **Innovation Policy (CS01/001/001)**	The file **Study1 (CS01/001/001/001)** is created; the classification code **(001)** is assigned to it.
4.	Create a new sub-file **Policies** in the file **Study1 (CS01/001/001/001)**	The sub-file **Policies (CS01/001/001/001/001)** is created; the classification code **(001)** is assigned to it.
5.	Create a new volume **P1** in the sub-file **Policies (CS01/001/001/001/001)**	The volume **P1 (CS01/001/001/001/001/001)** is created; the classification code **(001)** is assigned to it.
6.	Capture **Document1 (RD01)** as record **Policy1** in volume **P1 (CS01/001/001/001/001/001)**	The record **Policy1 (CS01/001/001/001/001/001/001)** is created; a classification code **(001)** is assigned to the record and its components.

c. postcondition(s)

• The ERMS associates to a new class, file, sub-file, volume, record and its components a classification code.

III. Test result	
defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.2.2 Uniqueness of a Fully Qualified Classification Code

I. Global test case information			
test case id:	T7.1.2.2		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	An administrative role checks the documentation to ensure that the fully-qualified classification code used by the ERMS is unique within a classification scheme hierarchy.		
Req.-ID:	7.1.2		

II. Test case

a. precondition(s)

- User documentation is available.

b. test steps

step	action/operation	check/ expected result
1.	Read the ERMS's documentation carefully with regard to the uniqueness of the fully-qualified classification code.	The documentation contains information about the uniqueness of the fully-qualified classification code.

c. postcondition(s)

- Information about the uniqueness of the fully-qualified classification code is found in the ERMS's documentation.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.2.3 Retaining Uniqueness of Fully-Qualified Classification Codes after Relocation

I. Global test case information			
test case id:	T7.1.2.3		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	An administrative role relocates a record, volume, sub-file, file and class to another class. A check is made to confirm that the ERMS assigns a new classification code and a fully –qualified classification code to all entities after their relocation.		
Req.-ID:	7.1.3		

II. Test case

a. precondition(s)

- **TestClassificationScheme1** is created
- The record **Comment1 (CS01/001/002/001/001/001/001)** consists of two components.
- Logged in as user of role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Relocate the class **Information Technology (CS01/001/002)** below of class **Quality (CS01/002)**	The class **Information Technology** is relocated. The class itself, the file, sub-file, volume, record and its component below the class received a new classification code.

c. postcondition(s)

- When relocating a record, volume, sub-file, file and class to another class, the ERMS ensures that all entities are assigned a new classification code and a fully –qualified classification code after their relocation.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.2.4 Storage of Classification Codes as Metadata Elements

I. Global test case information			
test case id:	T7.1.2.4		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	A check is made to confirm that the ERMS assigns to each record and its aggregations a metadata with its classification code. Please note: MoReq2 requests assigning a classification code to components. The assigning of the classification depends highly on the ERMS. To avoid doubts we do not state a value for the classification code of components within the test data repository.		
Req.-ID:	7.1.4		

II. Test case

a. precondition(s)

- **TestClassificationScheme1** is created.
- The class **Product policies (CS01/001)** and all its entities are created (see test case T7.1.2.1)
- Logged in as user of the role **CentralAdministrator**

b. test steps

step	Action/operation	check/ expected result
1.	Check the metadata of the class **Product policies (CS01/001)**	The ERMS stored the classification code to each entity of the class **Product policies (CS01/001)** as metadata element

c. postcondition(s)

- The ERMS is able to store the classification code of a class, file, sub-file, volume, record and its components as metadata element.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.3 Automatic / Manual Generation

Abstract: The test focus of the following chapter lies on the generation of classification codes by the given options of the ERMS.

T7.1.3.1 Generation of the Classification Code

I. Global test case information			
test case id:	T7.1.3.1		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	A check is made to confirm that the ERMS allows an administrative role to configure how the classification code for descendant entities of a new created class is created: automatically by the ERMS or by an authorised user or an external application.		
Req.-ID:	7.1.8		

II. Test case

a. precondition(s)

- Logged in as user of the role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Check the options to configure the generation of the classification code for descendant entities of a newly created class	The ERMS allows the following options to generate a classification code for the entities of a newly created class: • automatic generation by the ERMS • generation by an authorised user or an external application

c. postcondition(s)

- The ERMS allows an administrative role to configure the generation of the classification code of descendant entities of a newly created class either automatically by the ERMS or by a user or an external application.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.3.2 Automatic Generation of the Classification Code by the ERMS

I. Global test case information			
test case id:	T7.1.3.2		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	An administrative role creates a class and a file in an existing class. A check is made to confirm that the classification code is automatically assigned by the ERMS.		
Req.-ID:	7.1.9		

II. Test case

a. precondition(s)

- **TestClassificationScheme2** is created
- The top level class **Executive Committee (CS02/001)** is configured to assign the classification code of descendant entities automatically (see also test case T7.1.3.1)
- Logged in as user of role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Create the class **Resourcing** in the class **Executive Committee (CS02/001).**	The class **Resourcing (CS02/001/002)** is created. The ERMS assigned the next sequential number 002 as classification code to the class.
2.	Create the file **Result** in the class **Resourcing (CS02/001/002)**	The file **Result (CS02/001/002/001)** is created. The ERMS assigned 001 (start value) as classification code to the file.

c. postcondition(s)

- The ERMS automatically assigned the next available number as classification code to newly generated entities.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.1.3.3 Manual Generation of a Classification Code by a User

I. Global test case information			
test case id:	T7.1.3.3		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	An administrative role creates a file in an existing class. The ERMS validates the manually entered classification codes within the class.		
Req.-ID:	7.1.10		

II. Test case

a. precondition(s)

- **TestClassificationScheme2** is created
- The top level class **Corporate Communication (CS02/002)** is configured to assign the classification code of descendant entities manually by the user (see also test case T7.1.3.1)
- Logged in as user of role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Create the class **Business Segments** in the class **Corporate Communication (CS02/002).** Enter manually the classification code **002**.	The class **Business Segments (CS02/002/002)** is created; the classification code **002** is assigned to it.
2.	Attempt to create the class **Human Resources.** Enter manually the classification code **002**.	The ERMS denies the assignment of **002** as classification code to the class; **002** already exists.

c. postcondition(s)

- The ERMS accepts the manually entered Classification Code and validates it for uniqueness within its parents.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.2System Identifiers

REMARKS	TEST DATA:	The test data for the following tests (test module 7, chapter T7.2) can be found in the corresponding test data repository. The reference chapter is called 'on **chapter T7.2 – System Identifiers.**
	TEST EXECUTION:	step (1) Please read the chapter **Testframework – Introduction/Chapter 2** before starting the test execution. step (2) The test cases are designed in a way that enables you to skip certain test cases when required. However, the outcome of some test cases might be a precondition of subsequent tests. Where there are dependencies we have inserted test references into the precondition section of the test case. Careful attention should be given to the preconditions of the test cases.

T7.2.1 Association/Uniqueness/ Storage

Abstract: This chapter deals with the assigning of system identifiers to entities, the aspect of uniqueness and the storing of system identifiers as metadata elements.

T7.2.1.1 Association of a Newly Generated Object With a System Identifier

I. Global test case information			
test case id:	T7.2.1.1		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	An administrative role creates a new classification scheme, class, file, sub-file, volume, record, record extract, retention and disposition schedule. A check is made to confirm that the ERMS associates each of these objects to a system identifier.		
Req.-ID:	7.2.1		

II. Test case

a. precondition(s)

- Logged in as user of role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Create a new classification scheme **TestclassificationScheme1**	The classification scheme **TestclassificationScheme1** is created; a new system identifier is assigned to it.
2.	Create a new class **Marketing**	The class **Marketing (CS01/001)** is created; a new system identifier is assigned to it.
3.	Create a new file **Strategic business units** within the class **Marketing (CS01/001)**	The file **Strategic business units (CS01/001/001)** is created; a new system identifier is assigned to it.
4.	Create a new sub-file **Competitor A** within the file **Strategic business units (CS01/001/001)**	The sub-file **Competitor A (CS01/001/001/001)** is created; a new system identifier is assigned to it.
5.	Create a new volume **SWOT20** within the sub-file **Competitor A (CS01/001/001/001)**	The volume **SWOT20 (CS01/001/001/001/001)** is created; a new system identifier is assigned to it.
6.	Capture a new record **Strategic planning** within **the** volume **SWOT20 (CS01/001/001/001/001)**	The record **Strategic planning (CS01/001/001/001/001/001)** is captured; a new system identifier is assigned to it.
7.	Capture a new record extract **Performance** within **the** volume **SWOT20 (CS01/001/001/001/001)**	The record extract **Performance (CS01/001/001/001/001/002)** is created; a new system identifier is assigned to it.
8.	Create a new retention and disposition schedule **Test**	The retention and disposition schedule **Test (RDS01)** is created; a new system identifier is assigned to it.

c. postcondition(s)

- The ERMS associates to a new classification scheme, class, file, sub-file, volume, record, record extract, retention and disposition schedule and document a system identifier.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed

Remarks	*tester*
	_____ date, signature

T7.2.1.2 Storage of a System Identifier as a Metadata Element

I. Global test case information			
test case id:	T7.2.1.2		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	A classification scheme, class, file, sub-file, volume, record, record extract and retention and disposition schedule are created. A check is made to confirm that the ERMS is able to assign the System Identifier as a metadata element to each entity.		
Req.-ID:	7.2.3		

II. Test case

a. precondition(s)

- **TestClassificationScheme1** and all its entities are created (see test case T7.2.1.1).
- The retention and disposition schedule **Test (RDS01)** is created (see test case T7.2.1.1).
- Logged in as user of role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Check the metadata on **TestClassificationScheme1** and all its entities respectively to the system identifier	The ERMS stored the system identifier of each entity as a metadata element.
2.	Check the metadata on the retention and disposition schedule **Test (RDS01)** respectively to the system identifier	The ERMS stored the system identifier of the retention and disposition schedule **Test (RDS01)** as a metadata element.

c. postcondition(s)

- The ERMS is able to store the System Identifier as metadata element of a classification scheme, class, file, sub-file, volume, record, record extract and retention and disposition schedule.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.2.1.3 The UUID Algorithm

I. Global test case information			
test case id:	T7.2.1.3		
test case priority:	☐ Mandatory	☒ Optional	☐ Not Testable
test case description:	An administrative role checks the documentation to confirm whether the ERMS uses the UUID algorithm (as specified in ISO/IEC 9834-9 and ITU-T Rec. X.667) to generate a globally unique System Identifier. Please note, that this test case strongly depends on your vendor's support. They should clearly demonstrate that the ERMS uses the UUID algorithm (as specified in ISO/IEC 9834-9 and ITU-T Rec. X.667) to generate a System Identifier.		
Req.-ID:	7.2.5		

II. Test case

a. precondition(s)

- Logged in as user of the role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Read the ERMS's documentation carefully with regard to the algorithm that generates the System identifier.	The documentation contains information about the algorithm that generates the System identifier.

c. postcondition(s)

- The ERMS uses and documents the UUID algorithm (as specified in ISO/IEC 9834-9 and ITU-T Rec. X.667) to generate globally unique System Identifiers.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.2.2 Others:

Abstract: This chapter summarises all other test cases.

T7.2.2.1 Entering or Usage of a System Identifier for ERMS Functions

I. Global test case information			
test case id:	T7.2.2.1		
test case priority:	☒ Mandatory	☐ Optional	☐ Not Testable
test case description:	An administrative role executes three ERMS functions (e.g. search, create a record, delete a file). A check is made to confirm that the administrative role does not have to enter or use a system identifier.		
Req.-ID:	7.2.6		

II. Test case

a. precondition(s)

- Logged in as user of role **CentralAdministrator**

b. test steps

step	action/operation	check/ expected result
1.	Execute three ERMS functions	The three functions are executed: The ERMS did not require the entering of the system identifier.

c. postcondition(s)

- Three ERMS functions were executed. There was no need to enter or use the System identifier.

III. Test result

defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.2.2.2 Uniqueness of System Identifiers

I. Global test case information			
test case id:	T7.2.2.2		
test case priority:	☐ Mandatory	☐ Optional	☒ Not Testable
test case description:	The corresponding requirement is not testable.		
Req.-ID:	7.2.2		

II. Test case		
a. precondition(s)		
•		
b. test steps		
step	action/operation	check/ expected result
1.		
c. postcondition(s)		
•		

III. Test result	
defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature

T7.2.2.3 Globally Uniqueness of System Identifiers

I. Global test case information			
test case id:	T7.2.2.3		
test case priority:	☐ Mandatory	☐ Optional	☒ Not Testable
test case description:	The corresponding requirement is not testable.		
Req.-ID:	7.2.4		

II. Test case		
a. precondition(s)		
•		
b. test steps		
step	action/operation	check/ expected result
1.		
c. postcondition(s)		
•		

III. Test result	
defects / deviations	*verdict*
	☐ passed ☐ failed
Remarks	*tester*
	_____ date, signature