

EbRim implementation with GeoNetwork and Omar

TOOLBOX SOFTWARE

Acceptance Test Plan

Authors:

M. Fanciulli

30/04/09

Reviewed by:

P. Nencioni

30/04/09

Approved by:

S. Gianfranceschi

30/04/09

ERG-ATP-4100-INT :**Document Id**
1-30/04/2009 : **Issue**
0-30/04/2009 : **Revision**



Document change record

<i>Issue</i>	<i>Issue date</i>	<i>Pages/section effected</i>	<i>Reason for change</i>
1.0	30/04/2009	All	Initial version

ERG-ATP-4100-INT : **Document Id**
1-30/04/2009 : **Issue**
0-30/04/2009 : **Revision**



Distribution List

<i>Company</i>	<i>Name</i>	<i>Function</i>	<i>N° of copies</i>
ESA	Pier Giorgio Marchetti	ESA Technical Officer	1
Intecs	Simone Gianfranceschi	Ergo Project Manager	1

Table of Content

<u>1. INTRODUCTION</u>	<u>7</u>
1.1. PURPOSE	7
1.2. GLOSSARY	7
1.2.1. ABBREVIATIONS	7
1.2.2. DEFINITION OF TERMS	9
1.3. REFERENCES	9
1.3.1. NORMATIVE REFERENCES	9
1.3.2. INFORMATIVE REFERENCES	9
<u>2. ORGANISATION OF TEST ACTIVITIES</u>	<u>10</u>
2.1. FACTORY ACCEPTANCE ACTIVITIES	10
2.1.1. ACTIVITY DEFINITION	10
2.1.2. DESCRIPTION OF A FACTORY ACCEPTANCE SESSION	10
2.1.3. VERIFICATION METHOD	10
2.1.4. ACTIVITY RESULTS	11
<u>3. TEST ENVIRONMENT</u>	<u>12</u>
3.1. HARDWARE & SOFTWARE CONFIGURATION	12
3.2. TEST TOOLS	13
<u>4. TEST SPECIFICATION</u>	<u>14</u>
4.1. TEST DESIGN	14
4.2. TEST SCENARIO TS_01 CATALOGUE SERVICE CREATION	14
4.2.1. TEST CASE TS_01_01 CATALOGUE SERVICE IMPLEMENTING THE STANDALONE APPROACH	14
4.2.2. TEST CASE TS_01_02 CATALOGUE SERVICE IMPLEMENTING THE GATEWAY APPROACH	15
4.2.3. TEST CASE TS_01_03 ADDING OPERATIONS TO A CATALOGUE SERVICE IMPLEMENTING THE STANDALONE APPROACH	17
4.2.4. TEST CASE TS_01_04 ADDING OPERATIONS TO A CATALOGUE SERVICE IMPLEMENTING THE GATEWAY APPROACH	18
4.3. TEST SCENARIO TS_02 HARVESTING DATA	19
4.3.1. TEST CASE TS_02_01 HARVESTING DATA AND RETRIEVING THEIR GML DOCUMENTS	19



4.4. TEST SCENARIO TS_03 SOAP INTERFACES VALIDATION	20
4.4.1. TEST CASE TS_03_01 MAXRECORDS HANDLING	20
4.4.2. TEST CASE TS_03_02 VALIDATION OF THE EO PRODUCT CATALOGUE SERVICE	21
4.4.3. TEST CASE TS_03_03 VALIDATION OF THE CIM CATALOGUE SERVICE.....	22
4.5. TEST SCENARIO TS_04 ACCESSING DOCUMENTATION	23
4.5.1. TEST CASE TS_04_01 ACCESSING DOCUMENTATION	23
<u>5. TRACEABILITY MATRICES.....</u>	<u>24</u>
5.1. SOFTWARE REQUIREMENTS VS TEST CASES.....	24
5.2. TEST CASES VS SOFTWARE REQUIREMENTS.....	26

1.INTRODUCTION

1.1.Purpose

This document is the Acceptance Test Plan (ATP) for the ERGO project software, and represents a formal deliverable of work package 4100.

The objectives of this plan are to:

- Define the validation approach;
- Describe the activities needed for the preparation and execution of testing;
- Define the testing environment;
- Define the sequence of the validation/acceptance tests;
- Serve as a guide in the definition of the test procedures.

1.2.Glossary

1.2.1.Abbreviations

<i>Acronym</i>	<i>Extended Form</i>
ATP	Acceptance Test Plan
ATS	Abstract Test Suite
ATR	Acceptance Test Report
EO	Earth Observation
FAT	Factory Acceptance Test
GMES	Global Monitoring for Environment and Security
HMA	Heterogeneous Mission Accessibility
HTTP	Hypertext Transfer Protocol
HW	Hardware
I/F	Interface
ICD	Interface Control Document
NA	Not Applicable

Acronym	Extended Form
OGC	Open Geospatial Consortium
SOAP	Simple Object Access Protocol
SR	Software Requirements
SRD	Software Requirements Document
SSD	Software Specification Document
SSE	Service Support Environment
SSL	Secure Sockets Layer
SUM	Software User Manual
SW	Software
TBC	To Be Confirmed
TBD	To Be Defined
TBW	To Be Written
TN	Technical Note
UML	Unified Modeling Language
URD	User Requirements Document
URL	Universal Resource Locator
WSDL	Web Services Description/Definition Language
XML	Extensible Mark-up Language
XSD	XML Schema Definition
XSLT	Extensible Style Language Transformation

1.2.2. Definition of Terms

1.3.References

1.3.1.Normative References

In case of conflict between two or more applicable documents, the higher document will prevail.

- [NR1] ERGO Technical Proposal, Id. ERGO-TEC-PROP-354-07-SP-PI, issue 1, Revision 0, 05/02/2008.
- [NR2] Toolbox Software Requirement Document, Id. ERG-SRD-2100-INT, issue 1 Revision 1, 06/06/2008.
- [NR3] ERGO Project Management Plan, Id. ERG-PMP-1000-INT, Issue 1, Revision 1, Date 30/04/2009
- [NR4] ERGO Project Assurance Plan, Id. ERGO-PAP-1000-INT, Issue 1, Revision 0, Date 05/05/2008

1.3.2.Informative references

The following documents, although not a part of this test procedure, amplify or clarify its contents.

- [IR1] SOAP Simple Object Access Protocol 1.1, W3C Note 08 May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [IR2] Hypertext Transfer Protocol -- HTTP/1.1, RFC 2616, U.C. Irvine, DEC W3C/MIT, DEC, W3C/MIT, W3C/MIT, January 1997, <http://www.normos.org/ietf/rfc/rfc2616.txt>
- [IR3] Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, <http://www.w3.org/TR/wsdl>
- [IR4] XML Schema, <http://www.w3.org/TR/xmlschema-0/>, W3C Recommendation, 2 May 2001.
- [IR5] Extensible Mark-up Language (XML) 1.0, W3C Recommendation 10 February 1998, <http://www.w3.org/TR/REC-xml>.
- [IR6] XSL Transformations (XSLT) Version 1.0, W3C Recommendation 16 November 1999, <http://www.w3.org/TR/xslt>.
- [IR7] Web Services Addressing (WS-Addressing) <http://www.w3.org/Submission/ws-addressing/>

2.ORGANISATION OF TEST ACTIVITIES

This document defines the acceptance test plan for the Toolbox software and its integration with the ebRR catalogue. The validation consists of Factory Acceptance (i.e. verification against software requirements) activities, as described in the following section.

2.1.Factory Acceptance Activities

2.1.1.Activity Definition

The objective of the Factory Acceptance Test is to check that ERGO software satisfies all the requirements listed in the Software Requirement Document [NR2].

The FAT tests are run at INTECS premises. The following tasks are part of the factory acceptance:

- Installation of ERGO software
- Execution of the formal acceptance tests
- Writing of the test execution report

2.1.2.Description of a Factory Acceptance Session

The Factory Acceptance Test session starts with the verification of the testing environment and the presence of all required people. During this "Test Readiness Review", it will be checked whether the software is ready to be submitted to the factory acceptance tests (e.g. open actions, open change requests, and state of documentation).

The next activity is the execution of the formal factory acceptance. If a difference is detected between the observed behavior of the software and the expected behavior described in the Test Procedures, then the test engineer raises a problem report. In case a blocking problem is encountered, the test case is skipped and testing continues with the next test case.

Finally at the end of the session, a Factory Acceptance Test Report is written. Depending on the number of major problems that are detected, a retest of some test cases may be appended to the testing activities, if a new release with a number of corrections is available.

2.1.3.Verification Method

The following verification methods are envisaged for test execution:

- **Analysis [A]** This verification method implies use of analytical techniques (such as system engineering analysis, statistics, mathematical modeling, simulations) and shall be used to verify such requirements.

- **Review of Design [D]** This verification method may be used when approved Design reports, technical descriptions, engineering drawings unambiguously show that the requirement is met.
- **Inspection [I]** Verification by inspection is only done when testing is insufficient or inappropriate. This method of verification is for those requirements that are normally performed by some form of visual inspection. This would include workmanship, labeling, envelope requirements etc.
- **Demonstration [M]** This verification method may be used when actual conduct can verify achievement of requirements such as service and access, transportability, human engineering features and processes hardware. A requirement which is of an operational or functional nature and is not quantified by a specific measurable parameter may be verified by demonstration.
- **Similarity [S]** This verification method may be used when there is proof that the item is similar or identical in design and manufacturing processes to another previously qualified to equivalent or more stringent criterion.
- **Test [T]** A requirement may be verified by test alone if the form of the specification is such that the requirement can be directly measured.

2.1.4. Activity Results

The outputs of the Acceptance Test phase is a signed Acceptance Test Report (ATR) which lists for each test scenario/test case whether it failed or was successful, and a general acceptance statement.

Three cases are possible:

- **Rejection:** the reasons are written in the report; the software is corrected according to the normal change control and configuration management procedures; a new factory acceptance test session is scheduled;
- **Conditional acceptance:** some problems have been found, but the factory acceptance is signed providing that they will be corrected.
- **Full acceptance:** the factory acceptance is signed.

3. Test Environment

In this chapter the software and hardware resources required to perform the Factory Acceptance Test sessions are listed.

The Ergo Test Environment is provided as a zip file whose content is structured in subdirectories, one for each test case. In order to install the Test Environment, its zip package shall be copied and it shall be unzipped.

Below an example of the possible structure:

```
<UNZIP_DIR>
  |-<TS_01>
    |- <TS_01_01>
    |- <TS_01_02>
  |-<TS_02>
    |- <TS_02_01>
  |-<TS_03>
    |- <TS_03_01>
    |- <TS_03_02>
    |- <TS_03_03>
```

3.1. Validation Test Cases

3.2. Validation tests are performed by executing CTL based test procedures. Some are directly taken from ATS defined in the HMA-T project, other have been defined in order to automatized some specific checks.

A CTL test suite is made up of one or several tests to be executed by the Team Engine tool. A CTL test is made of the following main steps:

- Define the type (XML or SOAP) of the request;
- Define the mandatory (URL of the Toolbox Catalogue Service) and optional parameters of the request;
- Include either directly the request or the link to the request, if this is provided in a separate file, as usually happens;
- Check that a response is returned by the Toolbox Catalogue Service;
- Inspect the response to test whether it is compliant with what expected.

Given that these steps are the same for all the test cases, they will not be repeated in the description of the test cases: they will be simply referred to, whereas details will be given about the specific request to be sent to the Toolbox Catalogue Service.

3.3. Hardware & Software Configuration

The test environment is made up of the HW/SW configurations for the TOOLBOX and TeamEngine systems.

The HW/SW configuration for the TOOLBOX is as follows:

Host	Hardware	Software
PC1	Dual CPU AMD Opteron 246 (detected 1992.536 MHz processor), 4 GB of RAM	Linux (Fedora Core 9) or Windows XP Apache Tomcat (version 5.5 or greater) Java Runtime Environment (version 1.5 or greater) Toolbox software Postgres (version 8.3 or greater) with PostGIS extension Firefox 3.0.x

The HW/SW configuration for the TeamEngine is as follows:

Host	Hardware	Software
PC2	I586 Intel processor + 2GB RAM	Windows XP Apache Tomcat (version 5.5 or greater) Java Runtime Environment (version 1.6.x) Firefox 3.0.x TeamEngine tool software

An internet connection is required to complete most of the test cases.

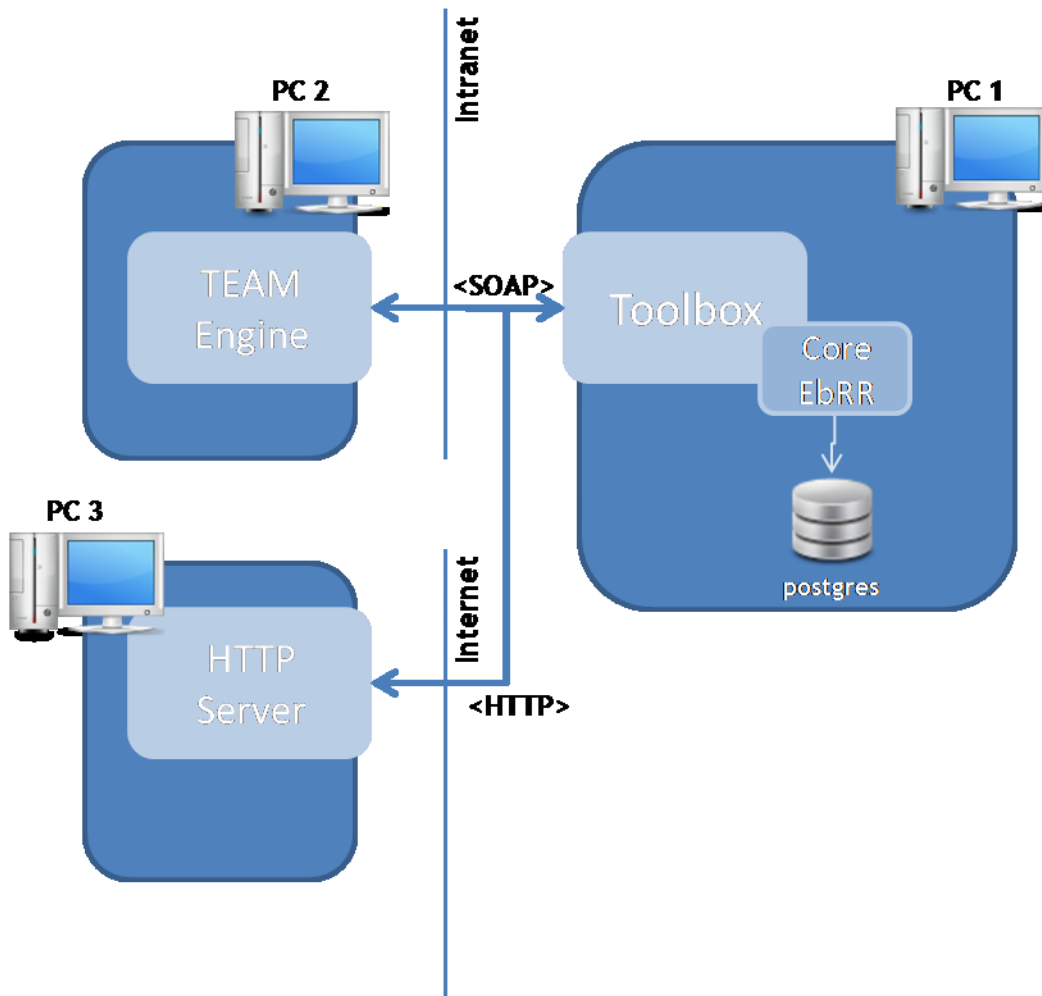


Figure 1 Test Environment

In Figure 1 is described the test environment that is going to be used. PC3 is a generic HTTP server which can host XML documents in order to perform the harvesting tests.

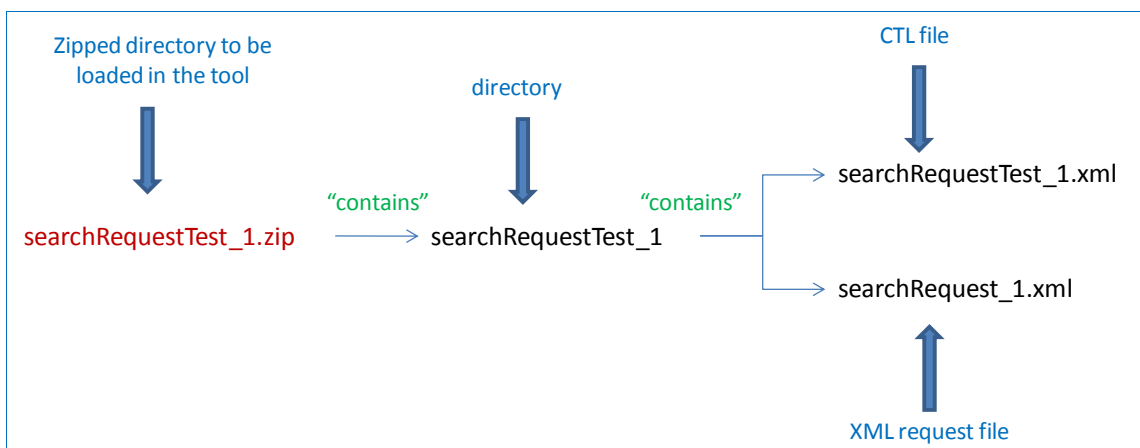
An external catalogue is also tested in the gateway mode so any requests received by Toolbox are processed and forwarded. In this case an internet connection to the external catalogue is mandatory.

3.4. Test tools

The Test tool used for the tests described in this document is the TeamEngine, able to execute tests written in the CTL language **Errore. L'origine riferimento non è stata trovata.** The TeamEngine is a WEB application running as a Tomcat service, able to load, execute and log CTL test suites. Each test suite is made up an arbitrary number of tests.

The TeamEngine is distributed as a “war” file to be deployed under Tomcat and it is accessible as http://tomcat_host:8080/TeamEngine; through the provided GUI, the user can:

- 1) Loads the CTL test suite, which shall be provided as a zipped directory file, as in the example described in the following figure. Notice that the name of the directory and that of the CTL file [without suffix] shall be the same; the directory shall contain all of the resources needed to execute the tests:
 - a. The CTL file is the main and mandatory file, which lists the steps to be executed to carry out the test suite; the tests are described using the CTL language
 - b. Other files are optional, but usually provided, such as files containing the actual XML requests to be sent to TOOLBOX.



- 2) Execute the CTL test suite by selecting the test suite and clicking on the “Execute” button.

Inspect the response displayed on the GUI, showing the pass/fail result.

4. Test Specification

4.1. Test Design

The testing is divided into **Test Scenarios** and **Test Cases**.

Test Scenarios are group of tests, which have a common theme or objective (e.g. Installation, communication mechanism used etc). Each Scenario shall be divided into a number of sub-cases. Each sub-case is a **Test Case**.

The test descriptions should be read as specifications for the corresponding test procedures. The tests are performed manually.

4.2. Test Scenario TS_01 Catalogue service creation

4.2.1. Test Case TS_01_01 Catalogue service implementing the StandAlone approach

Prerequisites	<ul style="list-style-type: none">• Toolbox RE 8.0 installed
Input	
Steps	<ol style="list-style-type: none">1. Login into the Toolbox Administration Web Pages2. Click on “Services Management”3. Click on “Create a new Service”4. Fill the service name field with “OGC06-131_standalone” and click on “Next”5. Select “Catalogue” from the combo box and click on “Next”6. Click on “Next”7. Verify that the following interfaces are listed:<ol style="list-style-type: none">a. OGC-06-131r4b. OGC-07-038c. OGC-05-025r38. Select “OGC-06-131r4” from the combo box and click on “Next”9. Verify that the following modes are available:<ol style="list-style-type: none">a. StandAloneb. Gateway10. Select “StandAlone” from the combo box and click on “Next”

	<ol style="list-style-type: none">11. Click on “Create”12. Verify that the following variables are listed in the “Configure service variables”:<ol style="list-style-type: none">a. maxRecordsb. ebRRExternalInstanceUrlc. ebRRDbNamed. ebRRDbUrle. ebRRDbPwdf. ebRRDbUser13. Modify value of variable maxRecords to 2514. Click on “Configure”
Results	The service has been successfully created
Test pass/failure criteria	
References	

4.2.2. Test Case TS_01_02 Catalogue service implementing the Gateway approach

Prerequisites	<ul style="list-style-type: none">• Toolbox RE 8.0 installed
Input	
Steps	<ol style="list-style-type: none">1. Login into the Toolbox Administration Web Pages2. Click on “Services Management”3. Click on “Create a new Service”4. Fill the service name field with “OGC07-038_gateway” and click on “Next”5. Select “Catalogue” from the combo box and click on “Next”6. Click on “Next”7. Verify that the following interfaces are listed:<ol style="list-style-type: none">a. OGC-06-131r4

	<ol style="list-style-type: none"> b. OGC-07-038 c. OGC-05-025r3 <ol style="list-style-type: none"> 8. Select “OGC-07-038” from the combo box and click on “Next” 9. Verify that the following modes are available: <ol style="list-style-type: none"> a. Gateway 10. Select “Gateway” from the combo box and click on “Next” 11. Click on “Create” 12. Verify that the following variables are listed in the “Configure service variables”: <ol style="list-style-type: none"> a. maxRecords 13. Modify value of variable maxRecords to 25 14. Click on “Configure”
Results	The service is created
Test pass/failure criteria	The service has been successfully created
References	

4.2.3. Test Case TS_01_03 Adding operations to a catalogue service implementing the StandAlone approach

Prerequisites	<ul style="list-style-type: none"> • TS_01_01 successfully executed
Input	
Steps	<ol style="list-style-type: none"> 1. Login into the Toolbox Administration Web Pages 2. Select “OGC06-131_standalone” service from the drop down in the main page 3. Click on “Operations management” 4. Click on “Add operation” 5. Verify that the following operations are available: <ol style="list-style-type: none"> a. GetCapabilities b. DescribeRecord

	<ol style="list-style-type: none">c. GetRecordsd. GetRecordBylde. Harvest <ol style="list-style-type: none">6. Select “GetCapabilities” from the combo box and click on “Next”7. Verify that is not requested to specify a Toolbox script.8. Click on “Create”9. Click on “Configure”10. Verify that the operation has been added11. Repeat steps from 4 to 10 until all operations have been added. Verify that each step the list in step 5 is filtered according with the operations already created
Results	SOAP operations are created
Test pass/failure criteria	All operations have been created without the need of specifying logic scripts
References	

4.2.4. Test Case TS_01_04 Adding operations to a catalogue service implementing the Gateway approach

Prerequisites	<ul style="list-style-type: none">• TS_01_02 successfully executed
Input	
Steps	<ol style="list-style-type: none">1. Login into the Toolbox Administration Web Pages2. Select “OGC07-038_gateway” service from the drop down in the main page3. Click on “Operations management”4. Click on “Add operation”5. Verify that the following operations are available:<ol style="list-style-type: none">a. GetCapabilitiesb. DescribeRecordc. GetRecords

	<ol style="list-style-type: none"> d. GetRecordByld e. Harvest 6. Select “GetCapabilities” from the combo box and click on “Next” 7. Verify that a Toolbox script shall be specified. 8. Click on “Browse” and select the file <code><Ergo_Test_Environment>/TS_01/TS_01_04/GetCapabilities/GetCapabilities.xml</code> 9. Click on “Create” 10. Click on “Configure” 11. Verify that the operation has been added 12. Repeat steps from 4 to 11 until all operations have been added. Use the following file in each step: <code><Ergo_Test_Environment>/TS_01/TS_01_04/<operation>/<operation>.xml</code> 13. Verify that each step the list in step 5 is filtered according with the operations already created
Results	SOAP operations are created
Test pass/failure criteria	All operations have been created. For each operation a logic script has been specified.
References	

4.3. Test Scenario TS_02 Harvesting data

4.3.1. Test Case TS_02_01 Harvesting data and retrieving their GML documents

Prerequisites	<ul style="list-style-type: none"> • TS_01_03 successfully executed
Input	TS_02_01.ctl file in TS_02/TS_02_01 directory
Steps	<ol style="list-style-type: none"> 1. Login into the Toolbox Administration Web Pages 2. Select “TOOLBOX Configuration” 3. Select “Change the configuration” 4. Update the field “Repository Home Directory” to an

accessible directory.

5. Click on “Configure”
6. Select the service “OGC-06-131_standalone” from the drop down.
7. Select “Test Center”
8. Select “Harvest”
9. Fill the HTTP Url with the following:
http://toolbox.esrin.esa.int/Download/Ergo/TS_02_01.xml
10. Click on “Harvest”
11. Verify that a success message is displayed.
12. Confirm it and click on “Monitoring Center”
13. Select “Display Synchronous Instances”
14. Check that a new instance is available for the Harvest operation. Verify that its status is “Completed”
15. Open a browser and put the following link:
http://localhost:8080/TOOLBOX/services/OGC06-131_standalone?request=GetRepositoryItem&id=aa00aa&service=CSW
16. Verify that a document can be downloaded and that it is the same referred in step 9
17. Move to the directory configured in the step 4 and check that the harvested document has been stored.
18. Execute the CTL test with the following parameters (see 3.4):
 - a. Target URL
`http://<ip>:<port>/TOOLBOX/services/OGC06-131_standalone`

Results

The validation has successfully completed without any error

Test pass/failure criteria

All tests have been performed successfully, reporting an update in the metadata stored.

References

4.4. Test Scenario TS_03

4.4.1. Test Case TS_03_01 MaxRecords handling

Prerequisites	<ul style="list-style-type: none"> • Toolbox RE 8.0 installed • ATS of CSW-ebRIM and CIM executed against both services OGC06-131_standalone and OGC07-038_gateway
Input	
Steps	<ol style="list-style-type: none"> 1. Log in into the Toolbox Administration Web pages 2. Select “OGC06-131_standalone service from the drop down 3. Select “Monitoring Center” 4. Select “ Display Synchronous Instances” 5. Select one of the GetRecords instances. 6. Inspect executed scripts in order to verify that the variable maxRecords has been automatically created and set by Toolbox 7. Repeat steps from 3 to 7 for the service “OGC07-038_gateway”
Results	MaxRecords variable is properly handled
Test pass/failure criteria	The value provided by the user during the service configuration is passed to the service logic scripts
References	

4.4.2. Test Case TS_03_02 Performances of the EO Product catalogue service

Prerequisites	<ul style="list-style-type: none"> • Toolbox RE 8.0 installed
Input	TS_03_01.ctl file in TS_03/TS_03_01 directory
Steps	<ol style="list-style-type: none"> 1. Execute the CTL test suite (see 3.4).
Results	Test have been successfully executed.
Test pass/failure	No errors have been reported by the execution of the test suite

criteria
References

4.4.3. Test Case TS_03_03 Accessing documentation

Prerequisites	<ul style="list-style-type: none">• Toolbox RE 8.0 installed
Input	
Steps	<ol style="list-style-type: none">1. Login into the Toolbox Administration Web Pages2. Select “Documents” in the Quick Links section3. Verify that the shown documentation provides information about installation and configuration tasks
Results	The documentation has been accessed
Test pass/failure criteria	
References	

5.Traceability matrices

5.1.Software requirements vs Test cases

Requirement	Test case
ERG-SR-TBX-FUN-010	TS_01_01, TS_01_02
ERG-SR-TBX-FUN-020	Covered by ATS-EO, ATS-CIM
ERG-SR-TBX-FUN-030	TS_01_02
ERG-SR-TBX-FUN-040	TS_01_01
ERG-SR-TBX-FUN-050	TS_02_01
ERG-SR-TBX-FUN-060	TS_02_01
ERG-SR-TBX-FUN-070	TS_02_01
ERG-SR-TBX-FUN-080	TS_02_01
ERG-SR-TBX-FUN-090	Covered by ATS-EO, ATS-CIM
ERG-SR-TBX-PER-100	TS_03_02
ERG-SR-TBX-INT-110	Covered by ATS-EO, ATS-CIM
ERG-SR-TBX-INT-120	TS_01_03, TS_01_04, TS_02_01, TS_03_01
ERG-SR-TBX-INT-130	Covered by ATS-EO, ATS-CIM

ERG-SR-TBX-INT-140	TS_02_01
ERG-SR-TBX-INT-150	TS_02_01
ERG-SR-TBX-OPE-160	TS_01_01, TS_01_02
ERG-SR-TBX-OPE-165	TS_01_01
ERG-SR-TBX-OPE-170	TS_01_01, TS_01_02
ERG-SR-TBX-OPE-180	TS_01_01, TS_01_02
ERG-SR-TBX-OPE-190	TS_01_04
ERG-RB-TBX-OPE-200	TS_01_03
ERG-SR-TBX-OPE-210	TS_02_01
ERG-SR-TBX-OPE-220	TS_01_01, TS_01_02
ERG-SR-TBX-OPE-240	TS_03_01
ERG-SR-TBX-OPE-250	TS_03_01
ERG-SR-TBX-DES-260	ALL
ERG-SR-TBX-DES-270	ALL
ERG-SR-TBX-SEC-280	This requirement will be covered with development coming from the HMA-T project

ERG-SR-TBX-SEC-290	This requirement will be covered with development coming from the HMA-T project
ERG-SR-TBX-SEC-300	This requirement will be covered with development coming from the HMA-T project
ERG-SR-TBX-SCD-310	Inspection
ERG-SR-TBX-SCD-320	Inspection
ERG-SR-TBX-SCD-330	Inspection
ERG-SR-TBX-OTH-360	TS_03_03
ERG-SR-TBX-VAL-370	TS_03_01

5.2. Test cases vs Software requirements

Requirement	Test case
TS_01_01	ERG-SR-TBX-FUN-010, ERG-SR-TBX-FUN-040, ERG-SR-TBX-OPE-160, ERG-SR-OPE-165, ERG-SR-TBX-OPE-170, ERG-SR-TBX-OPE-180, ERG-SR-TBX-OPE-220
TS_01_02	ERG-SR-TBX-FUN-010, ERG-SR-TBX-FUN-030, ERG-SR-TBX-OPE-160, ERG-SR-TBX-OPE-170, ERG-SR-TBX-OPE-180, ERG-SR-TBX-OPE-220
TS_01_03	ERG-SR-TBX-INT-120, ERG-RB-TBX-OPE-200
TS_01_04	ERG-SR-TBX-INT-120, ERG-SR-TBX-OPE-190
TS_02_01	ERG-SR-TBX-FUN-050, ERG-SR-TBX-FUN-060, ERG-SR-TBX-FUN-070, ERG-SR-TBX-FUN-080, ERG-SR-TBX-INT-120, ERG-SR-TBX-INT-150, ERG-SR-TBX-OPE-210
TS_03_01	ERG-SR-TBX-FUN-050, ERG-SR-TBX-FUN-060, ERG-SR-TBX-INT-120, ERG-SR-TBX-OPE-240, ERG-SR-TBX-OPE-250, ERG-SR-TBX-VAL-370

ERG-ATP-4100-INT :**Document Id**
1-30/04/2009 : **Issue**
0-30/04/2009 : **Revision**



TS_03_02	ERG-SR-TBX-PER-100, ERG-SR-TBX-INT-120
TS_03_03	ERG-SR-TBX-OTH-360