

Appendix C – Detailed Testing

C.1 Functional Testing

Testing environment of the Database Integration Tool simulates the medical laboratory which has three computerized systems.

1. Patient registration system
2. Automated medical Testing System
3. Data warehouse for patient’s medical test records.

Patient Registration Database

labtests(test_id:vvarchar,test_name:vvarchar, costs:vvarchar, upperlevel:int, lowerlevel:int)
patient(patient_id:vvarchar, name:vvarchar, nic:vvarchar, address:vvarchar, tp:vvarchar, age:int, ref_doctor:vvarchar)
test_patients(patient_id:vvarchar, test_id:vvarchar, datetest:datetime)

Automated Sample testing System Database

labreport(testDate:datetime, results:vvarchar, patient_id:vvarchar, test_id:vvarchar, test_name:vvarchar)
testinfo (lowerlevel:vvarchar, test_id:vvarchar, test_name:vvarchar, upperlevel:vvarchar)

Data warehouse

patient_history(patient_id:vvarchar, address:vvarchar, tp:vvarchar, testname:vvarchar, result:vvarchar)

C.1.1 Test cases

Test Case ID / Title	001 – New Connection to a Database Management System
Purpose	Verify new connection function and user validation
PreReq	Oracle database / schema

	Communication path for Oracle Database
Test Data	Database type – Oracle Schema – ORGTEST Username – test Password – XXXX (incorrect password) Host – 192.168.47.197
Expected Results	Error message
Steps	<ol style="list-style-type: none"> 1. Select “Connections” from toolbar 2. Form the opening window selects the “New Connection” option 3. Then New Database Management System connection window appeared 4. Above information is provided and pressed the test connection button.
Test Results	“Connection Failed” message appears and “add connection” button does not activated.
Test Status	completed

Test Case ID / Title	002 – New Connection to a Database Management System
Purpose	Add a new database connection and display it in the Connected data base Tree view
PreReq	Oracle database / schema Communication path for Oracle Database
Test Data	Database type – Oracle Schema – ORGTEST Username – test Password – XXXX (correct password) Host – 192.168.47.197
Expected Results	New connection reference is added to the list
Steps	<ol style="list-style-type: none"> 1. “Connections” tool bar option is selected 2. Connection window is appeared 3. Form that window selects the “New Connection” option 4. Then New Database Management system connection window appeared 5. Above information provided and pressed the test connection button. 6. “Connection ok” message appeared and “add connection” button is activated. 7. Select add connection button 8. Refresh database connection list
Notes	New reference to ORGTEST database and its table list added to database list.
Test Status	completed

Test Case ID / Title	003 - Creating a Federated query Description
Purpose	Creating a federated query description and save it as XML file
PreReq	<p>Patient Registration System and the Medical Testing system databases are integrated to create a view that contains the patient details and the medical test results.</p> <p>Combine “patient” table of the Patient Registration Database and the “labreport” table of the Automated Sample testing System Database</p> <p><u>Federated View</u></p> <p>patient_id , address , TP ,name , test_id , test_results , test_name</p>
Test Data	<p>Patient Registration Database select patient.TP,patient.patient_id,patient.address,patient.name from patient</p> <p>Automated Sample testing System Database select labreport.patient_id,labreport.test_id,labreport.results,labreport.test_name from labreport</p>
Expected Results	Federated query description file
Steps	<ol style="list-style-type: none"> 1. Open New Federated Data set Window from the tool bar or Integration -> “New Federated Data set” from the menu bar 2. Focus on the column list and click on the TP , patient_id , address , name columns of the patient table patient_id, test_id, results, test_name from the labreport table 3. Focus on the Table List field and click on the patient table and labreport tables. 4. Then focus on join text field and specify the join condition as labreport.patient_id = patient.patient_id <p>Press the “create” button and save as a XML file.</p>
Test Results	<p>An xml file is created.</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="no"?> <dbviewObject></pre>

	<pre> <dbdescription> <attributes>name:design:dept:increment:rank</attributes> <connections>Registration:TestSystem</connections> </dbdescription> <Registration> <connectionstring>Registration</connectionstring> <sql>select patient.TP,patient.patient_id,patient.address,patient.name from patient</sql> <join>patient_id</join> </Registration> <TestSystem> <connectionstring>TestSystem</connectionstring> <sql>select labreport.patient_id,labreport.test_id,labreport.results,labreport. test_name from labreport</sql> <join>patient_id</join> </TestSystem> </dbviewObject> </pre>
Test Status	accepted

Test Case ID / Title	004 - Creating a Federated query Description
Purpose	Validating Federated query Description
PreReq	For the federated data description join columns must be appeared in the selection listed column as well.
Test Data	<p>Patient Registration Database select patient.TP ,patient.address,patient.name from patient</p> <p>Automated Sample testing System Database select labreport.patient_id,labreport.test_id,labreport.results,labreport.test_name from labreport</p>
Expected Results	Error message with required column
Steps	<p>Select TP , patient_id , address , name columns of the patient table</p> <p>test_id, results, test_name from the labreport table</p> <p>and specify the join condition as labreport.patient_id = patient.patient_id</p>

	press create button
Test results	System display a message saying user hasn't selected the labreport.patient_id column for selection list.

Test Case ID / Title	005 - Load Federated data set
Purpose	Load results of a federated data description.
PreReq	Federated data description XML created in the test case 003
Test Data	<pre><?xml version="1.0" encoding="UTF-8" standalone="no"?> <dbviewObject> <dbdescription> <attributes> </attributes> <connections>Registration:TestSystem</connections> </dbdescription> <Registration> <connectionstring>Registration</connectionstring> <sql>select patient.TP,patient.patient_id,patient.address,patient.name from patient</sql> <join>patient_id</join> </Registration> <TestSystem> <connectionstring>TestSystem</connectionstring> <sql>select labreport.patient_id,labreport.test_id,labreport.results,labreport. test_name from labreport</sql> <join>patient_id</join> </TestSystem> </dbviewObject></pre>
Steps	<ol style="list-style-type: none"> 1. Click on "load Fd Dataset" button on toolbar or Integration - > Retrieve data from the menu. 2. Using the opening window select the federated data descriptions file. 3. Result window shows the extracted data from the configuration file. 4. Then user loads the data by clicking the "load data" button
Results	Results header and the data appeared
Validation Criteria	<p>A test database is used to validate the results. "labreport" and "patient" tables are replicated to the testing database.</p> <p>patient(patient_id:VARCHAR,name:VARCHAR,NIC:VARCHAR,addres</p>

	<p>s:VARCHAR,TP:VARCHAR,age:INT,ref_doctor:VARCHAR)</p> <p>labreport(testDate:datetime,results:varchar,patient_id:varchar,test_id:varchar,test_name:varchar)</p> <p>equivalent SQL to the federated data description</p> <pre> select p.TP,p.patient_id,p.address,p.name,l.test_id,l.results,l.test_name from patient p, labreport l where p.patient_id = l.patient_id </pre> <p>Derived results of the above query is same as the results of the federated data description</p>
Test Status	Accepted

Test Case ID / Title	006 – Save federated data internally
Purpose	Results of a federated query should be able to save in to internal database. Destination table and the federated data description file must have same name. Before saving data, tool should check availability of the destination table. If the table already exists, tool should inform that to the user and ask for the conformation to append data to the existing data or delete existing data.
PreReq	patientHistroy.XML created in the test case 003
Test Data	patientHistroy.XML
Steps	<ol style="list-style-type: none"> 1. Open patientHistroy.XML file and load the data. Then press the “GENERATE AND STORE” button 2. Open (main menu -> Query -> Query Analyzer Internal DB) “patientHistroy” table will be display at the right side table list panel Execute “select * from patientHistroy” 3. Open another federated data loading window and load the patientHistroy.XML file and load the data. And again try to save to data internally. 4. From the appearing message box select “No” option to append data. 5. Using “Query Analyzer Internal DB” we can see that same data is replicated twice. 6. Next round follows same procedure as above and select

	<p>“YES” to delete existing data.</p> <p>7. Using “Query Analyzer Internal DB” we can see that only one data set is available.</p>
Notes	Database Integration tool properly saves the federated data to the internal database.

Test Case ID / Title	007 – Import and save data internally
Purpose	To check whether user can query data from a connected database and save in to internal data base.
PreReq	<p>Test database with schema</p> <p>people (NIC varchar(12),familyMem int) occupation (NIC varchar(12),industry varchar(15),industrylevel int) occupationrank (rank int,description varchar(10))</p> <p>Above schema is contain sample of people and their status</p>
Test Data	<pre>select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1</pre>
Steps	<ol style="list-style-type: none"> 1. Open data importation window (main menu -> Data Migration -> Import/Export data) Or (tool bar -> Import Data) 2. Select personRegSystem from the connection menu. 3. Create above query and press “view results” 4. Press “save results” button 5. From the opening schema mapping window, Edit destination combo box value and enter new table name (PATIENTINFO). Press create table button. 6. Enter mapping column name of the result set into the source destination mapping window. 7. Next press start button to save data in to the internal database. 8. Open intermediate database query window and query data in the PATIENTINFO table
Validation Criteria	<p>Externally executed results of the below query should be same as the data stored in the external table.</p> <pre>select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r</pre>

	where p.NIC = o.NIC and o.ilevel = r.rank order by 1
--	--

Test Case ID / Title	008 – Create user view using internally stored data
Purpose	Internal database tables can be query (using SQL) to generate user defined views.
Test Data	PATIENTINFO table and PATIENTTESTINFO table
Steps	<ol style="list-style-type: none"> 1. Open internal database query analyzer 2. Execute query <pre> select t.NIC,t.NAME,t.AGE,p.WORKAS,p.TYPE,t.TEST_NAME,t.TES TRESULTS from PATIENTINFO p,PATIENTTESTINFO t where p.NIC = p.NIC </pre>
Validation criteria	PATIENTINFO and PATIENTTESTINFO tables are recreated in the MS SQL server and execute the same query in the MS SQL query analyzer and results are compared to ensure the correctness.

Test Case ID / Title	009 – Transfer data from internal database to a connected database
Purpose	Database Integration Tool can be used to transfer internal saved data to a connected database. In this task case we conformed that tool is capable to retrieve data from a SQL statement and load data to a connected database.
PreReq	Test case 008 is successfully completed
Test Data	Queried data from test case 008
Steps	<ol style="list-style-type: none"> 1. Follows the same steps of the test case 008 and load data. 2. Press export results button of the query analyzer window 3. From opening schema mapping window map source and destination fields. 4. Open query analyzer for the connected database and check internal query results and exported results.
Notes	Internal database can be queried using SQL. Also that data can be transferred to a connected database.

Test Case ID / Title	010 – Transfer data from one connected database to a another connected database
Purpose	To check the proper behavior of transferring data between connected databases.
Test Data	<p>Query data from patientRegSystem</p> <pre>select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1</pre>
Steps	<ol style="list-style-type: none"> 1. Open export window from data migration menu 2. Select patientRegSystem as the destination and execute above query. 3. Press export button on the window and from the opening window create a new table that maps to the query output. Destination database is dataWH which is oracle data warehouse database. 4. After specifying the mapping source and destination columns, press start button.
Verification	Follows same procedure of test case 008

Test Case ID / Title	011 - Export Data to an XML file
Purpose	<p>Database integration tool can be used to export queried data to an XML file. And that XML files can be compressed using XMILL compression tool.</p> <p>In this test case we check the proper execution of data export function.</p>
PreReq	Connection to Person Registration Database
Test Data	<pre>select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1</pre> <p>From Person registration system database.</p>
Steps	<ol style="list-style-type: none"> 1. Open data export window 2. Place above query in the text area and view results of the query. 3. In the property file specify the saving locations for data file and the data structure file.

	<p>exportXMLData=E:\\DbSi\\XMLXPORT\\exportData.xml exportXMLTab=E:\\DbSi\\XMLXPORT\\exportTab.xml</p> <p>4. Press export data button 5. Four files has been created in the E:\\DbSi\\ XMLXPORT folder exportData.xml / exportData.xmi exportTab.xml / exportTab.xmi</p>
Notes	

Test Case ID / Title	012 – Encrypting Exported files
Purpose	To apply further security compressed XML files can be encrypted.
PreReq	E:\\DbSi\\XMLXPORT\\exportData.xmi E:\\DbSi\\XMLXPORT\\exportTab.xmi
Test Data	E:\\DbSi\\XMLXPORT\\exportData.xmi E:\\DbSi\\XMLXPORT\\exportTab.xmi
Expected Results	Encrypted ZIP file having extension “.zip.enc” and contains the
Steps	<ol style="list-style-type: none"> 1. Open encryption window from (menu -> Data Migration -> Encryption Export XML) 2. Specify the decrypted file location/folder (E:\\DbSi\\XMLXPORT) 3. Specify the encrypted file name E:\\DbSi\\TempXML\\test 4. Without specifying the key location press start button
Notes	E:\\DbSi\\TempXML\\test.zip and E:\\DbSi\\TempXML\\test.zip.enc files created.

Test Case ID / Title	013 – Decrypting exported files
Purpose	An objective of this test case is to check the proper execution of decryption function. Users should be able to decrypt the created file to uncover compresses data and data description files.
PreReq	Encrypted file
Test Data	samplezipenf.zip.enc contains compressed files and general XML export files
Expected Results	exportData.xmi exportTab.xmi exportData.xml exportTab.xml
Steps	<ol style="list-style-type: none"> 1. Open encryption window from (menu -> Data Migration -> Encryption Export XML) 2. Specify the decrypted file location/folder

	<p>(E:\DbSi\ TempXML)</p> <ol style="list-style-type: none"> Specify the encrypted file name E:\DbSi\TempXML\ samplezipenf.zip.enc Select decryption option Without specifying the key location press start button
Notes	Decryption function works properly

Test Case ID / Title	014 – Restore encrypted data
Purpose	Compressed export using XMILL can be restored to a connected database.
PreReq	Compressed XML export files of a record set(exportData.xmi and exportTab.xmi)
Test Data	<p>exportData.xmi exportTab.xmi</p> <p>of the query</p> <pre>select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1</pre>
Steps	<ol style="list-style-type: none"> Open “restore XML” window from (main menu -> export data -> restore XML) Specify the folder containing XMI files And select the target database and the table-importdata(this case creating a new table in dataWH database) Select the create table option and press the start button Reload database/tables list Open connected database query window Execute “select * from importdata” on and dataWH database and <pre>select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1</pre> <p>query on personsRegSystem database</p>

	8. Compare the results
Notes	Import and export functions of the database integration tool works properly.

C.2 Performance Testing

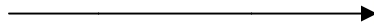
Performance is a very important factor for Database integration System like software. When software is testing for performance, bellow test cases are mainly focus on three functionalities.

1. Database Integration
2. Data archiving
3. Data transformation

Performance test is conducted using two Oracle database instances resides in a one server. And Database Integration Tool was run on a Dell inspiron 500i Lap Top.

Performance Testing Environment

Figure C.1 – Performance Test setup



Database Integration Tool

Fujitsu PrimePower 450
Oracle DB1 / Oracle DB2

Sever Configurations

Server Type	Fujitsu PrimePower 450
Processor	SPARC64TM V
Processor Clock speed	1.65GHz
Processor Quantity	2
Cache	Level 1 256KB/CPU
Main Memory	3 GB

Client Configurations

Server Type	Dell inspiron 500i
Processor	Intel Pentium 4
Processor Clock speed	1.7GHz

Processor Quantity	1
Main Memory	3 GB

C.2.1 Profiling Tool

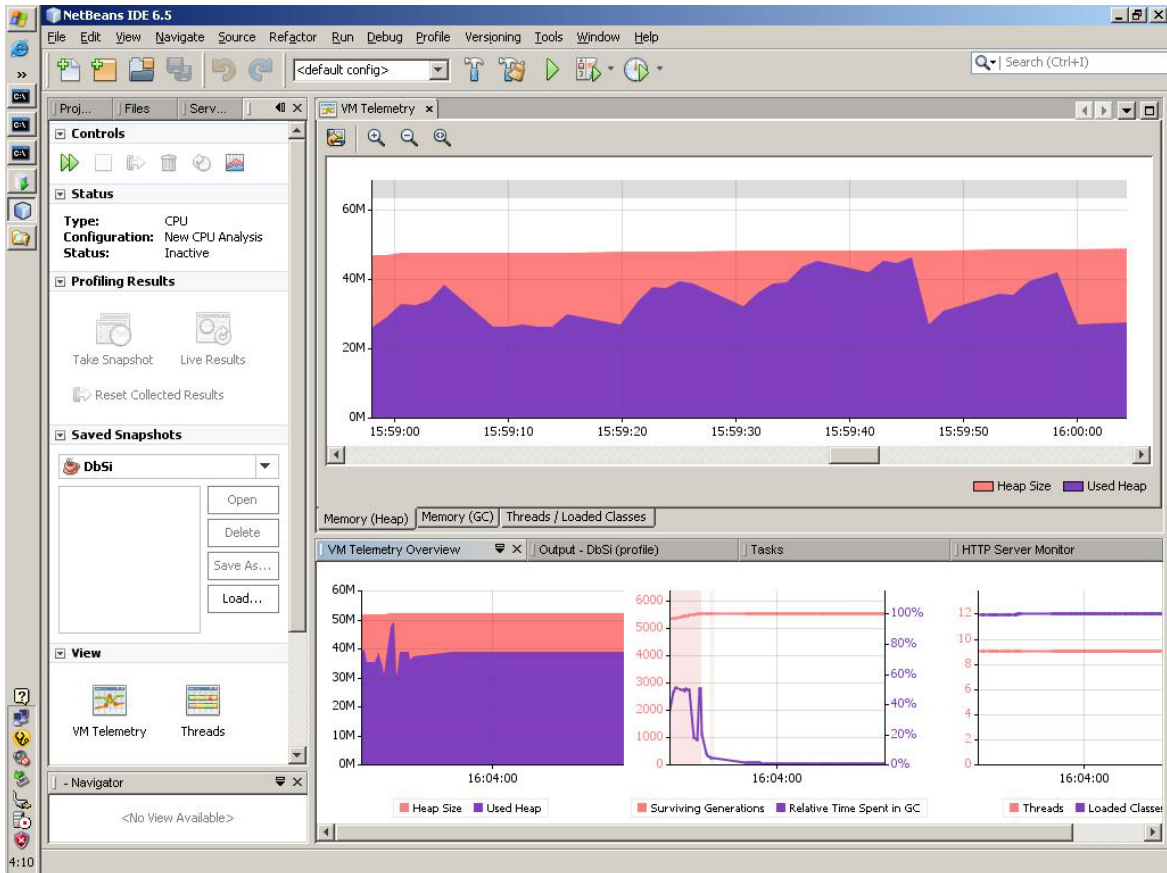


Figure C.2 – NetBeans Profiling Interface

NetBeans Profiling Tool is used to analyse the performance of Database Integration Tool. The NetBeans™ Profiler is an optional feature of the NetBeans Integrated Development Environment (IDE). The NetBeans Profiler is a powerful tool that provides important information about the runtime behavior of an application. Imposing relatively little overhead, the NetBeans Profiler tracks thread state, CPU performance, and memory usage. It uses innovative technology to allow you to tightly control exactly which parts of an application are profiled, resulting in reduced overhead and easier to interpret results. The profiled application can run locally or on a

remote system. And by being tightly integrated into the IDE workflow the NetBeans Profiler makes it easy to identify performance problems and memory leaks.

C.2.2 Test Cases

Test Case ID / Title	015 – Performance Testing - Database Integration
Purpose	In this test case we estimate the performance of the database integration according to data set size
PreReq	Two oracle database instances with schema can be used to create a federated database query.
Test Data	<p>Database 1 persontable(pid int, name VARCHAR2(20), locationid int);</p> <p>Database 2 locationdesc(locationid int, locationname VARCHAR2(20));</p> <hr/> <p>Federated Data description XML <dbviewObject> <dbdescription> <attributes></attributes> <connections>database1:database2</connections> </dbdescription> <database1> <connectionstring>database1</connectionstring> <sql>select PERSONTABLE.LOCATIONID, PERSONTABLE.NAME, PERSONTABLE.PID from PERSONTABLE</sql> <join>LOCATIONID</join> </database1> <database2> <connectionstring>database2</connectionstring> <sql>select LOCATIONDESC.LOCATIONNAME, LOCATIONDESC.LOCATIONID from LOCATIONDESC</sql> <join>LOCATIONID</join> </database2> </dbviewObject></p>

Steps	<p>Insert data to the persontable and locationdesc</p> <pre> begin for inc in 1..8000 loop insert into persontable(pid,name,locationid) values(inc,('name' inc),inc); end loop; end; begin for inc in 1..8000 loop insert into persontable(pid,name,locationid) values(inc,('name' inc),inc); end loop; end; </pre>
--------------	--

C.2.2.1 Performance Test Results Database Integration

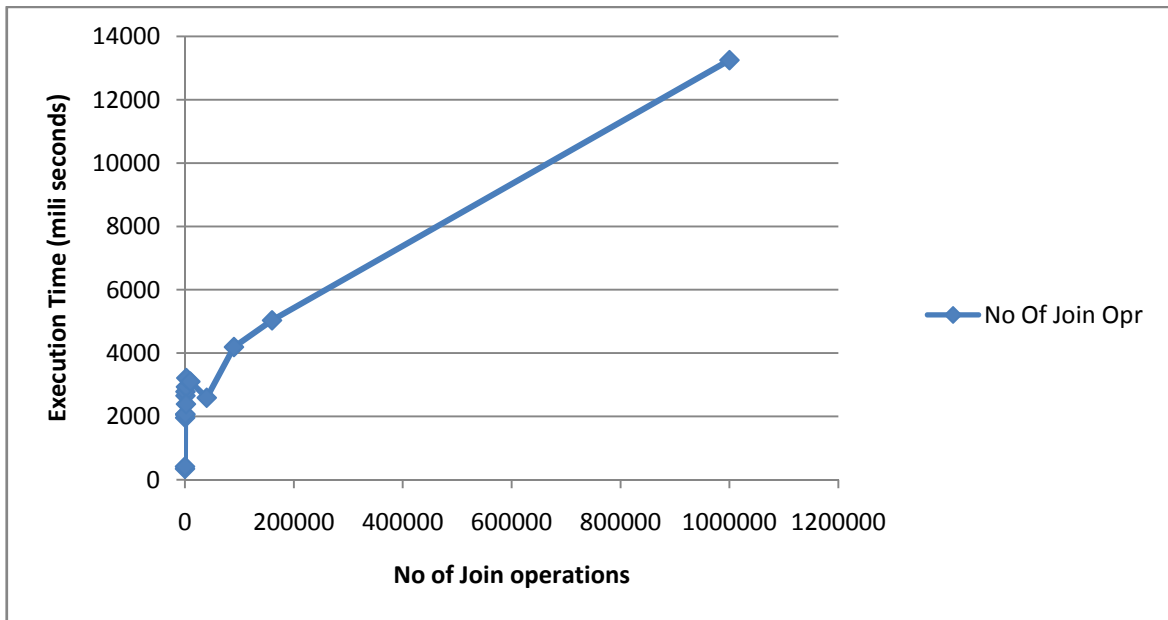
Execution Time is stated in milliseconds. Recorded time is medium of three execution rounds on same data set.

	Persontable No Of Records	locationdesc No Of Records	No Of Join Opr	Execution Time (ms)
1	10	10	100	345
2	20	10	200	416
3	40	10	400	1957
4	60	10	600	2054
5	70	10	700	2075
6	50	20	1000	2652
7	100	10	1000	2781
8	100	20	2000	2933
9	50	40	2000	2389
10	50	50	2500	3212
11	100	100	10000	3100
12	200	200	40000	2588
13	300	300	90000	4190
14	400	400	160000	5033
15	1000	1000	1000000	13251

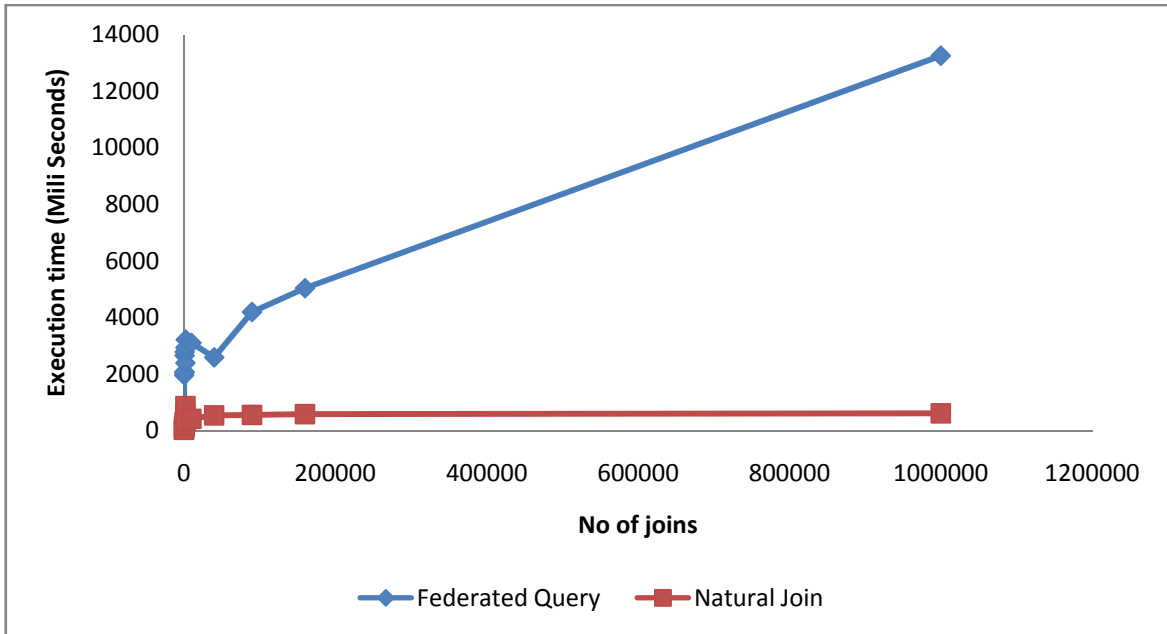
Table C.1 – Execution times federated query.

	persontable No Of Records	locationdesc No Of Records	No Of Join Opr	Execution Time
1	10	10	100	38
2	20	10	200	27
3	40	10	400	167
4	60	10	600	172
5	70	10	700	153
6	50	20	1000	182
7	100	10	1000	315
8	100	20	2000	569
9	50	40	2000	867
10	50	50	2500	355
11	100	100	10000	417
12	200	200	40000	536
13	300	300	90000	554
14	400	400	160000	583
15	1000	1000	1000000	612

Table C.2 – Execution Time Natural join operation

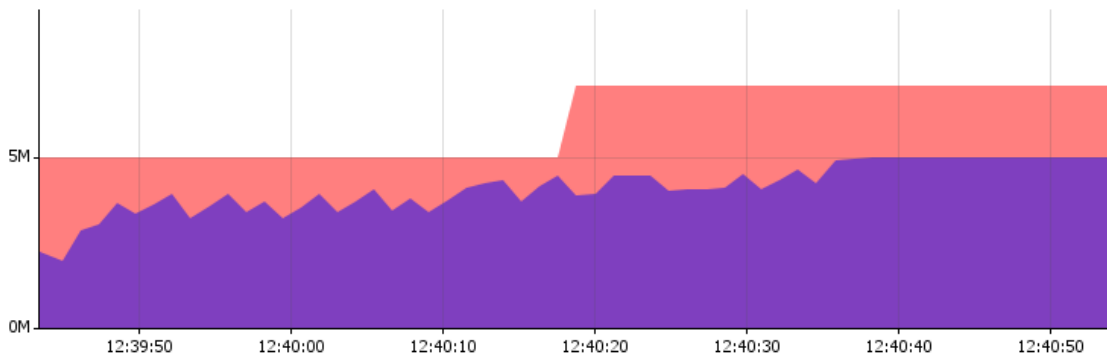


Graph C.1 – No of join operation Vs Execution time



Graph C.2 – Comparison of execution times (Federated query and natural join)

Above graph compare and contrast execution time of federated query and Natural join of same number of execution rounds



Graph C.3 - Memory usage graph – Database integration (200 records)

Heap allocation / Used heap memory

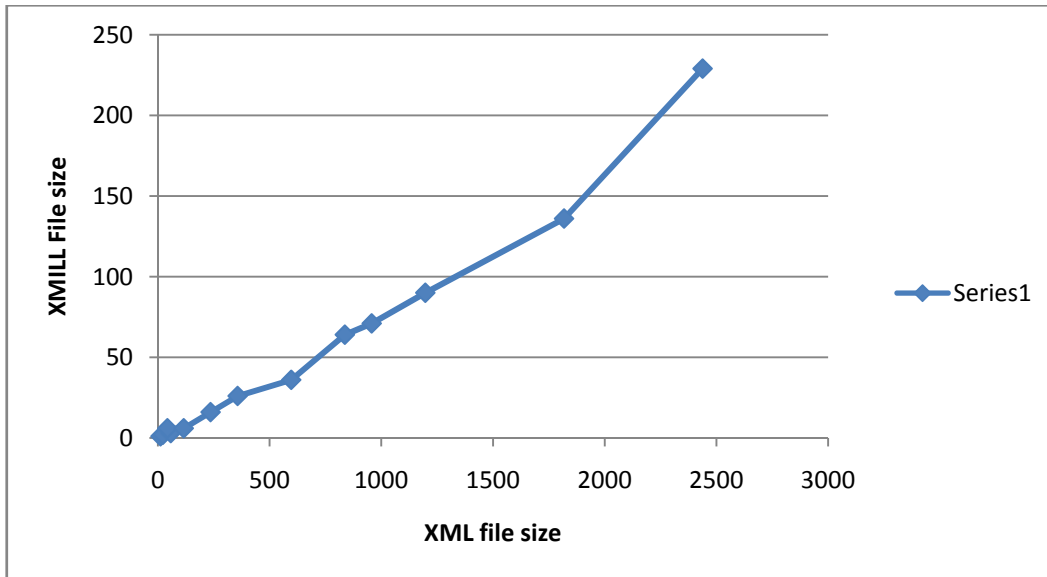
Test Case ID / Title	016 – Data archiving performance test
Purpose	In this test case we check compression rate, memory usage and time against the number of records.
PreReq	Oracle database instance
Test Data	<p>Database 2</p> <pre> persontable(pid int, name VARCHAR2(20), locationid int); locationdesc(locationid int, locationname VARCHAR2(20)); </pre> <p>Query</p> <pre> select pid , name ,locationid ,locationname from persontable p, locationdesc l where p.locationid = l.locationid; </pre>
Steps	<ol style="list-style-type: none"> 1. Open data import window place the above query on the text area. 2. Select database2 from connections list. 3. Change record count 4. <pre> begin for inc in 1..3000 loop insert into locationdesc(locationid,locationname) values(inc,('location' inc)); end loop; end; begin for inc in 1..3000 loop insert into persontable(pid,name,locationid) values(inc,('name' inc),inc); end loop; end; </pre> 5. Execute export XML
Notes	Inside \DbSi\XMLXPOR folder four files created for each export

C.2.2.2 Compression and archiving test results.

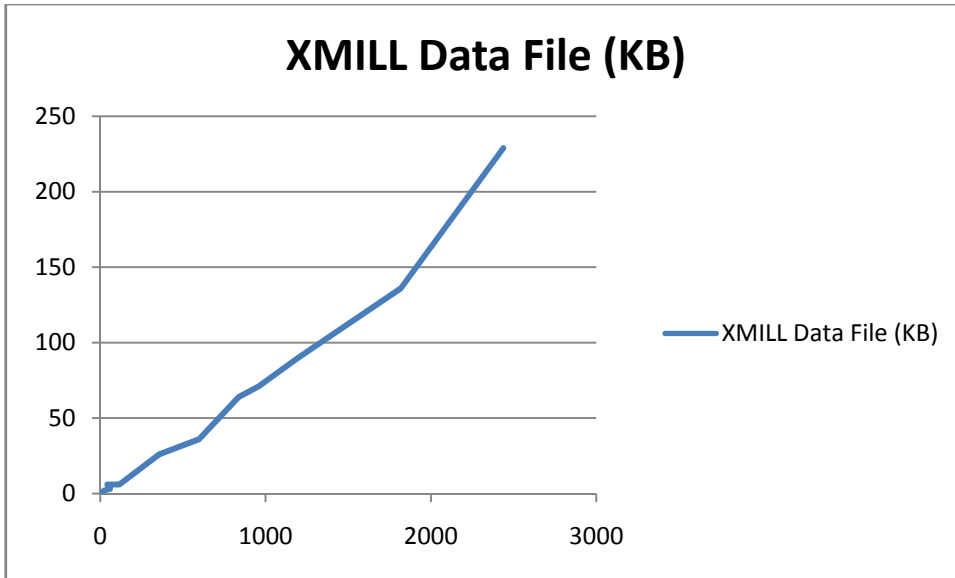
Execution Time is stated in milliseconds. Recorded time is medium of three execution rounds on same data set.

	No of Records	Time	Memory Heap	Memory Used	XML Data file size (KB)	XMILL Data File (KB)	Compression Rate %
1	100	5787	6	5	12	1	8.33
2	200	6609	7	6	23	2	8.70
3	500	7890	7	6	58	3	5.17
4	800	8038	7	6.5	43	6	13.95
5	1000	7714	8	6.5	116	6	5.17
6	2000	7814	9	6	236	16	6.78
7	3000	12391	9	7	357	26	7.28
8	5000	19175	11	8	597	36	6.03
9	7000	23729	12	11	837	64	7.65
10	8000	28968	12	12	957	71	7.42
11	10000	33098	12	12	1197	90	7.52
13	15000	43067	25	17	1818	136	7.48
15	20000	68987	30	22	2438	229	9.39

Table C.3 – Data compression and archiving



Graph C.4 – XMILL compression

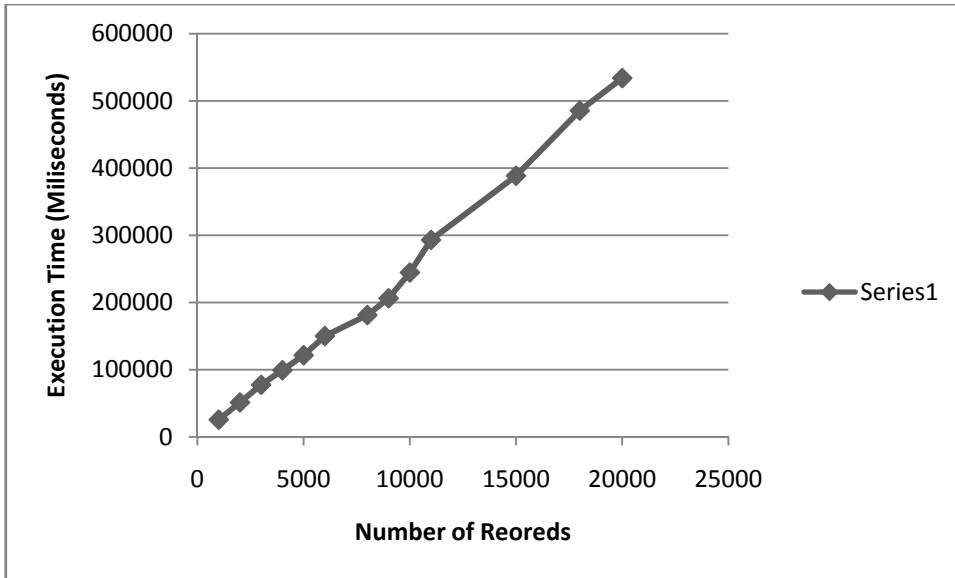


Graph C.5 – Number of record Vs XMILL file size

C.2.2.3 Data Transmission Performance Testing

	No of Records	Time (ms)	Memory Usage (MB)(heap - actual)
1	1000	25616	6 - 4
2	2000	51344	7 - 4
3	3000	77588	9-6
4	4000	99232	7-5
5	5000	121645	7-6
6	6000	150166	8-5
7	8000	181365	7-6
8	9000	206366	7-6
9	10000	244669	7-6
10	11000	293127	7-6
11	15000	388559	7-6
12	18000	485515	7-6
13	20000	533938	7-6

Table C.4 – Data transmission Test Results



Graph C.6 – Data Transmission Timing

C.3 Recovery Testing

Structural database configuration used in Recovery testing is same which is used in functional testing.

Test Case ID / Title	017 - Database Integration – A Database failed while integration process is running
Purpose	While database integration processes is running participating database may fail. In this test case we check system behavior if a database connection failed while integration process is running.
PreReq	Error free federated database description file. Required database connections.
Test Data	patientHistroy.XML
Expected Results	Error with failed database connection.
Steps	<ol style="list-style-type: none"> 1. Click on “load Fd Dataset” button on toolbar or Integration - > Retrieve data from the main menu. 2. Using the opening window select the federated data descriptions file. 3. Result window shows the extracted data from the configuration file. 4. Shutdown a one database 5. Press load data option
Notes	Log window shows and Red color massage saying that Result Set is empty because connection cannot be established. If database

	connection if failed while integration process is running it again shows that Error in Result set providers.
--	--

Test Case ID / Title	018 – Database Integration – Application Failed while Integration process is running
Purpose	While database Integration process is running tool itself can be failed (software).
PreReq	
Test Data	patientHistroy.XML
Expected Results	Should be able to restart and run the integration process without any error
Steps	<ol style="list-style-type: none"> 1. Click on “load Fd Dataset” button on toolbar or Integration - > Retrieve data from the main menu. 2. Using the opening window select the federated data descriptions file. 3. Result window shows the extracted data from the configuration file. 4. While data loading is running kill java runtime environment process 5. Then application closes immediately 6. Restart the application and load data for the patientHistroy.XML
Notes	Application can be crash and restart.

Test Case ID / Title	019 – Application crashers while integrated query data saving internally
Purpose	Application may crashed while saving federated data internally
PreReq	Sufficient number of records for patientHistroy.XML
Test Data	patientHistroy.XML
Expected Results	New Table without data
Steps	<ol style="list-style-type: none"> 1. Start loading federated data to the internal database. 2. While data loading kill java runtime process. 3. Restart application and query internal database.
Notes	PATIENTINFO table in empty.

Test Case ID / Title	020 – A Connected database crashes while federated data saving
Purpose	In this test we check the consequences of crashing a connected database while federated data saving
PreReq	Federated query and required database connections

Test Data	patientHistroy.XML
Expected Results	Table structure is created internally but data is not saved
Steps	<ol style="list-style-type: none"> 1. Start loading federated data to the internal database. 2. While data loading kill Patient Registration Database process.
Notes	PATIENTINFO table in empty.

Test Case ID / Title	021 – Source database crashers while data importing to intermediate database
Purpose	Database instance may be crashed while data is transmitting to another database.
PreReq	personRegSystem Database connection
Test Data	select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1
Expected Results	Only table structure is available
Steps	<ol style="list-style-type: none"> 1. Using test query execute “Save Results Set” window. 2. While processing shut down the source database
Notes	Logging window shows an error saying that data source is not available, and process terminates.

Test Case ID / Title	022 - Source database crashers while data importing to connected database.
Purpose	Database instance may be crashed while data is transmitting to another database.
PreReq	personRegSystem Database connection
Test Data	select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1
Expected Results	Only table structure is available
Steps	<ol style="list-style-type: none"> 1. Using test query, execute “Save Results Set” command. 2. While processing shut down the source database
Notes	Logging window shows an error saying that data source is not available, and process terminated.

Test Case ID / Title	023 – Destination Database crashes while data importing
Purpose	Testing Database integration tool's behavior if target database is failed while transmitting data.
PreReq	personRegSystem (MySQL) Database connection dataWH (Oracle) database connection
Test Data	select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1
Expected Results	Only table structure is available
Steps	<ol style="list-style-type: none"> 1. Using test query, execute "export result" command. 2. While processing shut down the destination database
Notes	Logging window shows an error saying that destination is not available, and process is terminated.

Test Case ID / Title	024 – Source Database crashes while data is exporting to XML
Purpose	Source connection may be crashed while data is exporting to XML file.
PreReq	personRegSystem Database connection
Test Data	select p.NIC NIC, o.industry workas ,r.description type from people p,occupation o,occupationrank r where p.NIC = o.NIC and o.ilevel = r.rank order by 1
Expected Results	Terminate data exporting
Steps	<ol style="list-style-type: none"> 1. Using test query, execute "export XML" command. 2. While processing shut down the source database
Notes	After few seconds Tool logs a message that source database cannot be reached. XML data file exist with already exported data.

Test Case ID / Title	025 – Destination database crashes while transmitting internally stored data to a connected database instance
Purpose	Internally stored data can be query and transmit to a connected database. If transmission link / destination is not reachable while running destination database should be consistent.
PreReq	Connection to dataWH(oracle) database
Test Data	

Expected Results	In this case only destination table is available.
Steps	<ol style="list-style-type: none"> 1. Open internal database query analyzer 2. Execute query select t.NIC,t.NAME,t.AGE,p.WORKAS,p.TYPE,t.TEST_NAME,t.TES TRESULTS from PATIENTINFO p,PATIENTTESTINFO t where p.NIC = p.NIC 3. Press export data button and soon operation started shut down oracle database dataWH instance / shutdown listener
Notes	New destination table available in the dataWH database and doesn't have data.

Test Case ID / Title	026 – Database Integration Tool crashers while transmitting internally saved data.
Purpose	Test consequences of crashing the tool while transmitting internally stored data.
PreReq	
Test Data	
Expected Results	Destination database should be consistent
Steps	<ol style="list-style-type: none"> 1. Open internal database query analyzer 2. Execute query select t.NIC,t.NAME,t.AGE,p.WORKAS,p.TYPE,t.TEST_NAME,t.TESTRE SULTS from PATIENTINFO p,PATIENTTESTINFO t where p.NIC = p.NIC 3. Press export data button and soon operation started kill java process.
Notes	Only destination table exist. Therefore we can guarantee that Database integration tool ensure consistency over database operations.