

## **Interoperability Testing for DAIS Working Group Specifications**

### Status of This Memo

This memo provides information regarding the testing of interoperability for implementations of the DAIS Working Group Specifications. It does not define any standards or technical recommendations. Distribution is unlimited.

### Copyright Notice

Copyright © Open Grid Forum (2006). All Rights Reserved.

### **Abstract**

The Data Access and Integration Services (DAIS) Working Group has submitted several specifications to the Global Grid Forum (GGF) recommendation track. This document proposes a process by which the interoperability of implementations of the specifications can be tested. The purpose of this document is to obtain sign-off on the proposed process so that the group can be confident that if the process is followed, a report on the implementation of the process will be accepted as an appropriate way of fulfilling the interoperability condition listed in [GFD.1].

Contents

Abstract .....	1
1. Introduction .....	3
2. Proposed Process .....	3
3. Testing Mandatory and Optional Features .....	3
3.1 WS-DAI .....	4
3.1.1 Mandatory Operations and Properties .....	4
3.1.2 Optional Operations, Parameters, Properties and Property Values .....	4
3.2 WS-DAIR .....	5
3.2.1 Mandatory Operations and Properties .....	5
3.2.2 Optional Operations, Parameters, Properties and Property Values .....	5
3.3 WS-DAIX .....	5
3.3.1 Mandatory Operations and Properties .....	5
3.3.2 Optional Operations, Parameters, Properties and Property Values .....	6
4. Implementing the Approach .....	6
4.1 Testing Strategy .....	6
4.2 Implementing test clients .....	7
4.3 Reporting of test results .....	7
4.4 Assessment of Implementation Viability .....	7
5. Security Considerations .....	8
6. Conclusion .....	8
7. Author Information .....	8
8. Intellectual Property Statement .....	9
9. Disclaimer .....	9
10. Full Copyright Notice .....	9
References .....	9

## 1. Introduction

The initial specifications of the DAIS Working Group [WS-DAI, WS-DAIR, WS-DAIX] have been submitted to the GGF Recommendation process. This document describes a process by which it is proposed to test the specifications for interoperability, as required in [GFD.1]. As a definition for interoperability, that of the European Telecommunications Standards Institute is applicable here (<http://portal.etsi.org/mbs/Testing/interop/interop.asp>):

Interoperability testing is the activity of proving that end-to-end functionality between (at least) two communicating systems is as required by those base systems' standards.

The document is specific to the DAIS specifications in two respects:

1. The nature of the interoperability required. Implementations of the specifications do not call themselves or each other. As such, the requirement is only to test *client interoperability*: that multiple implementations of a specification provide consistent results to a suite of tests.
2. The challenges identified for designers of a comprehensive test regime, which follow from certain features of the specifications.

The WS-DAI family of specifications define behaviours that may vary between different implementations of the specifications, and such differences principally reflect variety in the data resources to which the services provide access. It is therefore impractical to test every behaviour that could possibly be exhibited by one of the services without writing tests aimed at specific data resource management systems or by implementing or emulating the underlying data resources. However, the testing process detailed in this document aims to test all mandatory features of the specifications and as many optional features as is practical given the varying capabilities of the resource management systems accessed by the services that are being tested.

## 2. Proposed Process

The aim of the process is to test for client interoperability. For each of the specifications, it is proposed to:

1. Define a scenario that can be used as the basis of interoperability testing. This scenario will include populated databases; the structure and contents of the database affect test results.
2. Develop a sequence of operations and property queries that test implementations of the specifications under that scenario.
3. For each request/response in this sequence, define the correct inputs and outputs.
4. Execute the sequence of requests, checking the response received in each case.
5. Document the tests and the results of the tests in an Experimental document, as required in [GFD.1]. It is assumed that there is no requirement that the tested implementations of the specifications are widely available or supported longer term.

## 3. Testing Mandatory and Optional Features

The testing process *aspires* to test every mandatory feature in multiple implementations and every optional feature in at least one implementation<sup>1</sup>. In essence, the specifications define web services using WSDL, where the behaviour of a service is described by the values of named properties.

---

<sup>1</sup> In a few cases this document makes assumptions about what is mandatory and what is optional in the DAIS specifications; where this has been necessary feedback to that effect will be logged on the DAIS GridForge site for action.

Constructs of the following types may be mandatory in the specifications:

1. Operations.
2. Properties.

Constructs of the following types may be optional in the specifications:

1. Operations.
2. Operation parameters.
3. Properties.
4. Specific property values.

The mandatory and optional features of the WS-DAI, WS-DAIR and the WS-DAIX specifications are described in the following subsections.

### **3.1 WS-DAI**

#### **3.1.1 Mandatory Operations and Properties**

The following operations are mandatory:

1. CoreDataAccess::GetDataResourcePropertyDocument.
2. CoreDataAccess::DestroyDataResource.

All properties except ParentDataResource and DataResourceDescription in WS-DAI are mandatory.

#### **3.1.2 Optional Operations, Parameters, Properties and Property Values**

The following operations are optional:

1. CoreDataAccess::GenericQuery.
2. CoreResourceList::GetDataResourceList.
3. CoreResourceList::Resolve.

The following operation parameters are optional:

1. DatasetFormatURI of operations instantiating the template for Access operations.
2. PortTypeQname of operations instantiating the template for Factory operations.
3. ConfigurationDocument of operations instantiating the template for Factory operations.
4. PreferredTargetService of operations instantiating the template for Factory operations. In this case, if provided, the value may be ignored by the service.

The following properties are optional:

1. ParentDataResource.
2. DataResourceDescription.

Any service may support any of the following property values for which an enumerated list is provided in the specifications, namely:

1. DataResourceManagement (ExternallyManaged or ServiceManaged).
2. ConcurrentAccess (true or false).
3. Readable (true or false).
4. Writable (true or false).
5. TransactionInitiation (NotSupported, Automatic, Manual).
6. TransactionIsolation (NotSupported, ReadUncommitted, ReadCommitted, RepeatableRead, Serializable).

7. ChildSensitiveToParent (Insensitive, Sensitive).
8. ParentSensitiveToChild (Insensitive, Sensitive).

## 3.2 WS-DAIR

### 3.2.1 Mandatory Operations and Properties

The following operations are mandatory:

1. SQLAccess::GetSQLPropertyDocument.
2. SQLAccess::SQLExecute.
3. SQLAccessFactory::SQLExecuteFactory.
4. SQLResponse::GetSQLResponsePropertyDocument.
5. SQLResponse::GetSQLResponseItem.
6. SQLResponse::GetSQLRowSet.
7. SQLResponse::GetSQLUpdateCount.
8. SQLResponse::GetSQLReturnValue.
9. SQLResponse::GetSQLOutputParameter.
10. SQLResponse::GetSQLCommunicationsArea.
11. SQLResponseFactory::SQLRowsetFactory.
12. SQLRowset::GetSQLRowsetPropertyDocument.
13. SQLRowset::GetTuples.

The following properties are mandatory:

1. NumberOfSQLRowsets (from SQLResponse).
2. NumberOfSQLUpdateCounts (from SQLResponse).
3. NumberOfSQLReturnValues (from SQLResponse).
4. NumberOfSQLOutputParameters (from SQLResponse).
5. NumberOfSQLCommunicationAreas (from SQLResponse).
6. RowSchema (from SQLRowset).
7. NoOfRows (from SQLRowset).
8. AccessMode (from SQLRowset).

### 3.2.2 Optional Operations, Parameters, Properties and Property Values

No optional operations are defined in WS-DAIR.

The following properties are optional:

1. SchemaDescription.

Any service may support any of the following property values for which an enumerated list is provided in the specifications, namely:

1. AccessMode (Forward or Random).

## 3.3 WS-DAIX

### 3.3.1 Mandatory Operations and Properties

The following operations are mandatory:

1. XMLCollectionAccess::GetCollectionPropertyDocument
2. XMLCollectionAccess::AddDocuments
3. XMLCollectionAccess::RemoveDocuments
4. XMLCollectionFactory::DocumentSelectionFactory
5. XMLSequenceAccess::GetXMLSequencePropertyDocument

## 6. XMLSequenceAccess::GetItems

An implementation of WS-DAIX must support at least one query language interface, and thus it is mandatory for an implementation to support either:

1. XQueryAccess::XQueryExecute
2. XQueryFactory::XQueryExecuteFactory

or

1. XPathAccess::XPathExecute
2. XPathFactory::XPathExecuteFactory

The following properties are mandatory:

1. TopLevelCollection (from XMLCollection)
2. NumberOfDocuments (from XMLCollection)
3. SupportsCollections (from XMLCollection)
4. SupportsCollectionNesting (from XMLCollection)
5. SupportsSchemas (from XMLCollection)
6. NumberOfItems (from XMLSequence)

### 3.3.2 Optional Operations, Parameters, Properties and Property Values

The following operations are optional:

1. XUpdateAccess::XUpdateExecute
2. XMLCollectionAccess::AddSchema
3. XMLCollectionAccess::RemoveSchema
4. XMLCollectionAccess::CreateSubcollection
5. XMLCollectionAccess::RemoveSubCollection
6. XMLCollectionFactory::CollectionSelectionFactory

The following operation parameters are optional:

1. The CollectionName parameter of XML collection access and update operations. If this parameter is not specified then the top level collection is assumed.

Any service may support any of the property values for which an enumerated list is provided in the specifications, namely:

1. SupportsCollections (true or false).
2. SupportsCollectionNesting (true or false).
3. SupportsSchemas (true or false).

## 4. Implementing the Approach

### 4.1 Testing Strategy

As described in Section 2, testing will be based on hypothetical scenarios, one for the XML realization and one for the relational realization. For each scenario used as the basis for testing, a test suite will be defined. As the WS-DAI specification cannot be tested independently, both the WS-DAIR and WS-DAIX test suites will include tests covering the mandatory and optional features of WS-DAI. A test suite consists of:

1. A set of initial conditions that must be satisfied before the test suite can be executed.

2. A sequence of test cases, where each test case consists of either an operation or property query.

The test cases will be categorized as applying to the base WS-DAI specification, a specific realization, or both. Additionally, tests will be categorized as being either optional or mandatory with respect to the classification of operations, parameters, properties and property values presented in this document. Each test case specifies the contents of the request message XML instance that should be sent to the DAIS service and the expected output in terms of the response message received from the service. In some cases it will be impossible to specify the exact textual content of a response message, for example, different database management systems may return results in a different order even though the resource properties of the DAIS services which represent them are the same. In cases where the exact content of the response message cannot be specified, the response should be validated against its schema and the existence of specific values should be tested, where necessary without sensitivity to the ordering and different format of valid results. The execution of some test cases will modify the state of the resources exposed by the services being tested, and therefore the set of test cases that comprise a test suite must be executed completely and in the correct order. Fault testing is an essential part of each defined scenario. Test cases will be defined to test that each operation defined by the DAIS specifications generates response messages with the required fault elements that adhere to the specifications.

A test case is passed when the actual response message matches that of the expected response defined by the test case. A client needs to execute the test suite only once using a specific implementation, there is no need to execute multiple instances of the same test. Tests related to features classified as optional in this document do not impact upon the assessment of interoperability between multiple implementations and function as a quality assurance test for the specifications.

#### **4.2 Implementing test clients**

In general, it is expected that Apache Axis and Java will be utilised by the developers of DAIS implementations. The test suites themselves will therefore be written in Java utilising stubs automatically generated from the DAIS WSDL by the tools provided by Axis. There are two acceptable approaches which can be taken when testing an implementation of the specifications:

1. Use Axis to generate the stubs from the DAIS WSDL and execute the tests using the provided test suites.
2. Follow another approach to generating the client, for example using different tooling or a different programming language. If this approach is taken, the test suite must be reimplemented and the source code made publically available.

#### **4.3 Reporting of test results**

The outcome of the testing process for a DAIS specification is an Experimental document detailing the tests that have been executed and their results. The testing process for a specific specification is successful when all mandatory tests for that specification have been passed by two independent implementations and each optional test has been successfully implemented at least once. Unless previously published elsewhere, the Experimental document should either provide the client code used to execute the tests or specify how this code may be obtained.

The viability of including specific features of the DAIS specifications within test suites is now discussed.

#### **4.4 Assessment of Implementation Viability**

The following is an initial assessment as to the viability of implementing the proposed process at a level that meets the aspired level of testing from Section 3.

The development of two client interoperable implementations for the types of construct that may be mandatory in the specifications:

1. Operations. The only problem anticipated is that some XML databases support only XPath, and thus some implementations of WS-DAIX will not implement the XQuery operations.
2. Properties. No problems are anticipated.

The development of a single implementation for the types of construct that may be optional in the specifications:

1. Operations. No problems are anticipated.
2. Operation parameters. The only problem anticipated is the PreferredTargetService parameter of factory port types, which is a challenging functionality to implement.
3. Properties. No problems are anticipated.
4. Specific property values. It is unrealistic to test for all values, because some parameters describe complex behaviours, e.g. ChildSensitiveToParent, ParentSensitiveToChild, TransactionIsolation. The behaviours attributable to these properties depend upon the underlying data resources exposed by the services and cannot be practically tested without writing tests aimed at specific data resource management systems. In addition, some parameter values involve composition with other specifications, e.g. TransactionInitiation.

In the specifications, support for WSRF data resources is optional. There will be no tests of the behavior of messages defined in the WSRF specifications; their behavior is assumed to have been fully tested elsewhere. The WS-DAI specification defines similar operations to the WSRF specifications for property access. Regardless of the method used (the WSRF or WS-DAI operations), access is to the same underlying properties. It is trivial to test access to the same properties using different operations and there is therefore no requirement to test property access using the WSRF defined operations, as tests which access the same properties, using the WS-DAI defined operations, will already exist in the test suites. The tests should be able to run with minimal modifications using both WSRF and non-WSRF data resources.

## 5. Security Considerations

This document does not address security issues.

## 6. Conclusion

This document has described a process by which it is proposed to test the interoperability of the specifications produced by the DAIS Working Group.

## 7. Author Information

Steven Lynden,  
School of Computer Science,  
University of Manchester,  
Oxford Road,  
Manchester M13 9PL,  
United Kingdom.

Norman W. Paton,  
School of Computer Science,  
University of Manchester,  
Oxford Road,



Manchester M13 9PL,  
United Kingdom.

Dave Pearson,  
Oracle Corporation Ltd,  
Thames Valley Park,  
Reading,  
Berkshire RG6 1RA,  
United Kingdom.

## 8. Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

## 9. Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

## 10. Full Copyright Notice

Copyright (C) Open Grid Forum (2006). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

## References

[GFD.1]

C. Catlett, GGF Document Series, Global Grid Forum Document GFD.1  
(<http://www.gridforum.org/documents/GFD.1.pdf>).

[WS-DAI]

M. Antonioletti, M. Atkinson, S. Laws, S. Malaika, N. W. Paton D. Pearson and G. Riccardi *Web Services Data Access and Integration (WS-DAI) Specification Version 1.0*. Draft, Global Grid Forum (<http://forge.gridforum.org/tracker/index.php?aid=1711>)

[WS-DAIR]

M. Antonioletti, B. Collins, A. Krause, S. Laws, J. Magowan, S. Malaika, N. W. Paton *Web Services Data Access and Integration – The Relational Realisation (WS-DAIR) Specification Version 1.0*. Draft, Global Grid Forum  
(<http://forge.gridforum.org/tracker/index.php?aid=1712>)

[WS-DAI]

M. Antonioletti, S. Hastings, A. Krause, S. Langella, S. Laws, S. Malaika, N. W. Paton *Web Services Data Access and Integration – The XML Realisation (WS-DAIX) Specification Version 1.0*. Draft, Global Grid Forum  
(<http://forge.gridforum.org/tracker/index.php?aid=1713>)