Open GIS Consortium, Inc.

OpenGIS® Simple Features Test Protocol

For OLE/COM

99-037r2

Release Date: August 6, 1999
## Contents

0  PREFACE .......................................................................................................................... IV

0.1  SUBMITTING COMPANIES ................................................................................................. IV

0.2  SUBMISSION CONTACT POINTS ........................................................................................ IV

0.3  DOCUMENT CONVENTIONS .............................................................................................. IV

1  OVERVIEW .......................................................................................................................... 5

2  TEST DESCRIPTION AND SCOPE ..................................................................................... 6

2.1  OLE DB PROVIDER ......................................................................................................... 7

2.2  COM INTERFACES ........................................................................................................... 9

3  TEST DATA .......................................................................................................................... 12

3.1  TEST DATA SEMANTICS .................................................................................................. 12

3.2  TEST DATA POINTS AND COORDINATES .................................................................. 13

4  CONFORMANCE ITEMS ..................................................................................................... 16

4.1  CONFORMANCE ITEMS FOR OLE DB PROVIDERS ....................................................... 16

4.2  CONFORMANCE ITEMS FOR COM INTERFACES .......................................................... 17
0 Preface

0.1 Submitting Companies

The following companies submitted this test protocol document:

• Computer Aided Development Corporation (Cadcorp) Ltd.
• Intergraph Corporation

0.2 Submission Contact Points

All questions about the joint submission should be directed to:

Adam Gawne-Cain
Cadcorp Ltd
4 Fitzroy Square
London W1P 5AH
United Kingdom
adam@cadcorpdev.co.uk

Sam Bacharach
Intergraph Corporation
2051 Mercator Drive
Reston, VA 20191
703-264-5711
sabachar@ingr.com

0.3 Document Conventions

The Courier New font has been used to indicate code segments.
1 Overview

This document describes the conformance testing guidelines that the Open GIS Consortium will use in the testing of software implementations for conformance to its specification entitled Open GIS Simple Features for OLE/COM, Revision 1.1.

The Open GIS Consortium, Inc. (OGC) maintains a brand (in the form of a certification mark) that cannot be used in connection with a software product by any organization unless they have submitted a software product to OGC’s conformance testing, successfully completed this testing, and received a certificate stating such success. Organizations that have earned the certification mark may use it in ways defined within this document. This set of rules ensures that users who buy products that are branded can be sure that the products carrying the certification mark have been submitted to a testing process. The primary purpose of the conformance testing process is to protect the value of the OpenGIS® brand as an element of OGC’s program to promote interoperability between diverse geoprocessing systems.

This document defines the following:

- A general description of the conformance tests and the scope of the tests
- A description of the test data
- A description of the test programs
2 Test Description and Scope

According to the OpenGIS Simple Features Specification for OLE/COM, Revision 1.1 specification (hereafter referred to as “the specification”), there are three components:

- **OLE DB** for implementing data providers
- **ADO** for presenting a simplified data access model on top of OLE DB
- **Geometry and Spatial References** for detailed geometry and reference operations

The specification reads:

> Each of these components is implemented using the Microsoft Component Object Model (COM). ADO and the Geometry and Spatial Reference objects implement IDispatch and are therefore accessible to rapid development languages such as Visual Basic, Java, and Power Builder. All COM interfaces are accessible from lower level languages such as C++ for optimal performance.

And:

> ADO and OLEDB are the facilities by which consumers and providers communicate data and metadata. To achieve a sense of seamless interoperability, standards must be defined to allow GIS consumers and providers to communicate GIS information using these facilities. These standards enable an OLEDB data consumer to determine the GIS capabilities of an OLEDB data provider and retrieve GIS information in a predictable manner.

An OLEDB data consumer that cares nothing about GIS may still utilize data from an OGIS data provider as if it were any other OLEDB data provider. This consumer will simply not take advantage of the GIS information. Similarly, an OGIS data consumer can use the same OLEDB API, without the GIS standards, to utilize data from any OLEDB data provider.

These standards involve:

- **OGIS Data Provider Registry Entries**—OGIS data providers must register support for the “OGISDataProvider” component category so that consumers can distinguish them from other OLEDB Data Providers.
- **GIS Metadata**—Required and Optional GIS Metadata and the manner by which the client retrieves it are described. To a great extent, this is the bulk of the proposal—this is how the client knows there is GIS information and how to get it.

Test suite software is provided to verify conformance of:

1. OLE DB Providers, for accessing features in a database independent manner
2. COM interfaces, for handling geometry and spatial references.

The test suite software is provided as two Visual Basic programs, **SFProviderTest.exe** and **SFTest.exe**. The source code for both programs is included, as it is necessary to modify the code for implementation-specific calls, and to run the programs in VB runtime mode to identify certain failures to conform.

The test scripts check that all mandatory features have been implemented, and can be called, but they do not check that the returned values are correct.

The two test scripts are written in Microsoft Visual Basic. VB6 was chosen as the test script language, since it has good compatibility with COM, and is widely available.
Unfortunately, the test scripts cannot be completely implementation independent. Small modifications must be made for a specific implementation. For example, the VB programs must be referenced to the implementers Simple Features component(s).

2.1 OLE DB Provider

The specification states:

There is no strict “level of compliance” scheme for OGIS data servers. Clients are expected to use standard COM and OLE DB techniques for determining if a data server supports a specified type of functionality. It is the client’s responsibility to decide what action it will take if a data provider fails to support a given service.

The minimum level of support that a data server must pass is:

- Support the minimum set of interfaces as defined by the OLE DB standard
- Require providers to register support for the “OGISDataProvider” component category.
- Geometry values must be exposed as Well Know Binaries (WKB) as described in Section 4.

Conformance with the specification for an OLE DB Provider is tested by the program SFProviderTest.exe. The following steps describe the process of running the test:

Install Candidate Implementation

Installation of the candidate implementation will be specific to that implementation. No guidelines are given on the process for installing an OLE DB Provider implementation.

Register at Least One Feature Table

The process of registering feature tables is implementation specific, and no guidelines are given on this process.

Run Microsoft Visual Basic 6

Source code for the OLE DB Provider test is supplied so that organizations submitting candidate software can examine the tests being performed. It is not necessary to run the test in Visual Basic “run time” mode, as no modifications to the code are required, but it may prove useful in identifying areas of non-conformance. The test program may be run as an executable program without Visual Basic if desired.

Run the VB Program

The test program may be run from within Visual Basic or as a compiled executable. Submitting organizations may choose to run the test from within Visual Basic as a debugging aid. Whether the program is started from within Visual Basic, or as an executable program, the dialog shown below will be displayed, enabling the testing personnel to select the registered OLE DB Provider.
The Form_Load event attempts to make a connection using the Data Link class of the Microsoft OLE DB Service Component 1.0 Type Library. The test program reports success or failure:
Choose the Tests You Wish to Perform

The test program verifies mandatory Registry entries, and optionally three schema tests. Additionally, WKB Geometry is verified using the optional schemas and OLE DB schemas. Testing personnel can use the check boxes to select the components of the candidate implementation to be tested.

Press the “Run Tests” Button

When the test program is run, the selected tests are performed, and the dialog displays a message indicating success or failure for each test.

2.2 COM Interfaces

The specification states:

ADO is a Microsoft implementation of an OLEDB data consumer. While it is not an OLEDB data provider, its purpose is to provide the essence of OLEDB data to automation clients via a standard automation server. This allows easy access to data from high level languages such as Visual Basic and Power Builder. The ADO model is analogous to, but much simpler than the OLEDB model. The fundamental objects in the ADO object hierarchy are:

- Connection—This bundles the DSO and the Session and allows the automation client to connect to the data source, access schema information and begin posing queries.
• Command—This exposes the OLEDB Command capabilities, allowing queries to be posed (including spatial queries for OGIS data servers).

• Recordset—This is the result of a query or a request for information.

• Field—Column values that one would access via IRowset::GetData in OLEDB are accessed as the Value property of Field objects in the Recordset’s Fields collection.

Conformance with the specification for COM Interfaces is tested by the program SFTest.exe. The following steps describe the process of running the test:

Install Candidate Implementation
Installation of the candidate implementation will be specific to that implementation. No guidelines are given on the process for installing a COM Interface implementation.

Run Microsoft Visual Basic 6
Source code for the COM Interface test program is supplied so that organizations submitting candidate software can examine the tests being performed. It is also necessary to modify the Form_Load event of the program to cater for implementation specific processes. After modification, the test program may be run as an executable program without Visual Basic if desired.

Load the Visual Basic Project
Open the Visual Basic project SFTest.vbp and display the Form_Load event procedure.

Link to the Candidate Implementation Type Library.
Use the Project References command to link to the candidate implementation type library.

Edit the Form_Load Event
It is necessary to modify the code for this event to use the candidate’s implementation of OpenGIS Simple Features library classes. The unmodified code for this event includes the following lines:

```vba
' Get factories, using implementation specific code.
Set factorySR = New CadcorpSF.SpatialReferenceFactory
Set factoryG = New CadcorpSF.GeometryFactory
Set factorySRA = New CadcorpSF.SpatialReferenceAuthorityFactory
```

The candidate implementation of a COM Interface should contain classes named SpatialReferenceFactory, GeometryFactory, and SpatialReferenceAuthorityFactory. These classes should be prefixed with the name of the candidate implementation’s OpenGIS Simple Features Library.

The test program includes a routine to draw the sample map described in Section 3 of this document. The map data is provided as sample.txt, and is in Well Known Text (WKT) format. To draw the map using the candidate implementation’s OpenGIS Simple Features Library, it will be necessary to modify the Click event of the command button Command2.

Run the VB Program.
The test program may be run from within Visual Basic or as a compiled executable. Submitting organizations may choose to run the test from within Visual Basic as a debugging aid. Whether the
program is started from within Visual Basic, or as an executable program, the dialog shown below will be displayed, enabling the testing personnel to select the methods and interfaces to be tested:

![Test implementation of OpenGIS COM Simple Features](image)

**Choose the Tests You Wish to Perform**

The test program verifies mandatory and optional interfaces and methods. Testing personnel can use the check boxes to select the components of the candidate implementation to be tested.

**Press the “Run Tests” Button**

When the test program is run, the selected tests are performed, and the dialog displays a message indicating success or failure.
3 Test Data

The data is a synthetic data set, developed, by hand, to exercise the functionality of the specification. It is a set of features that make up a map (see Figure 1) of a fictional location called Blue Lake. This section describes the test data in detail.

![Image of Test Data Concept (Joe’s Blue Lake Vicinity Map)](image)

3.1 Test Data Semantics

The semantics of this data set are as follows:

- A rectangle of the Earth is shown in UTM coordinates. Horizontal coordinates take meaning from POSC Horizontal Coordinate System #32214. Note 500,000 meters false Easting, and WGS74. Units are meters. (see http://www.petroconsultants.com/epsgweb/epsg.htm)
- Blue Lake (which has an island named Goose Island) is the prominent feature.
- There is a watercourse flowing from North to South. The portion from the top neatline to the lake is called Cam Stream. The portion from the lake to the bottom neatline has no name (Name value is “Null”)
- There is an area place named Ashton.
- There is a State Forest whose administrative area includes the lake and a portion of Ashton. Roads form the boundary of the State Forest. The “Green Forest” is the State Forest minus the lake.
- Route 5 extends across the map. It is two lanes where shown in black. It is four lanes where shown in Red.
• There is a major divided highway, Route 75, shown in a double black line, one line for each part of the divided highway. These two lines are seen as a multi-line.
• There is a bridge (Cam Bridge) where the road goes over Cam Stream, a point feature.
• Main Street shares some pavement with Route 5, and is always four lanes wide.
• There are two buildings along Main Street; each can be seen either as a point or as a rectangle footprint.
• There is a one-lane road forming part of the boundary of the State Forest, shown in brown.
• There are two fish ponds, which are seen as a collective, not as individuals; that is, they are a multi-polygon.

3.2 Test Data Points and Coordinates

Figure 2 depicts the points that are used to represent the map.

![Points in the Blue Lake data set.](image)

The following table gives these coordinates associated with each point.

<table>
<thead>
<tr>
<th>Point</th>
<th>Easting</th>
<th>Northing</th>
<th>Point</th>
<th>Easting</th>
<th>Northing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>48</td>
<td>26</td>
<td>52</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>38</td>
<td>48</td>
<td>27</td>
<td>52</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>62</td>
<td>48</td>
<td>28</td>
<td>50</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>72</td>
<td>48</td>
<td>29</td>
<td>52</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>84</td>
<td>48</td>
<td>30</td>
<td>62</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>84</td>
<td>42</td>
<td>31</td>
<td>66</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>84</td>
<td>30</td>
<td>32</td>
<td>66</td>
<td>32</td>
</tr>
</tbody>
</table>
The following gives entire test data in Well Known Text (WKT) format, as supplied in the file `sample.txt`:

```
"PROJCS['UTM_ZONE_14N', GEOGCS['World Geodetic System 72',DATUM['WGS_72', SPHEROID['WGS_1972', 6378135, 298.26]], PRIMEM['Greenwich', 0], UNIT['Meter', 1.0]], PROJECTION['Transverse_Mercator'], PARAMETER['False_Easting', 500000.0], PARAMETER['False_Northing', 0.0], PARAMETER['Central_Meridian', -99.0], PARAMETER['Scale_Factor', 0.9996], PARAMETER['Latitude_of_origin', 0.0], UNIT['Meter', 1.0]]"
17
101,"Blue Lake","POLYGON( (52 18, 66 23, 73 9, 48 6, 52 18), (59 18, 67 18, 67 13, 59 13, 59 18) )"
102,"Route 5","LINESTRING( 0 18, 10 21, 16 23, 28 26, 44 31 )"
103,"Route 5","LINESTRING( 44 31, 56 34, 70 38 )"
104,"Route 5","LINESTRING( 70 38, 72 48 )"
105,"Main Street","LINESTRING( 70 38, 84 42 )"
106,"Dirt Road by Green Forest","LINESTRING( 28 26, 28 0 )"
109,"Green Forest","MULTIPOLYGON( ((28 26, 28 0, 84 0, 84 42, 28 26), (52 18, 66 23, 73 9, 48 6, 52 18)), ((59 18, 67 18, 67 13, 59 13, 59 18)) )"
110,"Cam Bridge","POINT( 44 31 )"
```
111,"Cam Stream","LINESTRING(38 48, 44 41, 41 36, 44 31, 52 18 )"
112,"Cam Stream", "LINESTRING( 76 0, 78 4, 73 9 )"
113,"123 Main Street","GEOMETRYCOLLECTION( POINT( 52 30 ), POLYGON( ( 50 31, 54 31, 5429, 50 29, 50 31 ) ))"
114,"215 Main Street","GEOMETRYCOLLECTION( POINT( 64 33 ), POLYGON( ( 66 34, 62 34, 6232, 66 32, 66 34 ) ))"
115,"Neat Line", "POLYGON( ( 0 0, 0 48, 84 48, 84 0, 0 0 ) )"
117,"Ashton","POLYGON( ( 62 48, 84 48, 84 30, 56 30, 56 34, 62 48 ) )"
118,"Goose Island", "POLYGON( ( 67 13, 67 18, 59 18, 59 13, 67 13 ) )"
119,"Route 75", "MULTILINESTRING( (10 48, 10 21, 10 0), (16 0, 10 23, 16 48 ) )"
120,"Stock Pond", "MULTIPOLYGON( ( ( 24 44, 22 42, 24 40, 24 44 ), ( 26 44, 26 40, 28 42, 26 44 ) ) )"

[ indicates that the subsequent line of text is a continuation.]
4 Conformance Items

This section details the tests that are to be executed. Each test constitutes a Conformance Item in the terminology of the Conformance Testing Program document. The ID in the following tables is used to reference the specific Conformance Item:

4.1 Conformance Items for OLE DB Providers

The OpenGIS GUIDs are defined in OLEDBGIS.h supplied separately, and are initialized as constants within the test program.

<table>
<thead>
<tr>
<th>ID</th>
<th>Functionality Tested</th>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>Registry Settings (Mandatory)</td>
<td>The Registry should contain the following key: HKEY_LOCAL_MACHINE\Software\CLASSES\CLSID{ProviderCLSID}\Implemented Categories{CATID_OGISDataProvider} CATID_OGISDataProvider is defined as {A0690A28-FAF5-11D1-BAF5-080036DB0B03} in OLEDBGIS.h</td>
<td>If any level in reading the hierarchy of this key fails, a suitable message will be displayed. Otherwise a success message is displayed.</td>
</tr>
<tr>
<td>P1</td>
<td>Feature Tables Schema (Optional)</td>
<td>The column names and values of schema DBSCHEMA_OGIS_FEATURE_TABLES are interrogated</td>
<td>If the Feature Tables Schema cannot be opened, a message is displayed. Otherwise each field name and value is reported for visual comparison of conformance with the specification.</td>
</tr>
<tr>
<td>P2</td>
<td>Geometry Columns Schema (Optional)</td>
<td>The column names and values of schema DBSCHEMA_OGIS_GEOMETRY_COLUMNS are interrogated</td>
<td>If the Geometry Columns Schema cannot be opened, a message is displayed. Otherwise each field name and value is reported for visual comparison of conformance with the specification.</td>
</tr>
<tr>
<td>P3</td>
<td>Spatial Ref Systems Schema (Optional)</td>
<td>The column names and values of schema DBSCHEMA_OGIS_SPATIAL_REF_SYSTEMS are interrogated</td>
<td>If the Spatial Ref Systems Schema cannot be opened, a message is displayed. Otherwise each field name and value is reported for visual comparison of conformance with the specification.</td>
</tr>
</tbody>
</table>
An attempt is made to open schema DBSCHEMA_OGIS_FEATURE_TABLES to enumerate Feature Tables. If successful, an attempt is made to open schema DBSCHEMA_OGIS_GEOMETRY_COLUMNS to get Spatial Ref System ID. If successful, an attempt is made to open DBSCHEMA_OGIS_SPATIAL_REF_SYSTEMS to get Spatial Ref WKT. If successful, the structure of the tables, column names and spatial reference system is evaluated. If any of the schemas cannot be opened a message will be displayed. Otherwise the Geometry WKB is tested, and success or failure is reported. Optionally the table data may be displayed, e.g.:
GID: 15
OGIS_GEOMETRY: Polygon with 1 ring(s)
fid: 118
name: Goose Island

The same tests are performed as in test P4, but using all OLE DB schemas. The test is performed where the schema IDs are those specified in OLEDBGIS.h

If any of the schemas cannot be opened a message will be displayed. Otherwise the Geometry WKB is tested, and success or failure is reported. Optionally the table data may be displayed, e.g.:
GID: 15
OGIS_GEOMETRY: Polygon with 1 ring(s)
fid: 118
name: Goose Island

### 4.2 Conformance Items for COM Interfaces

<table>
<thead>
<tr>
<th>ID</th>
<th>Functionality Tested</th>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Geometry Interfaces (Mandatory)</td>
<td>Well Known Text (WKT) geometry co-classes are created using the candidate implementation’s GeometryFactory, and the mandatory interfaces for each geometry are tested, as below:</td>
<td>The number of mandatory interfaces passing the test is reported. A message is displayed listing any failures.</td>
</tr>
<tr>
<td></td>
<td>POINT</td>
<td>IPoint, IGeometry, IWks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LINESTRING</td>
<td>ILineString, ICurve, IGeometry, IWks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POLYGON</td>
<td>IPolygon, ISurface, IGeometry, IWks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MULTIPOINT</td>
<td>IGeometryCollection, IGeometry, IWks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MULTILINESTRING</td>
<td>IMultiCurve, IGeometryCollection, IGeometry, IWks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MULTIPOLYGON</td>
<td>IMultiSurface, IGeometryCollection, IGeometry, IWks</td>
<td></td>
</tr>
</tbody>
</table>
### C2 Geometry Interfaces (Optional)

The geometry co-classes listed in test C1 are created, and the following optional interfaces for each geometry are tested:
- ISpatialRelation
- ISpatialRelation2
- ISpatialOperator

A message is displayed of the number of optional interfaces found. Examination of the program in runtime mode will reveal which optional interfaces are not present in the candidate implementation.

### C3 Geometry Methods (Mandatory)

Mandatory geometry methods are tested, by calling all methods on co-classes. Return values are NOT checked, but the VB runtime program can catch errors due to method implementations failing. The statement:

```vbnet
On Error GoTo failed
```

may be commented out of the Test3 procedure to allow VB to stop on failure.

Well Known Text (WKT) geometry co-classes are created using the candidate implementation’s GeometryFactory, and the mandatory methods for each geometry are tested, as below:

<table>
<thead>
<tr>
<th>IPoint</th>
<th>Clone, Coords, Dimension, Envelope, Extent2D, IsEmpty, IsSimple, Project, SpatialReference, x, y, SetEmpty</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILineString</td>
<td>Clone, Dimension, EndPoint, Envelope, Extent2D, IsClosed, IsEmpty, IsSimple, Length, NumPoints, Point, Project, SpatialReference, StartPoint, Value, SetEmpty</td>
</tr>
<tr>
<td>IPolygon</td>
<td>Centroid, Clone, Dimension, Envelope, Extent2D, ExteriorRing, InteriorRing, IsEmpty, IsSimple, NumInteriorRings, PointOnSurface, Project, SpatialReference, SetEmpty</td>
</tr>
</tbody>
</table>

If all methods tested pass the test, a “Passed” message is displayed. If any method fails, this test is halted with a failure message. The method causing the failure may be discovered by running the test in VB runtime mode.
<table>
<thead>
<tr>
<th>Interface</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMultiCurve</td>
<td>Clone, Dimension, Envelope, Extent2D, Geometry, IsClosed, isEmpty, IsSimple, Length, NumGeometries, Project, SpatialReference, SetEmpty</td>
</tr>
<tr>
<td>IMultiSurface</td>
<td>Area, Centroid, Clone, Dimension, Envelope, Extent2D, Geometry, isEmpty, IsSimple, NumGeometries, PointOnSurface, Project, SpatialReference, SetEmpty</td>
</tr>
<tr>
<td>IGeometryCollection</td>
<td>Clone, Dimension, Envelope, Extent2D, Geometry, isEmpty, IsSimple, NumGeometries, Project, SpatialReference, SetEmpty</td>
</tr>
</tbody>
</table>

Additionally, IWks methods are tested on geometry co-classes. Point, Linestring, Polygon, MultiPoint, MultiLineString, MultiPolygon and GeometryCollection geometries are created using the candidate implementation’s GeometryFactory CreateFromWKT method. The IWks methods ExportToWKT, ExportToWKB, ImportFromWKT and ImportFromWKB are tested.

**C4 Geometry Methods (Optional)**

Point, Linestring, Polygon, MultiPoint, MultiLineString, MultiPolygon and GeometryCollection geometries are created using the candidate implementation’s GeometryFactory CreateFromWKT method. The following methods are tested:

- for ISpatialRelation:
  - Contains
  - Crosses
  - Disjoint
  - Equals
  - Intersects
  - Overlaps
  - Touches
  - Within
- for ISpatialRelation2:
  - Relate
- for ISpatialOperator:
  - Boundary
  - Buffer
  - ConvexHull
  - Distance
  - Difference

If all methods tested pass the test, a “Passed” message is displayed. If any method fails, this test is halted with a failure message. The method causing the failure may be discovered by running the test in VB runtime mode.
| C5 | Spatial Reference Interfaces | Intersection  
SymmetricDifference  
Union | A WktSpatialReference is created in the Form_Load event for use as the basis of this test. The following interfaces are tested and the SpatialReferenceInfo interface tested for each:  
AngularUnit / Unit  
Ellipsoid  
GeodeticSpatialReference  
GeographicCoordinateSystem  
HorizontalDatum  
LinearUnit / Unit  
ParameterInfo / Parameter  
ProjectedCoordinateSystem  
Projection | A total of 26 interfaces are tested. Failure of any test will result in an appropriate failure message. Success is indicated if all tests are passed. |
| C6 | Spatial Reference Methods | The following Spatial Reference methods are tested:  
AngularUnit  
Abbreviation, Alias, Authority, Code, Name, Remarks, WellKnownText, RadiansPerUnit  
Ellipsoid  
Abbreviation, Alias, Authority, Code, Name, Remarks, WellKnownText, AxisUnit, InverseFlattening, SemiMajorAxis, SemiMinorAxis  
GeodeticSpatialReference  
Abbreviation, Alias, Authority, Code, Name, Remarks, WellKnownText  
GeographicCoordinateSystem  
Abbreviation, Alias, Authority, Code, Name, Remarks, WellKnownText, AngularUnit, HorizontalDatum, PrimeMeridian  
HorizontalDatum  
Abbreviation, Alias, Authority, Code, Name, Remarks, WellKnownText, Ellipsoid  
LinearUnit  
Abbreviation, Alias, Authority, Code, Name, Remarks, WellKnownText, MetersPerUnit | A “Passed” or “Failed” message is displayed at the end of this test. The statement:  
On Error GoTo failed  
may be commented out of the Test6 procedure to allow VB to stop on failure to determine the method(s) causing failure. |
### Geographic Transform Interfaces

A GeographicTransform is created using the candidate implementation’s SpatialReferenceAuthorityFactory CreateGeographicTransform method (BursaWolf) in the Form_Load event. The GeographicTransform and SpatialReferenceInfo interfaces are tested.

A total of 2 interfaces are tested. Failure of any test will result in an appropriate failure message. Success is indicated if all tests are passed.

### Geographic Transform Methods

The following GeographicTransform methods are tested: Abbreviation, Alias, Authority, Code, Name, Remarks, WellKnownText, DefaultParameterInfo, ParameterInfo, Forward, Inverse, SourceGCS, TargetGCS

A “Passed” or “Failed” message is displayed at the end of this test. The statement: On Error GoTo failed may be commented out of the Test8 procedure to allow VB to stop on failure.
|   |   | to determine the method(s) causing failure. |