# Sensor Observation Service (SOS) TeamEngine™ Test Script Changes

This document describes changes that have been made to SOS test scripts to enable and enhance testing of NOAA Integrated Ocean Observing System (IOOS) Server Output. It is organized by the test script files that were modified, listed in SOS test suite execution order, and the functions and elements that were modified in those files. The change descriptions include comments on the problems they are intended to solve, the TeamEngine design features they work around, and the benefits they provide. Examine the test script files to see the changes that have been made. The changes are noted in XML comments like this: <!-- 2010-12-30 PwD -->.

## SOS\_ETS.xml

This file contains the compliance test suite for Sensor Observation Service (SOS) version 1.0. It structures and controls execution of the test suite.

### <ctl:test name=*"sos:Main"*>

The <ctl:form> element contents were modified to change the style and present the test user with options to select any or all tests for execution in a test session. The purpose of this change was to allow test of individual SOS capabilities, to avoid wasting time re-testing capabilities that consistently pass test, and to allow testing to focus on test failures. The modified HTML form presented to the user is shown in Figure 1 below.

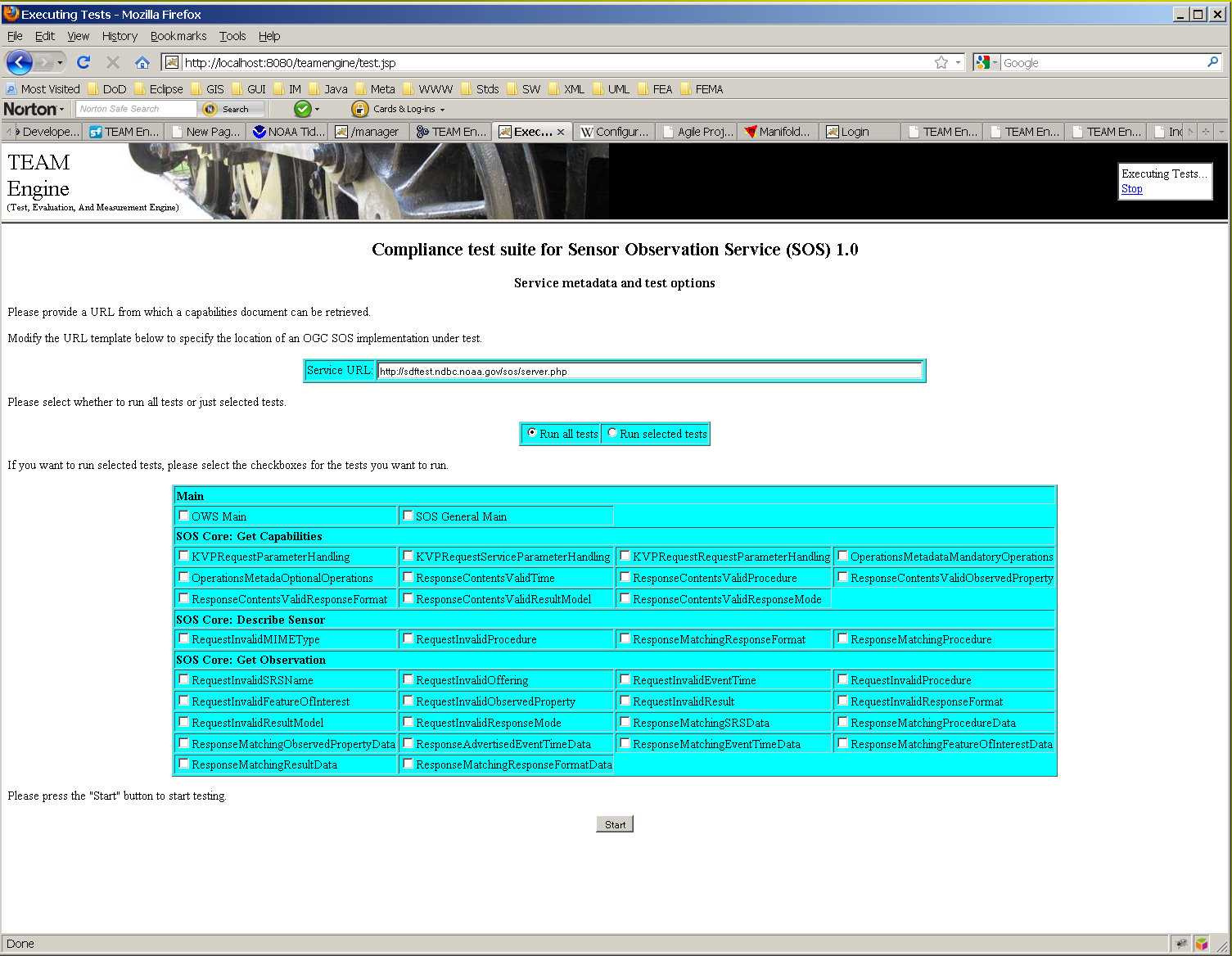


Figure SOS Test Suite Run Options HTML Page

The radio buttons and check boxes on this form set values which are used to set new XSLT variables in the test script. These variables are checked within this test element to control main test component execution, and passed to the <ctl:test name=*"sos:core-main"*> test in another package as parameters that are checked to control execution of SOS core tests. The Boolean values of these parameters could be tested as Booleans in the test where they were created, but had to be tested as strings, e.g. $var = ‘true’, after being passed (as strings) to the sos:core-main test. The values of these variables are posted to the TeamEngine consol display and log files using <ctl:message> elements.

### <ctl:test name="sos:general-SOS.General-InvalidRequest.1">

This test checked for ExceptionReports with an exceptionCode value of InvalidRequest. However, InvalidRequest is not listed in SOS or in OWS Common v 1.1 or 2.0 as a valid exception code. So this test was changed to call the function sosFunctions:checkExceptionReportwith an exceptionReport parameter value of MissingParameterValue instead of InvalidRequest. The test assertion and error messages were changed accordingly.

### <ctl:test name=*"sos:core-main"*>

This test was modified to use additional test run option parameters set by sos:Main as described above to conditionally run subsidiary tests defined in other files. It was modified to call the new function sosFunctions:capabilitiesOfferingName to get the value of a new offeringName parameter used in calls to DescribeSensor tests. It was modified to call the new function sosFunctions:observationOfferingObservedPropertyAsTextXML instead of the existing function sosFunctions:observationOfferingObservedProperty to set the value of the observationOfferingObservedProperty parameter passed to GetObservation tests. It was modified to use these values in calls to the following tests:

getObservation:core-SOS.GetObservation-RequestInvalidSRSName.1 getObservation:core-SOS.GetObservation-RequestInvalidEventTime.1 getObservation:core-SOS.GetObservation-RequestInvalidProcedure.1 getObservation:core-SOS.GetObservation-RequestInvalidFeatureOfInterest.1 getObservation:core-SOS.GetObservation-RequestInvalidObservedProperty.1 getObservation:core-SOS.GetObservation-RequestInvalidResult.1 getObservation:core-SOS.GetObservation-RequestInvalidResponseFormat.1 getObservation:core-SOS.GetObservation-RequestInvalidResultModel.1 getObservation:core-SOS.GetObservation-RequestInvalidResponseMode.1 getObservation:core-SOS.GetObservation-ResponseMatchingResultData.1

This test was also modified to detect and provide messages about ExceptionReport responses returned by the modified function sosFunctions:xmlValidatingParser described below under functions.xsd. See the descriptions of the new functions under functions.xsd below.

## GetCapabilities.xml

This file contains GetCapabilities tests for the SOS compliance test suite.

### <ctl:test name= "getCapabilities:core-SOS.GetCapabilities-ResponseContentsValidObservedProperty.1">

This test was modified to allow the observed property to be specified as a URL as well as a URN by calling sosFunctions:isValidURI instead of sosFunctions:isValidDefinitionURN. See the function description under functions.xml below.

## DescribeSensor.xml

This file contains DescribeSensor tests for the SOS compliance test suite.

### <ctl:test name="describeSensor:core-SOS.DescribeSensor-ResponseMatchingResponseFormat.1">

This test was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function description under functions.xml below.

### <ctl:test name="describeSensor:core-SOS.DescribeSensor-ResponseMatchingProcedure.1">

This test was using the optional sml:Term/@definition attribute value as the term identifier in the server response. It was was modified to call *sosFunctions*:getTermIdentifier to find the procedure ID in the response.

This test was also modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

## GetObservation.xml

This file contains GetObservation tests for the SOS compliance test suite.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidSRSName.1">

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidEventTime.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidProcedure.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidFeatureOfInterest.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidObservedProperty.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidResult.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidResponseFormat.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidResultModel.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-RequestInvalidResponseMode.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-ResponseMatchingResultData.1>

This test was changed to have an offeringName parameter instead of an offeringId parameter, and to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. See the description of <ctl:function name="sosFunctions:capabilitiesOfferingName"> under functions.xml below, which is called by SOS\_ETS.xml to set the value of this parameter when this test is called.

### <ctl:test name="getObservation:core-SOS.GetObservation-ResponseMatchingSRSData.1">

This test was changed to skip observations not offered in a text/xml response format. It was changed to call sosFunctions:capabilitiesOfferingName to get an offeringName value to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. It was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

### <ctl:test name="getObservation:core-SOS.GetObservation-ResponseMatchingProcedureData.1">

This test was changed to skip observations not offered in a text/xml response format. It was changed to call sosFunctions:capabilitiesOfferingName to get an offeringName value to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. It was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

### <ctl:test name="getObservation:core-SOS.GetObservation-ResponseMatchingObservedPropertyData.1">

This test was changed to skip observations not offered in a text/xml response format. It was changed to call sosFunctions:capabilitiesOfferingName to get an offeringName value to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. It was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

### <ctl:test name="getObservation:core-SOS.GetObservation-ResponseMatchingEventTimeData.1">

This test was changed to call sosFunctions:capabilitiesOfferingName to get an offeringName value to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. It was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

### <ctl:test name="getObservation:core-SOS.GetObservation-ResponseMatchingFeatureOfInterestData.1">

This test was changed to call sosFunctions:capabilitiesOfferingName to get an offeringName value to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. It was changed to call sosFunctions:observationOfferingObservedPropertyAsTextXMLto set the <observedProperty> parameter for the GetObservation request it makes to an SOS server. It was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

### <ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingResponseFormatData.1"*>

This test was changed to call sosFunctions:capabilitiesOfferingName to get an offeringName value to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. It was changed to call sosFunctions:observationOfferingObservedPropertyAsTextXMLto set the <observedProperty> parameter for the GetObservation request it makes to an SOS server. It was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

### <ctl:test name="getObservation:core-SOS.GetObservation-ResponseMatchingResultModel.1">

This test was changed to skip observations not offered in a text/xml response format. It was changed to call sosFunctions:capabilitiesOfferingName to get an offeringName value to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. It was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

### <ctl:test name="getObservation:core-SOS.GetObservation-ResponseMatchingResponseMode.1">

This test was changed to skip observations not offered in a text/xml response format. It was changed to call sosFunctions:capabilitiesOfferingName to get an offeringName value to use its value to set the <offering> parameter for the GetObservation request it makes to an SOS server. It was modified to check for and provide messages in response to an Exception report returned from sosFunctions:xmlValidatingParser. See the function descriptions under functions.xml below.

## functions.xml

This file contains definitions of XPath functions that standardize sets of XSLT operations that are commonly used in the SOS test suite. Two namespaces were added to the <ctl:package> document element to support the sosFunctions:getTermIdentifier function which was added (see below).

### <ctl:function name="sosFunctions:isValidURN">

This function only found ogc:: URNs to be valid, but NOAA has established its own ioos:: URN scheme. So the test was modified to also allow IOOS URNs.

### <ctl:function name="sosFunctions:isValidURI">

This function was added to validate URLs as well as URNs as values used to specify observed properties. It is used by <ctl:test name=*"getCapabilities:core-SOS.GetCapabilities-ResponseContentsValidObservedProperty.1"*> in GetCapabilities.xml (see above).

### <ctl:function name="sosFunctions:isValidMIMETypeFormat">

This function only found MIME types with a subtype=value parameter to be valid, according to OGC’s incorrect use and requirement of the word 'subtype' in a MIME type such as text/xml; subtype="om/0.0.0" in the SOS specification. The structure of a MIME identifier is type/subtype, so here 'text' is the type and 'xml' is the subtype. Anything after the ';' is a parameter.  See, e.g., RFC 2045 Section 5.1: <http://tools.ietf.org/html/rfc2045#section-5.1> So this function was modified to find the following IOOS MIME types to be valid:

text/xml;schema="ioos/0.6.1"

application/ioos+xml;version=0.6.1

text/csv

text/tab-separated-values

application/vnd.google-earth.kml+xml

### <ctl:function name="sosFunctions:observationOfferingObservedPropertyAsTextXML">

The purpose of this function is to enable testing of Observations in text/xml format, which is required by SOS Observation tests so that an XML validating parser can be used to validate the Observation responses. It is a work-around for the fact that <ctl:function name=*"sosFunctions:observationOfferingObservedProperty"*> called by SOS observation tests to select an observedProperty to test, simply returns the first one. Unfortunately, the first NOAA IOOS observedProperty is air\_temperature, which is not provided in a text/XML format. Also unfortunately, there is no practicable way to express in a given SOS Observation Offering that not all Formats are available for all Observed Properties. Consequently, three IOOS Observation tests would fail simply because air\_temperature had been selected for test. So this function returns the first observedProperty if there is only one, or the first that is available in text/xml format, if any, or the first observation offering if not. It has to issue a request to the SOS server, which produces the undesirable side effects of slowing down the testing and creating larger log files. So better Observation test coverage has a cost. This function is called by the following tests in GetObservation.xml:

<ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingProcedureData.1"*> <ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingFeatureOfInterestData.1"*> <ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingResponseFormatData.1"*>

### <ctl:function name="sosFunctions:capabilitiesOfferingName">

The orginal Observation tests in GetObservation.xml (see above) were incorrectly using the observation offering gml:id XML ID value instead of the gml:name URN value to request observations. Consequently, the observation tests were failing. This function returns the gml:name URN value to use to request observations. It is called by the following tests in GetObservation.xml:

<ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingSRSData.1"*> <ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingProcedureData.1"*> <ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingObservedPropertyData.1"*> <ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseAdvertisedEventTimeData.1"*> <ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingEventTimeData.1"*> <ctl:test name=*"getObservation:core-SOS.GetObservation-ResponseMatchingFeatureOfInterestData.1"*>

and the <ctl:test name=*"sos:core-main"*> test in SOS\_ETS.xml, which passes the return value as a parameter to all the other getObservation tests.

### <ctl:function name="sosFunctions:getTermIdentifier">

Originally, <ctl:test name=*"describeSensor:core-SOS.DescribeSensor-ResponseMatchingProcedure.1"*> was using the optional sml:Term/@definition attribute value as the term identifier. Making an optional schema element a requirement for passing a conformance test is a serious design error. Not all NOAA stations returned the optional definition attribute; the term identifier is often returned in a child sml:value element instead. This function returns the term identifier provided in either location.

### <ctl:function name="sosFunctions:xmlValidatingParser">

In the TeamEngine, whenever a parser is used to validate a server response, the actual request to the server is made from within the parser framework, and the response is not returned to the calling test script, which only receives any parser output, or null if no validation errors were found in the response. This design makes it impossible for a test script writer to choose which schema to apply to validate a response based on the response contents, without making a separate initial (duplicate) request to the server that is not linked to a parser. In IOOS testing, appropriate ExceptionReport results were being hidden from the console output, instead being reported as validation errors. To work around this problem, this function was modified to use the owsExceptionReport.xsd schema in addition to whatever schema is passed in as a parameter, so that a valid ExceptionReport response will not cause schema validation errors. This change required changes to every test that uses this function with a schema other than owsExceptionReport.xsd, to test the return value explicitly for an Exception report as well as schema validation errors, and to return a meaningful error message to the tester. This function is called for all tests.

### <ctl:function name="sosFunctions:mimeSubtype">

This function simply returned the value of the subtype=value parameter, according to OGC’s incorrect use and requirement of the word 'subtype' in a MIME type such as text/xml; subtype="om/0.0.0" in the SOS specification. The structure of a MIME identifier is type/subtype, so here 'text' is the type and 'xml' is the subtype. Anything after the ';' is a parameter.  See, e.g., RFC 2045 Section 5.1: <http://tools.ietf.org/html/rfc2045#section-5.1> So this function was modified to return “om/1.0.0” for MIME types without a (sic) subtype parameter, instead of an empty string.