APPENDIX D

CRITERIA/TEST SCENARIO CROSS-REFERENCE

D-1 GENERAL. This appendix provides a matrix for the mapping between the candidate subtest criteria and the candidate scenarios used to determine to what extent a client or server implements the IAS specification. To reduce the resources required to test each criteria independently, test scenarios are developed to measure test criteria for several subtests at the same time. Each subtest will have a table showing the mapping between candidate subtest criteria from Appendix C and the candidate test scenarios from Appendix E.

Table D-1. Subtest 1, Server Disseminate()

SUBTEST 1, SERVER DISSEMINATE()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality.															
a. Disseminate product to one or more destinations	X	X													
b. Disseminate product with response service:															
(1) IMMEDIATE	X														
(2) QUEUED	X														
c. A valid list of request IDs is returned	X														
2. Method Exceptions:															
a. BadProductReference	X														
b. BadLocationSpec	X														
c. ResponseServiceNotAvailable	X														
d. TooManyRequests															
e. BAD_INV_ORDER															
3. Parameters Interface:															
a. Disseminate with ProductCompression:															
(1) “UNC”, “NITF20” only	X	X													
(2) “JPEG8”, “NITF20” only	X														
(3) “JPEG12”, “NITF20”	X														

(4) “TFRD43”, TFRD only	X														
b. Disseminate with ProductDataFormat:															
(1) “NITF20”	X														
(2) “TFRD”	X														
c. GeographicDatum not used.															
d. Disseminate with ProductRRDS set to requested	X														

Table D-2. Subtest 2, Client Disseminate()

SUBTEST 2, CLIENT DISSEMINATE()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Request dissemination of product to one/more destinations	X	X													
b. Request dessemination with response service:															
(1) IMMEDIATE	X														
(2) QUEUED	X														
c. Returned request IDs used to track progress	X														
2. Method Exceptions:															
a. BadProductReference	X														
b. BadLocationSpec	X														
c. ResponseServiceNotAvailable	X														
d. TooManyRequests															
e. BAD_INV_ORDER															
3. Parameters Interface:															
a. Request dissemination with ProductCompression set to:															
(1) “UNC, “NITF20” only	X														
(2) “JPEG8, “NITF20” only	X														
(3) “JPEG12”, “NITF20” only	X														

(4) “TFRD43”, “TFRD” only	X														
b. Disseminate with ProductDataFormat:															
(1) “NITF20”	X														
(2) “TFRD”	X														
c. GeographicDatum” not used.															
d. Disseminate with ProductRRDS set to resolutions	X														

Table D-3. Subtest 3, Server Check_Completion()

SUBTEST 3, SERVER CHECK_COMPLETION()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Status of disseminate request is returned to user	X														
b. Status of create request is returned to the user		X									X				
c. Status of get_subimage request is returned to user						X									
d. Status COMPLETED returned when transfer is complete	X	X				X					X				
e. Status IN_PROGRESS returned when transfer is ongoing	X	X													
f. Status ABORTED returned when transfer is aborted															
g. Status CANCELED returned when transfer is canceled	X	X													
h. Status PENDING returned when transfer is awaiting service	X														
i. Status OTHER returned when ((TBD))															
j. Reason limited to not more than 80 characters.	X	X				X					X				
2. Method Exceptions:															
a. BadRequestId	X														
b. BAD_INV_ORDER															
3. Parameters Interface: None.															

Table D-4. Subtest 4, Client Check_Completion()

SUBTEST 4, CLIENT CHECK_COMPLETION()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Use previous disseminate RequestId to determine status	X														
b. Use previous create call to determine status		X									X				
c. Use previous get_subimage call to determine status						X									
d. COMPLETED status displayed when transfer completed	X	X				X					X				
e. IN_PROGRESS status displayed when transfer is ongoing	X	X													
f. ABORTED status displayed when transfer is aborted															
g. CANCELED status displayed when transfer is canceled	X	X													
h. PENDING status displayed when transfer is awaiting service	X														
i. OTHER status displayed when ((TBD))															
j. Reason limited to not more than 80 characters.	X	X				X					X				
2. Method Exceptions:															
a. BadRequestId	X														
b. BAD_INV_ORDER															
3. Parameters Interface: None.															

Table D-5. Subtest 5, Server Cancel()

SUBTEST 5, SERVER CANCEL()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Dissemination request is terminated by client	X														
b. Creation request is terminated by client		X													
c. Get_subimage request is terminated by the client						X									
d. Status COMPLETED returned when complete	X														
e. Status ABORTED returned when transfer is aborted															
2. Method Exceptions:															
a. BadRequestId	X														
b. BAD_INV_ORDER															
3. Parameters Interface: None.															

Table D-6. Subtest 6, Client Cancel()

SUBTEST 6, CLIENT CANCEL()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Use requestID from previous dissemination to cancel terminated by client	X														
b. Use request ID from previous create to cancel		X													
c. Use request ID from previous get_subimage to cancel						X									
d. COMPLETED status displayed when complete	X														
e. ABORTED status displayed when transfer is aborted															
2. Method Exceptions:															
a. BadRequestId	X														
b. BAD_INV_ORDER															
3. Parameters Interface: None.															

Table D-7. Subtest 7, Server Create()

SUBTEST 7, SERVER CREATE()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Image product's LocationSpec is created in server's library		X		X							X	X			
b. Creation_attribute metadata is saved in the library		X													
c. A valid request id is returned		X													
d. A valid product reference is returned		X													
2. Method Exceptions:															
a. BadCreationAttributes		X													
b. BadLocationSpec		X													
b. BAD_INV_ORDER				X											
3. Parameters Interface: None.															

Table D-8. Subtest 8, Client Create()

SUBTEST 8, CLIENT CREATE()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Request creation of product in library of image file		X		X							X	X			
b. Creation_attribute metadata specified for files w/o metadata		X													
c. Returned request id used to check complete or cancel		X													
d. Returned product reference used with other methods		X													
2. Method Exceptions:															
a. BadCreationAttributes		X													
b. BadLocationSpec		X													
b. BAD_INV_ORDER				X											
3. Parameters Interface: None.															

Table D-9. Subtest 9, Server Open_Array()

SUBTEST 9, SERVER OPEN_ARRAY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Open_array is called before others in array request interface			X												
b. Parameter access_kind is unused			X												
2. Method Exceptions:															
a. BadProductReference			X												
b. ProductUnavailable			X												
c. BAD_INV_ORDER															
3. Parameters Interface: None.															

Table D-10. Subtest 10, Client Open_Array()

SUBTEST 10, CLIENT OPEN_ARRAY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Open_array is called before others in array request interface			X												
b. Parameter access_kind is N/A			X												
2. Method Exceptions:															
a. BadProductReference			X												
b. ProductUnavailable			X												
c. BAD_INV_ORDER															
3. Parameters Interface: None.															

Table D-11. Subtest 11, Server Close_Array()

SUBTEST 11, SERVER CLOSE_ARRAY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Close_array is called after others in array request interface			X												
2. Method Exceptions:															
a. BadProductReference			X												
b. ArrayNotOpen			X												
c. BAD_INV_ORDER															
3. Parameters Interface: None.															

Table D-12. Subtest 12, Client Close_Array()

SUBTEST 12, CLIENT CLOSE_ARRAY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Close_array is called after others in array request interface			X												
2. Method Exceptions:															
a. BadProductReference			X												
b. ArrayNotOpen			X												
c. BAD_INV_ORDER															
3. Parameters Interface: None.															

Table D-13. Subtest 13, Server Get_Region()

SUBTEST 13, SERVER GET_REGION()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provides pixel data from array product based on region_spec			X												
b. The RegionData returns region_spec and region_data			X												
c. The region_spec parameter is specialized with DisplayRegion Data type															
d. The region_header NameValueList contain the entries :															
(1) "NCOLS"			X												
(2) "NROWS"			X												
(3) "PixelDepth"			X												
2. Method Exceptions:															
a. BadProductReference			X												
b. BadRegionSpec			X												
c. BadRegionData (when Compression Not Available)			X												
d. BadRegionData (when Format Not Available)			X												
e. ArrayNotOpen			X												
f. BAD_INV_ORDER															
3. Parameters Interface:															
a. RegionParameters structure contains size and resolution data			X												

b. ProductCompression, not used. Data is uncompressed.			X												
c. ProductDataFormat, not used.															
d. Region data optimized by HorzMotionRate & VertMotionRate															
e. SupportData, not used.															

Table D-14. Subtest 14, Client Get_Region()

SUBTEST 14, CLIENT GET_REGION()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provides x-y center coordinates in DisplayRegionSpec			X												
b. The RegionData returned is region_spec and region_data			X												
c. The region-header NameValueList contains the entries:															
(1) "NCOLS"			X												
(2) "NROWS"			X												
(3) "PixelDepth"			X												
2. Method Exceptions:															
a. BadProductReference			X												
b. BadRegionSpec			X												
c. BadRegionData (when Compression Not Available)			X												
d. BadRegionData (when Format Not Available)			X												
e. ArrayNotOpen			X												
f. BAD_INV_ORDER															
3. Parameters Interface:															
a. Request size and resolution data in RegionParameters			X												
b. ProductCompression, not used. Data is uncompressed.			X												
c. ProductDataFormat, not used.															

d. Provide indication of future region requests in HorizMotionRate and VertMotionRate															
e. SupportData, not used.															

Table D-15. Subtest 15, Server Get_Mutiple_Regions()

SUBTEST 15, SERVER GET_MULTIPLE_REGIONS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provides multi-region pixel data based on region_spec list			X												
b.The list of region data returns region_spec and region data			X												
c. The region_spec is specialized with TileRegionSpec data															
d. The region_header NameValueList contains the entries:															
(1) "NCOLS"			X												
(2) "NROWS"			X												
(3) "PixelDepth"			X												
2. Method Exceptions:															
a. BadProductReference			X												
b. BadRegionSpec			X												
c. BadRegionData (when Compression Not Available)			X												
d. BadRegionData (when Format Not Available)			X												
e. ArrayNotOpen			X												
f. BAD_INV_ORDER															
3. Parameters Interface:															
a. ProductCompression, not used. Data is uncompressed.			X												
b. ProductDataFormat, not used.															

c. Region Data optimized by HorzMotionRate & VertMotionRate															
d. SupportData, not used.															

Table D-16. Subtest 16, Client Get_Mutiple_Regions()

SUBTEST 16, CLIENT GET_MULTIPLE_REGIONS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provide x-y center, size, and resolution in TileRegionSpec			X												
b. The list of region data returns region_spec and region data			X												
c. The region_header NameValueList contains the entries:															
(1) "NCOLS"			X												
(2) "NROWS"			X												
(3) "PixelDepth"			X												
2. Method Exceptions:															
a. BadProductReference			X												
b. BadRegionSpec			X												
c. BadRegionData			X												
d. ArrayNotOpen			X												
e. BAD_INV_ORDER															
3. Parameters Interface:															
a. ProductCompression, not used. Data is uncompressed.			X												
b. ProductDataFormat, not used.															
d. Provide indication of future region requests in HorizMotionRate and VertMotionRate															
d. SupportData, not used.															

Table D-17. Subtest 17, Server Open()

SUBTEST 17, SERVER OPEN()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Connection opened with valid user identity and password				X			X	X	X	X	X				
b. No operations performed prior to invocation of open method				X							X				
2. Method Exceptions:															
a. AlreadyConnected				X											
b. BadOpenCriteria				X											
c. NO_PERMISSION				X											
3. Parameters Interface: None															

Table D-18. Subtest 18, Client Open()

SUBTEST 18, CLIENT OPEN()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provides valid open_criteria with user identity and password				X			X	X	X	X	X				
b. No operations performed prior to invocation of open method				X							X				
2. Method Exceptions:															
a. AlreadyConnected				X											
b. BadOpenCriteria				X											
c. NO_PERMISSION				X											
3. Parameters Interface: None															

Table D-19. Subtest 19, Server Close()

SUBTEST 19, SERVER CLOSE()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Closes the connection				X			X	X	X	X	X				
b. No operations performed after invocation of close method				X											
2. Method Exceptions:															
a. NoConnectionEstablished				X											
3. Parameters Interface: None															

Table D-20. Subtest 20, Client Close()

SUBTEST 20, CLIENT CLOSE()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Requests to close connection				X			X	X	X	X	X				
b. No operations performed after invocation of close method				X											
2. Method Exceptions:															
a. NoConnectionEstablished				X											
3. Parameters Interface: None															

Table D-21. Subtest 21, Server Get_Parameters()

SUBTEST 21, SERVER GET_PARAMETERS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. List of requested parameters & associated values is provided			X		X										
b. ImageProduct uses username/password in ContextInfo.					X										
2. Method Exceptions:															
a. BadName					X										
b. BAD_INV_ORDER															
3. Parameters Interface:															
a. ResultAttributes lists attributes returned by catalog queries					X										
b. Provide GeographicDatum as WGS84					X										
c. Provide requested reduced resolution sets in ProductRRDS					X										
d. Provide horizontal/vertical sizes & resolution in RegionParameters					X										
e. Provide one of the following in ProductCompression:															
(1) “UNC”, “NITF20” only					X										
(2) “JPEG8”, “NITF20” only					X										
(3) “JPEG12”, “NITF20” only					X										
(4) “TFRD43”, “TFRD20” only					X										
f. Provide one of the following in ProductDataFormat:															

(1) “NITF20”					X										
(2) “TFRD”					X										
g. Provide horizontal direction in HorizMotionRate					X										
h. Provide vertical direction in VertMotionRate					X										
i. SupportData, not used.															
j. InterfacesSupported, not used.															
k. Provide available catalog attributes in AvailableAttributes					X										
l. Provide queryable attributes in QueryableAttributes					X										
m. Provide max polygon vertices in MaxPolygonVertices					X										
n. Provide following key attributes in ReferenceAttributes:															
(1) “NROWS”					X										
(2) “NCOLS”					X										
o. ArrayProduct true for an array image			X		X										
p. Provide library location in LibraryReference					X										
q. Provide list of available parameters in ParameterNames					X										
r. Provide current setting of BrowseImageReturned					X										

Table D-22. Subtest 22, Client Get_Parameters()

SUBTEST 22, CLIENT GET_PARAMETERS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Requests a list of parameters			X		X										
b. The parameters names in the set:															
(1) ResultAttributes					X										
(2) BrowserImageReturned					X										
(3) GeographicDatum					X										
(4) ProductRRDS					X										
(5) RegionParameters					X										
(6) ProductCompression					X										
(7) ProductFormat					X										
(8) InterfacesSupported, not used.															
(9) AvailableAttributes					X										
(10) QueryableAttributes					X										
(11) MaxPolygonVertices					X										
(12) ReferenceAttributes					X										
(13) ArrayProduct					X										
(14) LibraryReference					X										
(15) ParameterNames					X										

(16) SupportData, not used.															
c. ImageProduct uses username/password in ContextInfo.					X										
2. Method Exceptions:															
a. BadName					X										
b. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-23. Subtest 23, Server Set_Parameters()

SUBTEST 23, SERVER SET_PARAMETERS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Allow changes to client-specific parameters only	X		X		X	X	X	X	X	X		X	X		
b. ImageProduct uses username/password in ContextInfo.					X										
2. Method Exceptions:															
a. BadName					X										
b. BadValue					X										
c. CannotSet					X										
d. BAD_INV_ORDER															
3. Parameters Interface:															
a. ResultAttributes lists attributes returned by catalog queries	X		X		X		X	X	X	X		X	X		
b. Provide GeographicDatum as “WGS84”	X				X	X		X	X	X		X			
c. Provide requested reduced resolution sets in ProductRRDS					X	X									
d. Provide horizontal/vertical sizes & resolution in RegionParameters			X		X										
e. Provide one of the following in ProductCompression:															
(1) “UNC”, “NITF20” only	X		X		X	X									
(2) “JPEG8”, “NITF20” only	X		X		X	X									
(3) “JPEG12”, “NITF20” only	X		X		X	X									

(4) “TFRD43”, “TFRD” only	X		X		X	X									
f. Provide one of the following in ProductDataFormat:															
(1) “NITF20”	X		X		X	X									
(2) “TFRD”	X		X		X	X									
g. Provide horizontal direction in HorizMotionRate					X										
h. Provide vertical direction in VertMotionRate					X										
i. Set BrowseImageReturned					X		X	X	X	X		X	X		

Table D-24. Subtest 24, Client Set_Parameters()

SUBTEST 24, CLIENT SET_PARAMETERS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Requests changes to list of parameters	X		X		X	X	X	X	X	X		X	X		
b. ImageProduct uses username/password in ContextInfo.					X										
2. Method Exceptions:															
a. BadName)					X										
b. BadValue					X										
c. CannotSet					X										
d. BAD_INV_ORDER															
3. Parameters Interface:															
a. Sets ResultAttributes to those returned by catalog queries	X				X		X	X	X	X		X	X		
b. Sets GeographicDatum to “WGS84”	X				X	X		X	X	X		X			
c. Request reduced resolution sets in ProductRRDS					X	X									
d. Set horizontal/vertical sizes & resolution in RegionParameters			X		X										
e. Set one of the following in ProductCompression:															
(1) “UNC”, “NITF20” only	X		X		X	X									
(2) “JPEG8”, “NITF20” only	X		X		X	X									
(3) “JPEG12”, “NITF20” only	X		X		X	X									

(4) “TFRD43”, “TFRD” only	X		X		X	X									
f. Set one of the following in ProductDataFormat:															
(1) “NITF20”	X		X		X	X									
(2) “TFRD”	X		X		X	X									
g. Set horizontal direction in HorizMotionRate					X										
h. Set vertical direction in VertMotionRate					X										
i. Provide current setting of BrowseImageReturned					X		X	X	X	X		X	X		

Table D-25. Subtest 25, Server Get_Subimage()

SUBTEST 25, SERVER GET_SUBIMAGE()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Subimage encompasses bounding box (unless unavailable)						X									
b. Upper_left parameter defines the NW corner of bounding box						X									
c. Lower_right parameter defines the SE corner of bounding box						X									
d. Location parameter defines where product is transferred						X									
e. A valid unique RequestId is returned						X									
2. Method Exceptions:															
a. BadCoord (unacceptable coordinate)						X									
b. BadCoord (no bounding box coverage))						X									
c. BadProductReference						X									
d. BadLocationSpec						X									
e. TooManyRequests															
f. BadRegionData when data requested is in the wrong format						X									
g. BadRegionData when data compression is unavailable						X									
h. BAD_INV_ORDER															
3. Parameters Interface:															
a. Generate subimage with ProductDataFormat set to:															
(1) “NITF20”						X									

(2) “TFRD”						X									
b. Generate subimage with ProductCompression set to:															
(1) “UNC”, “NITF20” only.						X									
(2) “JPEG8”, “NITF20” only.						X									
(3) “JPEG12”, “NITF20” only.						X									
(4) “TFRD43”, “TFRD” only.						X									
c. Generate subimage with ProductRDDS						X									
d. Generate subimage with GeographicDatum set to “WGS84”						X									

Table D-26. Subtest 26, Client Get_Subimage()

SUBTEST 26, CLIENT GET_SUBIMAGE()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Requests a portion of specified image product						X									
b. Upper_left parameter defines the NW corner of bounding box						X									
c. Lower_right parameter defines the SE corner of bounding box						X									
d. Returned RequestIds are used to monitor progress of request						X									
2. Method Exceptions:															
a. BadCoord						X									
b. BadProductReference						X									
c. BadLocationSpec						X									
d. TooManyRequests															
e. BadRegionData						X									
f. BAD_INV_ORDER															
3. Parameters Interface:															
a. Set ProductDataFormat to:															
(1) NITF20”						X									
(2) TFRD”						X									
b. Set ProductCompression to:															
(1) “UNC”, “NITF20” only						X									

(2) “JPEG8”, “NITF20” only						X									
(3) “JPEG12”, “NITF20” only						X									
(4) “TFRD43”, “TFRD” only						X									
c. Set ProductRDDS						X									
d. Set GeographicDatum to “WGS84”						X									

Table D-27. Subtest 27, Server Boolean_Query()

SUBTEST 27, SERVER BOOLEAN_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Returns valid QueryId with data specified in Boolean query							X						X		
2. Method Exceptions:															
a. BadQuerySyntax (when illegal syntax)							X								
b. BadQuerySyntax (when improper operation applied)							X								
c. BadAttribute															
d. BadQueryValue															
e. TooManyRequests															
f. BAD_INV_ORDER				X											
3. Parameters Interface:															
a. ResultAttributes returned specifying available attributes													X		
b. BrowseImageReturned													X		

Table D-28. Subtest 28, Client Boolean_Query()

SUBTEST 28, CLIENT BOOLEAN_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provides valid Boolean query							X						X		
b. QueryId returned is used by get_results to retrieve data							X						X		
2. Method Exceptions:															
a. BadQuerySyntax							X								
b. BadAttribute															
c. BadQueryValue															
d. TooManyRequests															
e. BAD_INV_ORDER				X											
3. Parameters Interface:															
a. ResultAttributes specifies which attributes are to be returned							X						X		
b. BrowseImageReturned							X						X		

Table D-29. Subtest 29, Server Polygonal_Query()

SUBTEST 29, SERVER POLYGONAL_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Returns valid QueryId with data specified in polygonal query								X							
b. Polygon is in a single enclosed area								X							
c. Polygon vertices are ordered in a counterclockwise direction								X							
2. Method Exceptions:															
a. BadQuerySyntax (when illegal syntax)															
b. BadQuerySyntax (when improper operation applied)															
c. BadAttribute															
d. BadQueryValue															
e. TooManyRequests															
f. BAD_INV_ORDER															
g. BadCoord								X							
h. TooFewVertices								X							
i. TooManyVertices								X							
3. Parameters Interface:															
a. ResultAttributes are returned when query results retrieved								X							
b. GeographicDatum is set to “WGS84”								X							
c. BrowseImageReturned								X							

Table D-30. Subtest 30, Client Polygonal_Query()

SUBTEST 30, CLIENT POLYGONAL_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provides valid Boolean query and polygon								X							
b. Received QueryId is used by get_results to retrieve data								X							
c. Polygon is a single enclosed area								X							
d. Polygon has from 3 to MaxPolygonVertices vertices								X							
e. Polygon vertices are ordered in counterclockwise direction								X							
2. Method Exceptions:															
a. BadQuerySyntax															
b. BadAttribute															
c. BadQueryValue															
d. TooManyRequests															
e. BAD_INV_ORDER															
f. BadCoord								X							
g. TooFewVertices								X							
h. TooManyVertices								X							
3. Parameters Interface:															
a. ResultAttributes specified which attributes are to be returned								X							
b. Set GeographicDatum to “WGS84”								X							

c. BrowseImageReturned								X							

Table D-31. Subtest 31, Server Elliptical_Query()

SUBTEST 31, SERVER ELLIPTICAL_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Returned QueryId used by get_results to return data									X						
b. Azimuth is measured in degrees from north of major axis									X						
c. Major and minor axes are measured in meters									X						
d. Circle query generated when major and minor axes are equal									X						
2. Method Exceptions:															
a. BadQuerySyntax (when illegal syntax)															
b. BadQuerySyntax (when improper operation applied)															
c. BadAttribute									X						
d. BadQueryValue									X						
e. TooManyRequests															
f. BAD_INV_ORDER															
g. BadCoord															
h. BadEllipse									X						
3. Parameters Interface:															
a. Returned ResultAttributes specify available attributes									X						
b. Set GeographicDatum to “WGS84”									X						
c. BrowseImageReturned									X						

Table D-32. Subtest 32, Client Elliptical_Query()

SUBTEST 32, CLIENT ELLIPTICAL_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provides a valid boolean query and ellipse									X						
b. Returned QueryId used by get_results to retrieve data									X						
b. Azimuth is measured in degrees from north of major axis									X						
c. Major and minor axes are measured in meters									X						
d. Circle query generated when major and minor axes are equal									X						
2. Method Exceptions:															
a. BadQuerySyntax															
b. BadAttribute									X						
c. BadQueryValue									X						
d. TooManyRequests															
e. BAD_INV_ORDER															
f. BadCoord															
g. BadEllipse									X						
3. Parameters Interface:															
a. ResultAttributes specifies which attributes are to be returned									X						
b. Set GeographicDatum to “WGS84”									X						
c. BrowseImageReturned									X						

Table D-33. Subtest 33, Server Point_Query()

SUBTEST 33, SERVER POINT_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Returns valid QueryId with data specified in point query										X					
2. Method Exceptions:															
a. BadQuerySyntax (when illegal syntax)										X					
b. BadQuerySyntax (when improper operation applied)										X					
c. BadAttribute															
d. BadQueryValue															
e. TooManyRequests															
f. BAD_INV_ORDER															
g. BadCoord										X					
3. Parameters Interface:															
a. ResultAttributes are returned when query results retrieved										X					
b. GeographicDatum is set to “WGS84”										X					
c. BrowseImageReturned										X					

Table D-34. Subtest 34, Client Point_Query()

SUBTEST 34, CLIENT POINT_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Provides a valid boolean query and geographic point										X					
b. Received QueryId used by get_results to retrieve data										X					
2. Method Exceptions:															
a. BadQuerySyntax										X					
b. BadAttribute															
c. BadQueryValue															
d. TooManyRequests															
e. BAD_INV_ORDER															
f. BadCoord										X					
3. Parameters Interface:															
a. ResultAttributes specified which attributes are to be returned										X					
b. Set GeographicDatum to “WGS84”										X					
c. BrowseImageReturned										X					

Table D-35. Subtest 35, Server Get_Results()

SUBTEST 35, SERVER GET_RESULTS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. QueryID is used to specify which results are to be retreived	X	X	X	X		X	X	X	X	X	X		X		
b. Number_of_hits_to_return provides number of hits returned	X	X	X	X		X	X	X	X	X	X		X		
c. Number_of_hits_remaining provides unretrieved hits left	X	X	X	X		X	X	X	X	X	X		X		
d. Product reference returned for each product found	X	X	X			X									
e. Returned attributes are in character form	X	X	X	X		X	X	X	X	X	X		X		
f. Attributes in QueryHit and QueryResults are in same order	X	X	X	X		X	X	X	X	X	X		X		
g. Returned browse_images are 1024x1024x8	X	X	X	X		X	X	X	X	X	X				
2. Method Exceptions:															
a. BadQueryId							X	X	X	X					
b. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-36. Subtest 36, Client Get_Results()

SUBTEST 36, CLIENT GET_RESULTS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. QueryId is used to specify results to be retrieved	X	X	X	X		X	X	X	X	X	X		X		
b. Number_of_hits_to_return specifies those hits to be returned	X	X	X	X		X	X	X	X	X	X		X		
c. Get_results can be called until number_of_hits_remaining = 0	X	X	X	X		X	X	X	X	X	X		X		
d. Returned product references can be used to access products	X	X	X			X									
e. Attributres are received in character form	X	X	X	X		X	X	X	X	X	X		X		
f. Attributes in QueryHit and QueryResult are in same order	X	X	X	X		X	X	X	X	X	X		X		
g. Returned browse_images are 1024x1024x8 or 512x512	X	X	X	X		X	X	X	X	X	X				
h. Last get_results may return less results than number_hits_to_return.															
2. Method Exceptions:															
a. BadQueryId							X	X	X	X					
b. BAD_INV_ORDER															
3. Parameters Interface:															

Table D-37. Subtest 37, Server CA Free_Results()

SUBTEST 37, SERVER CA FREE_RESULTS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Returns valid QueryId with query results to free							X	X	X						
b. Returns hits for QueryId that are no longer accessible							X	X	X						
2. Method Exceptions:															
a. BadQueryId							X								
b. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-38. Subtest 38, Client CA Free_Results()

SUBTEST 38, CLIENT CA FREE_RESULTS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. QueryId specifies query results to free							X	X	X						
2. Method Exceptions:															
a. BadQueryId							X								
b. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-39. Subtest 39, Server P&NF Standing Queries

SUBTEST 39, SERVER P&NF STANDING QUERIES															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Boolean_query same as for CAF; QueryId valid indefinitely										X	X				
b. Polygonal_query same as for CAF; QueryId valid indefinitely										X	X				
c. Elliptical_query same as for CAF; QueryId valid indefinitely										X	X				
d. Point_query same as for CAF; QueryId valid indefinitely										X	X				
e. P&NF results persist until free_results or remove_query called										X					
2. Method Exceptions: (Same as query methods in CAF)															
3. Parameters Interface: (Same as query methods in CAF)															

Table D-40. Subtest 40, Client P&NF Standing Queries

SUBTEST 40, CLIENT P&NF STANDING QUERIES															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Boolean_query same as for CAF; submits standing query											X	X			
b. Polygonal_query same as for CAF; submits standing query											X	X			
c. Elliptical_query same as for CAF; submits standing query											X	X			
d. Point_query same as for CAF; submits standing query											X	X			
e. Received QueryId is used by get_results to retrieve data											X				
2. Method Exceptions: (Same as query methods in CAF)															
3. Parameters Interface: (Same as query methods in CAF)															

Table D-41. Subtest 41, Server P&NF Free_Results()

SUBTEST 41, SERVER P&NF FREE_RESULTS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Only retrieved query results from QueryId are removed											X				
2. Method Exceptions:															
a. BadQueryId											X				
b. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-42. Subtest 42, Client P&NF Free_Results()

SUBTEST 42, CLIENT P&NF FREE_RESULTS()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Specifies which QueryId to delete results previously received											X				
2. Method Exceptions:															
a. BadQueryId											X				
b. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-43. Subtest 43, Server List_Queries()

SUBTEST 43, SERVER LIST_QUERIES()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. List of all standing QueryIds are returned											X	X			
b. New_results is set to TRUE for unretrieved QueryId results											X	X			
2. Method Exceptions:															
a. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-44. Subtest 44, Client List_Queries()

SUBTEST 44, CLIENT LIST_QUERIES()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. List of all standing QueryIds is received											X	X			
b. New_results is set to TRUE for unretrieved QueryId results											X	X			
c. Maintains correlation between QueryId and query details											X	X			
2. Method Exceptions:															
a. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-45. Subtest 45, Server Remove_Query()

SUBTEST 45, SERVER REMOVE_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Query and query results for associated QueryId are removed											X	X			
2. Method Exceptions:															
a. BadQueryId											X				
b. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-46. Subtest 46, Client Remove_Query()

SUBTEST 46, CLIENT REMOVE_QUERY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Requests cancellation of specified QueryId										X	X				
2. Method Exceptions:															
a. BadQueryId										X					
b. BAD_INV_ORDER															
3. Parameters Interface: None															

Table D-47. Subtest 47, Server Request_Notify()

SUBTEST 47, SERVER REQUEST_NOTIFY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Notify of new products if PNF_Callback is not NULL												X			
b. Notify of new products if email_address is not NULL												X			
2. Method Exceptions:															
a. BadQueryId												X			
b. BadEmailAddress												X			
c. BAD_PARAM												X			
d. INV_OBJREF												X			
3. Parameters Interface: None															

Table D-48. Subtest 48, Client Request_Notify()

SUBTEST 48, CLIENT REQUEST_NOTIFY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Requests notification of new products via callback_objectref												X			
b. Notified of new products if email_address is not NULL												X			
c. Either callback_objectref or email_address NULL; not both												X			
2. Method Exceptions:															
a. BadQueryId												X			
b. BadEmailAddress												X			
b. BAD_PARAM												X			
c. INV_OBJREF												X			
3. Parameters Interface: None															

Table D-49. Subtest 49, Server Request_Push(). Method deleted from specification.

Table D-50. Subtest 50, Client Request_Push(). Method deleted from specification.

Table D-51. Subtest 51, Server Notify()

SUBTEST 51, SERVER NOTIFY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Notify method called when new hits on standing query												X			
b. Query_results contains new hits for that query												X			
2. Method Exceptions: None															
3. Parameters Interface: None															

Table D-52. Subtest 52, Client Notify()

SUBTEST 52, CLIENT NOTIFY()															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Receives the notify call												X			
b. Maintains correlation between QueryId and query details												X			
2. Method Exceptions: None															
3. Parameters Interface: None															

Table D-53. Subtest 53, Server Push(). Method deleted from specification.

Table D-54. Subtest 54, Client Push(). Method deleted from specification.

Table D-55. Subtest 55, Boolean Query Syntax

SUBTEST 55,Boolean Query Syntax															
Criteria	Test Scenario														
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1. Method Functionality:															
a. Query Composition Types															
(1) Product Expression (2 or more Sum Expressions or Relation Expressions ANDed)													X		
(2) Sum Expression (2 or more Relation Expressions ORed), where attributes are same and operators are: = , < > ,like, not like (ORed expressions will be enclosed in parentheses)													X		
(3) Relation Expression (attribute, operator, and constant)													X		
b. Operators															
(1) Numeric operators: = , < > , > = , < = , > , <													X		
(2) String operators: like, not like													X		
c. Constants															
(1) Numeric constants: integers, floating point numbers													X		
(2) Constant strings are enclosed in single quotations													X		
(3) Wildcard character %, within string constant, matches 0 or more characters													X		
2. Method Exceptions: N/A															
3. Parameters Interface: N/A															