



Pacific Northwest Water Quality Data Exchange

Flow Configuration Document

Version 1.3

August 16, 2004

Version 1.3 prepared by:



Business Systems Development
Management Services Division

811 SW Sixth Avenue
Portland, OR 97204
(503) 229 6086

TABLE OF CONTENTS

1) NETWORK INFORMATION EXCHANGE..... 1

Common Data Services 1

PNWWQX.GetProjects 3

PNWWQX.GetStations 9

PNWWQX.GetMeasurements 15

PNWWQX.GetDCDataSources..... 21

PNWWQX.GetDCTaxonNames..... 24

PNWWQX.GetDCAnalyteNames 27

PNWWQX.GetDCStationsList..... 30

PNWWQX.GetDCReferenceData 33

Flows..... 38

Load Water Quality Data to Host Database..... 38

Load Data Source Metadata to Data Catalog..... 40

2) NODE SPECIFIC FLOW IMPLEMENTATION SHEET TEMPLATE 42

3) GENERIC DIAGNOSTICS/FEEDBACK FOR THE ENTIRE EXCHANGE 42

THIS PAGE INTENTIONALLY LEFT BLANK

1) Network Information Exchange

Common Data Services

Three common data services have been specified to be provided by the Pacific Northwest Water Quality Data Exchange (Exchange) nodes, with additional services provided only by the Oregon DEQ node for the Exchange Data Catalog:

- GetProjects
- GetStations
- GetMeasurements
- GetDCDataSources (Oregon DEQ only)
- GetDCTaxonNames (Oregon DEQ only)
- GetDCAnalyteNames (Oregon DEQ only)
- GetDCStationsList (Oregon DEQ only)
- GetDCReferenceData (Oregon DEQ only)

Additional data services are expected to be developed as the Exchange becomes operational to satisfy application design needs. For example, a specific data service may be required in the future to return a list of the station types used by a given data provider. Such additional data services may be targeted to a partner node or the Exchange Data Catalog depending on the nature of the query.

Figure 1 illustrates how these data services will be supported within the Exchange partner network node architecture, and how the required information will flow from the data source to the information requestor.

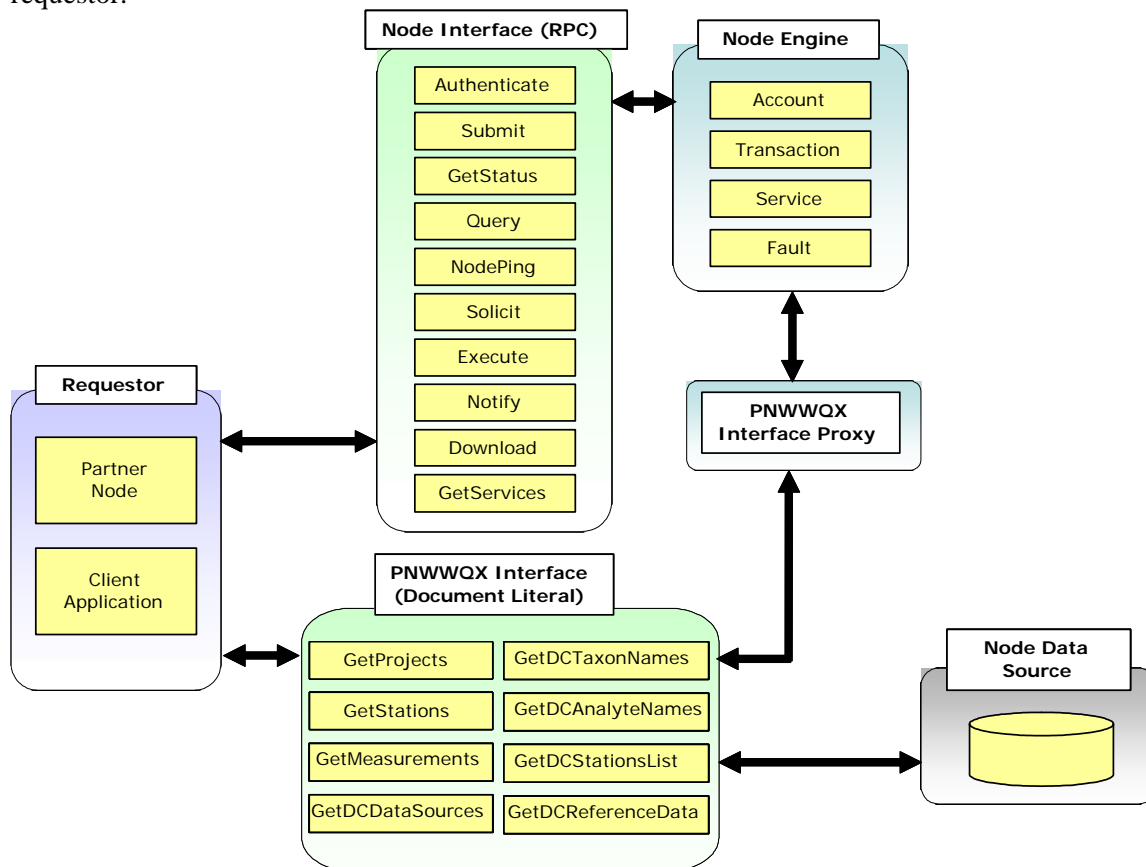


Figure 1: Node Information Flow

Remote Procedure Call Request Process

The national *Network Node Functional Specification Document V.1.1* requires each network node to utilize a “remote procedure call” or RPC methodology to process incoming requests for information against the partner’s data sources using a number of standard Web methods. The function of each of these methods is described in detail in the specification document.

Besides providing operational information about the partner node and ensuring that the appropriate security controls are applied, these methods serve to accept and pass on incoming requests from a requestor to either receive or supply information. These requests may be generated from another partner node or from a custom client application.

The standard Remote Procedure Call (RPC) node interface will pass on an incoming request to the Node Engine. The Node Engine parses the incoming request and then invokes the appropriate network service or flow-specific request processor to respond to the request. The request processor will then interrogate the appropriate partner data source.

The request processor will transform the results from the execution of the query into an XML document using the appropriate schema definition and will return a reference to this document to the Node Engine. The Node Engine in turn then returns this document to the requestor through the node interface. This mechanism is characterized as using an RPC methodology because neither the type or number of parameters passed to the node Query method, nor the values returned, are described by that method. Instead, these values are simply passed on as an array of strings.

Exchange Node Request Process

For the purposes of supporting the Exchange common data services, the Exchange partner nodes will support both the RPC methodology described above, as well as providing an additional “Document Literal” interface that might be accessed by the requestor directly to invoke the Exchange query methods.

The Document Literal interface will be provided by a single Exchange Web Service that in turn will expose the Exchange-specific query methods, for example, “GetProjects”, on the partner node such that they may be invoked directly by the requestor node or client application. For example, the requestor may invoke the GetProjects method of the Web Service passing the appropriate parameters described by this method. The method will execute and will then interrogate the appropriate partner data source. This essentially removes a layer of functionality from the process and enables the requestor to directly interact with the Exchange methods. This will provide a number of efficiencies, the most notable of which is gaining the ability to provide multiple well-defined input parameters and to easily provide arrays of values for a given parameter where allowed. The method will then transform the results from the execution of the query into an XML document using the appropriate schema definition, before returning this document to the requestor directly.

The RPC functionality will operate in the same fashion as described above with the standard node interface passing on a request to the Node Engine. The Node Engine parses the incoming request and then invokes the appropriate network service, for example, the Exchange service, to respond to the request. In this case, however, rather than executing the appropriate query method itself, the Exchange service will invoke the same document literal interface methods described above by means of a proxy component.

In summary, the core business logic inherent in the Exchange query methods will therefore exist in only one location and may be invoked directly using the document literal approach, or indirectly using the standard node RPC architecture. This approach to the implementation of the Exchange query methods will allow the node to be both highly scalable for future architectural changes, and at the same time to meet the current Network requirements encapsulated in the *Network Node Functional Specification Document V.1.1*.

PNWWQX.GetProjects

a) Exchange Availability

Provided by all Exchange nodes.

b) Naming Convention

PNWWQX.GetProjects

c) Data Service Timing/Initiation

On request.

No scheduled use of this data service is expected, although future applications may utilize scheduled requests as appropriate.

d) Payload Format (Schema)

PNWWQX_Projects_v.1.3.xsd

The Exchange DET has been designed to accommodate the transfer of unstructured documents or binary objects along with the appropriate XML payload. As specified by the *Network Node Functional Specification Document V.1.1*, such binary objects must be transferred as attachments to the SOAP message that encapsulates the XML payload using the DIME protocol.

If appropriate, the XML payload will include one or more references to the attached objects using a standard complex data type structure that includes an *href* attribute of content Id (CID) referencing the relevant attached document.

Alternatively, a URL may be provided to describe where the object is made available for download.

e) RPC Interface

Data Service Type

Query

The requesting node or application may invoke this data service using the Query method; typically where a smaller record set is expected. The following standard arguments must be provided as specified by the *Network Node Functional Specification Document V.1.1*:

securityToken: A security ticket issued by the service provider or a trusted security provider.

request: The name of the information request to be processed. This will be *PNWWQX.GetProjects*.

rowId: The starting row for the result set, it is a zero based index to the current result set. The value of rowId must be 0 if positioned-fetch is not requested.

maxRow: The maximum number of rows to be returned. The service provider uses a default value if maxRow is 0 or negative. The value of maxRow must be -1 if positioned-fetch is not requested. A special value, -1, means all records from the current row (specified by rowId) to the end of the result set.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the “|” character.

Solicit

The requesting node or application may invoke this data service using the Solicit method; typically where a very large record set is expected. The requestor must then be prepared to execute the necessary GetStatus and Download methods to retrieve the results, and to expect that the query request will take time to satisfy. It is expected that most large record sets resulting from a Solicit will be downloaded by the requestor node or application since the primary use of this data service will likely be application based.

securityToken: A security ticket issued by the service provider or a trusted security provider.

returnURL: A Node address where results can be submitted. The service provider must call the Submit method at the specified address if this argument is not empty. If the argument is empty, then it is the requester’s responsibility to download the result. It is expected that most large record sets resulting from a Solicit will be downloaded by the requestor.

request: The name of the information request to be processed. This will be *PNWWQX.GetProjects*.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the “|” character.

Parameters, Order, and Format

The following parameter values may be provided to the query in the defined sequence.

<i>Seq.</i>	<i>Parameter</i>	<i>Format</i>
1	rowId	Integer
2	maxRows	Integer
3	Providing organization name	String
4	Project organization name	String
5	Project name	String
6	Project start date	Date
7	Project end date	Date
8	Station contact organization name	String
9	Maximum location latitude	Decimal
10	Maximum location longitude	Decimal
11	Minimum location latitude	Decimal
12	Minimum location longitude	Decimal
13	Location descriptor context	String
14	Location descriptor name – unlimited array of values allowed (in combination with single context value)	String
15	Station type– unlimited array of values allowed	String
16	Station name	String
17	Sampling organization name	String
18	Field event start date	Date
19	Field event end date	Date
20	Sampled media– unlimited array of values allowed	String
21	Analyte name – unlimited array of values allowed	String
22	Taxonomic name– unlimited array of values allowed	String

Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXRead*.

f) Document Literal Interface

Definition

The Document Literal message interface will be defined by the appropriate WSDL file, including the methods and the type and number of parameters that can be accepted by the *PNWWQX* Web Service.

Arguments

The following parameters will be accepted by the *GetProjects* method.

<i>Parameter</i>	<i>Format</i>
rowId	Integer
maxRows	Integer
Providing organization name	String
Project organization name	String
Project name	String
Project start date	Date
Project end date	Date
Station contact organization name	String
Maximum location latitude	Decimal
Maximum location longitude	Decimal
Minimum location latitude	Decimal
Minimum location longitude	Decimal
Location descriptor context	String
Location descriptor name – unlimited array of values allowed (in combination with single context value)	String
Station type– unlimited array of values allowed	String
Station name	String
Sampling organization name	String
Field event start date	Date
Field event end date	Date
Sampled media– unlimited array of values allowed	String
Analyte name – unlimited array of values allowed	String
Taxonomic name– unlimited array of values allowed	String

Security

The *PNWWQX* Web Service and methods will be publicly available on the relevant Web server.

g) Data Service Management and Workflow

Query Parameters

The data service parameters will be used to filter the data that is provided in response to the query. For example, when a given criterion is provided, only records meeting that criterion will be returned. When that criterion is NOT provided, all records will be returned. When the data service parameter allows an array of values, records containing any (rather than all) of the specified values within the array will be returned. Filtering will be additive, in that only records meeting all input criteria will be provided.

Wild card values will be accepted for the following parameters:

- Providing organization name
- Project organization name
- Project name
- Station contact organization name
- Station name
- Sampling organization name

When the user wishes to use wild cards for a request, this must be indicated by means of the “%” character. The user may use these characters to request “begins with”, “ends with” and “contains” queries. The Node request processor will parse the incoming query parameters for the “%” character and will replace them with the appropriate wild card character for the target database platform.

Paging

Paging through result record sets will be accomplished using the starting row and maximum number of records parameters.

The query will order the returned records by:

- Providing organization name
- Project name

Error Handling

The following standard Network Node generic fault messages may be generated.

Error Code	Description
E_UnknownUser	User authentication failed
E_Query	The supplied database logic failed
E_TransactionId	A transaction ID could not be found
E_UnknownMethod	The requested method is not supported
E_ServiceUnavailable	The requested service is unavailable
E_AccessDenied	The operation could not be performed due to lack of privilege
E_InvalidToken	The securityToken is invalid
E_TokenExpired	The securityToken has expired
E_FileNotFound	The requested file could not be located
E_ValidationFailed	DET validation error
E_ServerBusy	The service is too busy to handle the request at this time, please try later
E_RowIdOutOfRange	The rowId parameter is out of range
E_FeatureUnsupported	The requested feature is not supported
E_VersionMismatch	The request is a different version of the protocol
E_InvalidFileName	The name element in the nodeDocument structure is invalid
E_InvalidFileType	The type element in the nodeDocument structure is invalid or not supported
E_InvalidDataFlow	The dataflow element in a request message is not supported
E_InvalidParameter	One of the input parameters is invalid
E_InternalError	An unrecoverable error occurred during processing the request
E_InvalidSQL	Syntax error in the SQL statement
E_AuthMethod	The authentication method is not supported
E_AccessRight	User privilege is insufficient for the operation

PNWWQX.GetStations

a) Exchange Availability

Provided by all Exchange nodes.

b) Naming Convention

PNWWQX.GetStations

c) Data Service Timing/Initiation

On request.

No scheduled use of this data service is expected, although future applications may utilize scheduled requests as appropriate.

d) Payload Format (Schema)

PNWWQX Stations v.1.3.xsd

The Exchange DET has been designed to accommodate the transfer of unstructured documents or binary objects along with the appropriate XML payload. As specified by the *Network Node Functional Specification Document V.1.1*, such binary objects must be transferred as attachments to the SOAP message that encapsulates the XML payload using the DIME protocol.

If appropriate, the XML payload will include one or more references to the attached objects using a standard complex data type structure that includes an *href* attribute of content Id (CID) referencing the relevant attached document.

e) RPC Interface

Data Service Type

Query

The requesting node or application may invoke this data service using the Query method; typically where a smaller record set is expected. The following standard arguments must be provided as specified by the *Network Node Functional Specification Document V.1.1*:

securityToken: A security ticket issued by the service provider or a trusted security provider.

request: The name of the information request to be processed. This will be *PNWWQX.GetStations*.

rowId: The starting row for the result set, it is a zero based index to the current result set. The value of rowId must be 0 if positioned-fetch is not requested.

maxRow: The maximum number of rows to be returned. The service provider uses a default value if maxRow is 0 or negative. The value of maxRow must be -1 if positioned-fetch is not requested. A special value, -1, means all records from the current row (specified by rowId) to the end of the result set.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the “|” character.

Solicit

The requesting node or application may invoke this data service using the Solicit method; typically where a very large record set is expected. The requestor must then be prepared to execute the necessary GetStatus and Download methods to retrieve the results, and to expect that the query request will take time to satisfy. It is expected that most large record sets resulting from a Solicit will be downloaded by the requestor node or application since the primary use of this data service will likely be application based.

securityToken: A security ticket issued by the service provider or a trusted security provider.

returnURL: A Node address where results can be submitted. The service provider must call the Submit method at the specified address if this argument is not empty. If the argument is empty, then it is the requester's responsibility to download the result. It is expected that most large record sets resulting from a Solicit will be downloaded by the requestor.

request: The name of the information request to be processed. This will be *PNWWQX.GetStations*.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the "|" character.

Parameters, Order, and Format

The following parameter values may be provided to the query in the defined sequence.

<i>Seq.</i>	<i>Parameter</i>	<i>Format</i>
1	rowId	Integer
2	maxRows	Integer
3	Providing organization name	String
4	Project organization name	String
5	Project name	String
6	Project start date	Date
7	Project end date	Date
8	Station contact organization name	String
9	Maximum location latitude	Decimal
10	Maximum location longitude	Decimal
11	Minimum location latitude	Decimal
12	Minimum location longitude	Decimal
13	Location descriptor context	String
14	Location descriptor name – unlimited array of values allowed (in combination with single context value)	String
15	Station type– unlimited array of values allowed	String
16	Station name	String
17	Sampling organization name	String
18	Field event start date	Date
19	Field event end date	Date
20	Sampled media– unlimited array of values allowed	String
21	Analyte name – unlimited array of values allowed	String
22	Taxonomic name– unlimited array of values allowed	String

Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXRead*.

f) Document Literal Interface

Definition

The Document Literal message interface will be defined by the appropriate WSDL file, including the methods and the type and number of parameters that can be accepted by the *PNWWQX* Web Service.

Arguments

The following parameters will be accepted by the *GetStations* method.

<i>Parameter</i>	<i>Format</i>
rowId	Integer
maxRows	Integer
Providing organization name	String
Project organization name	String
Project name	String
Project start date	Date
Project end date	Date
Station contact organization name	String
Maximum location latitude	Decimal
Maximum location longitude	Decimal
Minimum location latitude	Decimal
Minimum location longitude	Decimal
Location descriptor context	String
Location descriptor name – unlimited array of values allowed (in combination with single context value)	String
Station type– unlimited array of values allowed	String
Station name	String
Sampling organization name	String
Field event start date	Date
Field event end date	Date
Sampled media– unlimited array of values allowed	String
Analyte name – unlimited array of values allowed	String
Taxonomic name– unlimited array of values allowed	String

Security

The *PNWWQX* Web Service and methods will be publicly available on the relevant Web server.

g) Data Service Management and Workflow

Query Parameters

The data service parameters will be used to filter the data that is provided in response to the query. For example, when a given criterion is provided, only records meeting that criterion will be returned. When that criterion is NOT provided, all records will be returned. When the data service parameter allows an array of values, records containing any (rather than all) of the specified values within the array will be returned. Filtering will be additive, in that only records meeting all input criteria will be provided.

Wild card values will be accepted for the following parameters:

- Providing organization name
- Project organization name
- Project name
- Station contact organization name
- Station name
- Sampling organization name

When the user wishes to use wild cards for a request, this must be indicated by means of the “%” character. The user may use these characters to request “begins with”, “ends with” and “contains” queries. The Node request processor will parse the incoming query parameters for the “%” character and will replace them with the appropriate wild card character for the target database platform.

Paging

Paging through result record sets will be accomplished using the starting row and maximum number of records parameters.

The query will order the returned records by:

- Providing organization name
- Project name
- Station name

Error Handling

The following standard Network Node generic fault messages may be generated.

Error Code	Description
E_UnknownUser	User authentication failed
E_Query	The supplied database logic failed
E_TransactionId	A transaction ID could not be found
E_UnknownMethod	The requested method is not supported
E_ServiceUnavailable	The requested service is unavailable
E_AccessDenied	The operation could not be performed due to lack of privilege
E_InvalidToken	The securityToken is invalid
E_TokenExpired	The securityToken has expired
E_FileNotFound	The requested file could not be located
E_ValidationFailed	DET validation error
E_ServerBusy	The service is too busy to handle the request at this time, please try later
E_RowIdOutOfRange	The rowId parameter is out of range
E_FeatureUnsupported	The requested feature is not supported
E_VersionMismatch	The request is a different version of the protocol
E_InvalidFileName	The name element in the nodeDocument structure is invalid
E_InvalidFileType	The type element in the nodeDocument structure is invalid or not supported
E_InvalidDataFlow	The dataflow element in a request message is not supported
E_InvalidParameter	One of the input parameters is invalid
E_InternalError	An unrecoverable error occurred during processing the request
E_InvalidSQL	Syntax error in the SQL statement
E_AuthMethod	The authentication method is not supported
E_AccessRight	User privilege is insufficient for the operation

PNWWQX.GetMeasurements

a) Exchange Availability

Provided by all Exchange nodes.

b) Naming Convention

PNWWQX.GetMeasurements

c) Data Service Timing/Initiation

On request.

No scheduled use of this data service is expected, although future applications may utilize scheduled requests as appropriate.

d) Payload Format (Schema)

PNWWQX Measurements v.1.3.xsd

The Exchange DET has been designed to accommodate the transfer of unstructured documents or binary objects along with the appropriate XML payload. As specified by the *Network Node Functional Specification Document V.1.1*, such binary objects must be transferred as attachments to the SOAP message that encapsulates the XML payload using the DIME protocol.

If appropriate, the XML payload will include one or more references to the attached objects using a standard complex data type structure that includes an *href* attribute of content Id (CID) referencing the relevant attached document.

e) RPC Interface

Data Service Type

Query

The requesting node or application may invoke this data service using the Query method; typically where a smaller record set is expected. The following standard arguments must be provided as specified by the *Network Node Functional Specification Document V.1.1*:

securityToken: A security ticket issued by the service provider or a trusted security provider.

request: The name of the information request to be processed. This will be *PNWWQX.GetMeasurements*.

rowId: The starting row for the result set, it is a zero based index to the current result set. The value of rowId must be 0 if positioned-fetch is not requested.

maxRow: The maximum number of rows to be returned. The service provider uses a default value if maxRow is 0 or negative. The value of maxRow must be -1 if positioned-fetch is not requested. A special value, -1, means all records from the current row (specified by rowId) to the end of the result set.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the “|” character.

Solicit

The requesting node or application may invoke this data service using the Solicit method; typically where a very large record set is expected. The requestor must then be prepared to execute the necessary GetStatus and Download methods to retrieve the results, and to expect that the query request will take time to satisfy. It is expected that most large record sets resulting from a Solicit will be downloaded by the requestor node or application since the primary use of this data service will likely be application based.

securityToken: A security ticket issued by the service provider or a trusted security provider.

returnURL: A Node address where results can be submitted. The service provider must call the Submit method at the specified address if this argument is not empty. If the argument is empty, then it is the requester's responsibility to download the result. It is expected that most large record sets resulting from a Solicit will be downloaded by the requestor.

request: The name of the information request to be processed. This will be *PNWWQX.GetMeasurements*.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the "|" character.

Parameters, Order, and Format

The following parameter values may be provided to the query in the defined sequence.

<i>Seq.</i>	<i>Parameter</i>	<i>Format</i>
1	rowId	Integer
2	maxRows	Integer
3	Providing organization name	String
4	Project organization name	String
5	Project name	String
6	Project start date	Date
7	Project end date	Date
8	Station contact organization name	String
9	Maximum location latitude	Decimal
10	Maximum location longitude	Decimal
11	Minimum location latitude	Decimal
12	Minimum location longitude	Decimal
13	Location descriptor context	String
14	Location descriptor name – unlimited array of values allowed (in combination with single context value)	String
15	Station type– unlimited array of values allowed	String
16	Station name	String
17	Sampling organization name	String
18	Field event start date	Date
19	Field event end date	Date
20	Sampled media– unlimited array of values allowed	String
21	Analyte name – unlimited array of values allowed	String
22	Taxonomic name– unlimited array of values allowed	String

Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXRead*.

f) Document Literal Interface

Definition

The Document Literal message interface will be defined by the appropriate WSDL file, including the methods and the type and number of parameters that can be accepted by the *PNWWQX* Web Service.

Arguments

The following parameters will be accepted by the *GetMeasurements* method.

<i>Parameter</i>	<i>Format</i>
rowId	Integer
maxRows	Integer
Providing organization name	String
Project organization name	String
Project name	String
Project start date	Date
Project end date	Date
Station contact organization name	String
Maximum location latitude	Decimal
Maximum location longitude	Decimal
Minimum location latitude	Decimal
Minimum location longitude	Decimal
Location descriptor context	String
Location descriptor name – unlimited array of values allowed (in combination with single context value)	String
Station type– unlimited array of values allowed	String
Station name	String
Sampling organization name	String
Field event start date	Date
Field event end date	Date
Sampled media– unlimited array of values allowed	String
Analyte name – unlimited array of values allowed	String
Taxonomic name– unlimited array of values allowed	String

Security

The PNWWQX Web Service and methods will be publicly available on the relevant Web server.

g) Data Service Management and Workflow

Query Parameters

The data service parameters will be used to filter the data that is provided in response to the query. For example, when a given criterion is provided, only records meeting that criterion will be returned. When that criterion is NOT provided, all records will be returned. When the data service parameter allows an array of values, records containing any (rather than all) of the specified values within the array will be returned. Filtering will be additive, in that only records meeting all input criteria will be provided.

Wild card values will be accepted for the following parameters:

- Providing organization name
- Project organization name
- Project name
- Station contact organization name
- Station name
- Sampling organization name

When the user wishes to use wild cards for a request, this must be indicated by means of the “%” character. The user may use these characters to request “begins with”, “ends with” and “contains” queries. The Node request processor will parse the incoming query parameters for the “%” character and will replace them with the appropriate wild card character for the target database platform.

Paging

Paging through result record sets will be accomplished using the starting row and maximum number of records parameters.

The query will order the returned records by:

- Providing organization name
- Project name
- Station name
- Field event start date
- Analyte name
- Result value

Error Handling

The following standard Network Node generic fault messages may be generated.

Error Code	Description
E_UnknownUser	User authentication failed
E_Query	The supplied database logic failed
E_TransactionId	A transaction ID could not be found
E_UnknownMethod	The requested method is not supported
E_ServiceUnavailable	The requested service is unavailable
E_AccessDenied	The operation could not be performed due to lack of privilege
E_InvalidToken	The securityToken is invalid
E_TokenExpired	The securityToken has expired
E_FileNotFound	The requested file could not be located
E_ValidationFailed	DET validation error
E_ServerBusy	The service is too busy to handle the request at this time, please try later
E_RowIdOutOfRange	The rowId parameter is out of range
E_FeatureUnsupported	The requested feature is not supported
E_VersionMismatch	The request is a different version of the protocol
E_InvalidFileName	The name element in the nodeDocument structure is invalid
E_InvalidFileType	The type element in the nodeDocument structure is invalid or not supported
E_InvalidDataFlow	The dataflow element in a request message is not supported
E_InvalidParameter	One of the input parameters is invalid
E_InternalError	An unrecoverable error occurred during processing the request
E_InvalidSQL	Syntax error in the SQL statement
E_AuthMethod	The authentication method is not supported
E_AccessRight	User privilege is insufficient for the operation

PNWWQX.GetDCDataSources**a) Exchange Availability**

Provided only by Oregon DEQ Node.

b) Naming Convention

PNWWQX.GetDCDataSources

c) Data Service Timing/Initiation

On request.

No scheduled use of this data service is expected, although future applications may utilize scheduled requests as appropriate.

d) Payload Format (Schema)

PNWWQX_DC_DataSources_v.1.3.xsd

e) RPC InterfaceData Service TypeQuery

The requesting node or application may invoke this data service using the Query method; typically where a smaller record set is expected. The following standard arguments must be provided as specified by the *Network Node Functional Specification Document V.1.1*:

securityToken: A security ticket issued by the service provider or a trusted security provider.

request: The name of the information request to be processed. This will be *PNWWQX.GetDCDataSources*.

rowId: The starting row for the result set, it is a zero based index to the current result set. The value of rowId must be 0 if positioned-fetch is not requested.

maxRow: The maximum number of rows to be returned. The service provider uses a default value if maxRow is 0 or negative. The value of maxRow must be -1 if positioned-fetch is not requested. A special value, -1, means all records from the current row (specified by rowId) to the end of the result set.

parameters: None - an empty array.

Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXRead*.

f) Document Literal Interface

Definition

The Document Literal message interface will be defined by the appropriate WSDL file, including the methods and the type and number of parameters that can be accepted by the *PNWWQX* Web Service.

Arguments

No parameters will be accepted by the *GetDCDataSources* method.

Security

The *PNWWQX* Web Service and methods will be publicly available on the relevant Web server.

g) Data Service Management and Workflow

Query Parameters

The data service parameters will be used to filter the data that is provided in response to the query. For example, when a given criterion is provided, only records meeting that criterion will be returned. When that criterion is NOT provided, all records will be returned. Filtering will be additive, in that only records meeting all input criteria will be provided.

Paging

Paging through result record sets will be accomplished using the starting row and maximum number of records parameters.

The query will order the returned records by:

Providing organization name

Error Handling

The following standard Network Node generic fault messages may be generated by this method.

Error Code	Description
E_UnknownUser	User authentication failed
E_Query	The supplied database logic failed
E_TransactionId	A transaction ID could not be found
E_UnknownMethod	The requested method is not supported
E_ServiceUnavailable	The requested service is unavailable
E_AccessDenied	The operation could not be performed due to lack of privilege
E_InvalidToken	The securityToken is invalid
E_TokenExpired	The securityToken has expired
E_FileNotFound	The requested file could not be located
E_ValidationFailed	DET validation error
E_ServerBusy	The service is too busy to handle the request at this time, please try later
E_RowIdOutOfRange	The rowId parameter is out of range
E_FeatureUnsupported	The requested feature is not supported
E_VersionMismatch	The request is a different version of the protocol
E_InvalidFileName	The name element in the nodeDocument structure is invalid
E_InvalidFileType	The type element in the nodeDocument structure is invalid or not supported
E_InvalidDataFlow	The dataflow element in a request message is not supported
E_InvalidParameter	One of the input parameters is invalid
E_InternalError	An unrecoverable error occurred during processing the request
E_InvalidSQL	Syntax error in the SQL statement
E_AuthMethod	The authentication method is not supported
E_AccessRight	User privilege is insufficient for the operation

PNWWQX.GetDCTaxonNames

a) Exchange Availability

Provided only by Oregon DEQ Node.

b) Naming Convention

PNWWQX.GetDCTaxonNames

c) Data Service Timing/Initiation

On request.

No scheduled use of this data service is expected, although future applications may utilize scheduled requests as appropriate.

d) Payload Format (Schema)

PNWWQX_DC_Taxon_v.1.3.xsd

e) RPC Interface

Data Service Type

Query

The requesting node or application may invoke this data service using the Query method; typically where a smaller record set is expected. The following standard arguments must be provided as specified by the *Network Node Functional Specification Document V.1.1*:

securityToken: A security ticket issued by the service provider or a trusted security provider.

request: The name of the information request to be processed. This will be *PNWWQX.GetDCTaxonNames*.

rowId: The starting row for the result set, it is a zero based index to the current result set. The value of rowId must be 0 if positioned-fetch is not requested.

maxRow: The maximum number of rows to be returned. The service provider uses a default value if maxRow is 0 or negative. The value of maxRow must be -1 if positioned-fetch is not requested. A special value, -1, means all records from the current row (specified by rowId) to the end of the result set.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the “|” character.

Parameters, Order, and Format

The following parameter values may be provided to the query in the defined sequence.

<i>Seq.</i>	<i>Parameter</i>	<i>Format</i>
1	Taxon name	String

Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXRead*.

f) Document Literal Interface

Definition

The Document Literal message interface will be defined by the appropriate WSDL file, including the methods and the type and number of parameters that can be accepted by the *PNWWQX* Web Service.

Arguments

The following parameters will be accepted by the *GetDCTaxonNames* method.

<i>Parameter</i>	<i>Format</i>
Taxon name	String

Security

The *PNWWQX* Web Service and methods will be publicly available on the relevant Web server.

g) Data Service Management and Workflow

Query Parameters

The data service parameters will be used to filter the data that is provided in response to the query. For example, when a given criterion is provided, only records meeting that criterion will be returned. When that criterion is NOT provided, all records will be returned. Filtering will be additive, in that only records meeting all input criteria will be provided.

Wild card values will be accepted for the following parameters:

- Taxon name

When the user wishes to use wild cards for a request, this must be indicated by means of the “%” character. The user may use these characters to request “begins with”, “ends with” and “contains” queries. The Node request processor will parse the incoming query parameters for the “%” character and will replace them with the appropriate wild card character for the target database platform.

Paging

Paging through result record sets will be accomplished using the starting row and maximum number of records parameters.

The query will order the returned records by:

Taxon name

Error Handling

The following standard Network Node generic fault messages may be generated by this method.

Error Code	Description
E_UnknownUser	User authentication failed
E_Query	The supplied database logic failed
E_TransactionId	A transaction ID could not be found
E_UnknownMethod	The requested method is not supported
E_ServiceUnavailable	The requested service is unavailable
E_AccessDenied	The operation could not be performed due to lack of privilege
E_InvalidToken	The securityToken is invalid
E_TokenExpired	The securityToken has expired
E_FileNotFound	The requested file could not be located
E_ValidationFailed	DET validation error
E_ServerBusy	The service is too busy to handle the request at this time, please try later
E_RowIdOutOfRange	The rowId parameter is out of range
E_FeatureUnsupported	The requested feature is not supported
E_VersionMismatch	The request is a different version of the protocol
E_InvalidFileName	The name element in the nodeDocument structure is invalid
E_InvalidFileType	The type element in the nodeDocument structure is invalid or not supported
E_InvalidDataFlow	The dataflow element in a request message is not supported
E_InvalidParameter	One of the input parameters is invalid
E_InternalError	An unrecoverable error occurred during processing the request
E_InvalidSQL	Syntax error in the SQL statement
E_AuthMethod	The authentication method is not supported
E_AccessRight	User privilege is insufficient for the operation

PNWWQX.GetDCAnalyteNames

a) Exchange Availability

Provided only by Oregon DEQ Node.

b) Naming Convention

PNWWQX.GetDCAnalyteNames

c) Data Service Timing/Initiation

On request.

No scheduled use of this data service is expected, although future applications may utilize scheduled requests as appropriate.

d) Payload Format (Schema)

PNWWQX_DC_Analytes_v.1.3.xsd

e) RPC Interface

Data Service Type

Query

The requesting node or application may invoke this data service using the Query method; typically where a smaller record set is expected. The following standard arguments must be provided as specified by the *Network Node Functional Specification Document V.1.1*:

securityToken: A security ticket issued by the service provider or a trusted security provider.

request: The name of the information request to be processed. This will be *PNWWQX.GetDCAnalyteNames*.

rowId: The starting row for the result set, it is a zero based index to the current result set. The value of rowId must be 0 if positioned-fetch is not requested.

maxRow: The maximum number of rows to be returned. The service provider uses a default value if maxRow is 0 or negative. The value of maxRow must be -1 if positioned-fetch is not requested. A special value, -1, means all records from the current row (specified by rowId) to the end of the result set.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the “|” character.

Parameters, Order, and Format

The following parameter values may be provided to the query in the defined sequence.

<i>Seq.</i>	<i>Parameter</i>	<i>Format</i>
1	Analyte name	String

Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXRead*.

f) Document Literal Interface

Definition

The Document Literal message interface will be defined by the appropriate WSDL file, including the methods and the type and number of parameters that can be accepted by the *PNWWQX* Web Service.

Arguments

The following parameters will be accepted by the *GetDCAnalyteNames* method.

<i>Parameter</i>	<i>Format</i>
Analyte name	String

Security

The *PNWWQX* Web Service and methods will be publicly available on the relevant Web server.

g) Data Service Management and Workflow

Query Parameters

The data service parameters will be used to filter the data that is provided in response to the query. For example, when a given criterion is provided, only records meeting that criterion will be returned. When that criterion is NOT provided, all records will be returned. Filtering will be additive, in that only records meeting all input criteria will be provided.

Wild card values will be accepted for the following parameters:

- Analyte name

When the user wishes to use wild cards for a request, this must be indicated by means of the “%” character. The user may use these characters to request “begins with”, “ends with” and “contains” queries. The Node request processor will parse the incoming query parameters for the “%” character and will replace them with the appropriate wild card character for the target database platform.

Paging

Paging through result record sets will be accomplished using the starting row and maximum number of records parameters.

The query will order the returned records by:

Analyte name

Error Handling

The following standard Network Node generic fault messages may be generated by this method.

Error Code	Description
E_UnknownUser	User authentication failed
E_Query	The supplied database logic failed
E_TransactionId	A transaction ID could not be found
E_UnknownMethod	The requested method is not supported
E_ServiceUnavailable	The requested service is unavailable
E_AccessDenied	The operation could not be performed due to lack of privilege
E_InvalidToken	The securityToken is invalid
E_TokenExpired	The securityToken has expired
E_FileNotFound	The requested file could not be located
E_ValidationFailed	DET validation error
E_ServerBusy	The service is too busy to handle the request at this time, please try later
E_RowIdOutOfRange	The rowId parameter is out of range
E_FeatureUnsupported	The requested feature is not supported
E_VersionMismatch	The request is a different version of the protocol
E_InvalidFileName	The name element in the nodeDocument structure is invalid
E_InvalidFileType	The type element in the nodeDocument structure is invalid or not supported
E_InvalidDataFlow	The dataflow element in a request message is not supported
E_InvalidParameter	One of the input parameters is invalid
E_InternalError	An unrecoverable error occurred during processing the request
E_InvalidSQL	Syntax error in the SQL statement
E_AuthMethod	The authentication method is not supported
E_AccessRight	User privilege is insufficient for the operation

PNWWQX.GetDCStationsList

a) Exchange Availability

Provided only by Oregon DEQ Node.

b) Naming Convention

PNWWQX.GetDCStationsList

c) Data Service Timing/Initiation

On request.

No scheduled use of this data service is expected, although future applications may utilize scheduled requests as appropriate.

d) Payload Format (Schema)

PNWWQX_DC_Stations_v.1.3.xsd

e) RPC Interface

Data Service Type

Query

The requesting node or application may invoke this data service using the Query method; typically where a smaller record set is expected. The following standard arguments must be provided as specified by the *Network Node Functional Specification Document V.1.1*:

securityToken: A security ticket issued by the service provider or a trusted security provider.

request: The name of the information request to be processed. This will be *PNWWQX.GetDCStationsList*.

rowId: The starting row for the result set, it is a zero based index to the current result set. The value of rowId must be 0 if positioned-fetch is not requested.

maxRow: The maximum number of rows to be returned. The service provider uses a default value if maxRow is 0 or negative. The value of maxRow must be -1 if positioned-fetch is not requested. A special value, -1, means all records from the current row (specified by rowId) to the end of the result set.

parameters: None - an empty array.

Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXRead*.

f) Document Literal Interface

Definition

The Document Literal message interface will be defined by the appropriate WSDL file, including the methods and the type and number of parameters that can be accepted by the *PNWWQX* Web Service.

Arguments

No parameters will be accepted by the *GetDCStationsList* method.

Security

The *PNWWQX* Web Service and methods will be publicly available on the relevant Web server.

g) Data Service Management and Workflow

Query Parameters

The data service parameters will be used to filter the data that is provided in response to the query. For example, when a given criterion is provided, only records meeting that criterion will be returned. When that criterion is NOT provided, all records will be returned. Filtering will be additive, in that only records meeting all input criteria will be provided.

Paging

Paging through result record sets will be accomplished using the starting row and maximum number of records parameters.

The query will order the returned records by:

- Providing organization identifier

- Station identifier

Error Handling

The following standard Network Node generic fault messages may be generated by this method.

Error Code	Description
E_UnknownUser	User authentication failed
E_Query	The supplied database logic failed
E_TransactionId	A transaction ID could not be found
E_UnknownMethod	The requested method is not supported
E_ServiceUnavailable	The requested service is unavailable
E_AccessDenied	The operation could not be performed due to lack of privilege
E_InvalidToken	The securityToken is invalid
E_TokenExpired	The securityToken has expired
E_FileNotFound	The requested file could not be located
E_ValidationFailed	DET validation error
E_ServerBusy	The service is too busy to handle the request at this time, please try later
E_RowIdOutOfRange	The rowId parameter is out of range
E_FeatureUnsupported	The requested feature is not supported
E_VersionMismatch	The request is a different version of the protocol
E_InvalidFileName	The name element in the nodeDocument structure is invalid
E_InvalidFileType	The type element in the nodeDocument structure is invalid or not supported
E_InvalidDataFlow	The dataflow element in a request message is not supported
E_InvalidParameter	One of the input parameters is invalid
E_InternalError	An unrecoverable error occurred during processing the request
E_InvalidSQL	Syntax error in the SQL statement
E_AuthMethod	The authentication method is not supported
E_AccessRight	User privilege is insufficient for the operation

PNWWQX.GetDCReferenceData**a) Exchange Availability**

Provided only by Oregon DEQ Node.

b) Naming Convention

PNWWQX.GetDCReferenceData

c) Data Service Timing/Initiation

On request.

No scheduled use of this data service is expected, although future applications may utilize scheduled requests as appropriate.

d) Payload Format (Schema)

PNWWQX_DataCatalog_v.1.3.xsd

e) RPC InterfaceData Service Type*Query*

The requesting node or application may invoke this data service using the Query method; typically where a smaller record set is expected. The following standard arguments must be provided as specified by the *Network Node Functional Specification Document V.1.1*:

securityToken: A security ticket issued by the service provider or a trusted security provider.

request: The name of the information request to be processed. This will be *PNWWQX.GetDCReferenceData*.

rowId: The starting row for the result set, it is a zero based index to the current result set. The value of rowId must be 0 if positioned-fetch is not requested.

maxRow: The maximum number of rows to be returned. The service provider uses a default value if maxRow is 0 or negative. The value of maxRow must be -1 if positioned-fetch is not requested. A special value, -1, means all records from the current row (specified by rowId) to the end of the result set.

parameters: An array of string values representing the query parameters for the information request. Unused parameter values must be indicated in the parameter array. Where an individual query parameter itself includes an array of values, the discrete values will be delimited by the “|” character.

Parameters, Order, and Format

The following parameter values may be provided to the query in the defined sequence.

<i>Seq.</i>	<i>Parameter</i>	<i>Format</i>
1	rowId	Integer
2	maxRows	Integer
3	Providing organization name	String
4	Project organization name	String
5	Project name	String
6	Project start date	Date
7	Project end date	Date
8	Station contact organization name	String
9	Maximum location latitude	Decimal
10	Maximum location longitude	Decimal
11	Minimum location latitude	Decimal
12	Minimum location longitude	Decimal
13	Location descriptor context	String
14	Location descriptor name – unlimited array of values allowed (in combination with single context value)	String
15	Station type– unlimited array of values allowed	String
16	Station name	String
17	Sampling organization name	String
18	Field event start date	Date
19	Field event end date	Date
20	Sampled media– unlimited array of values allowed	String
21	Analyte name – unlimited array of values allowed	String
22	Taxonomic name– unlimited array of values allowed	String

Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXRead*.

f) Document Literal Interface

Definition

The Document Literal message interface will be defined by the appropriate WSDL file, including the methods and the type and number of parameters that can be accepted by the *PNWWQX* Web Service.

Arguments

The following parameters will be accepted by the *GetDCReferenceData* method.

<i>Parameter</i>	<i>Format</i>
rowId	Integer
maxRows	Integer
Providing organization name	String
Project organization name	String
Project name	String
Project start date	Date
Project end date	Date
Station contact organization name	String
Maximum location latitude	Decimal
Maximum location longitude	Decimal
Minimum location latitude	Decimal
Minimum location longitude	Decimal
Location descriptor context	String
Location descriptor name – unlimited array of values allowed (in combination with single context value)	String
Station type– unlimited array of values allowed	String
Station name	String
Sampling organization name	String
Field event start date	Date
Field event end date	Date
Sampled media– unlimited array of values allowed	String
Analyte name – unlimited array of values allowed	String
Taxonomic name– unlimited array of values allowed	String

Security

The PNWWQX Web Service and methods will be publicly available on the relevant Web server.

g) Data Service Management and Workflow

Query Parameters

The data service parameters will be used to filter the data that is provided in response to the query. For example, when a given criterion is provided, only records meeting that criterion will be returned. When that criterion is NOT provided, all records will be returned. When the data service parameter allows an array of values, records containing any (rather than all) of the specified values within the array will be returned. Filtering will be additive, in that only records meeting all input criteria will be provided.

Wild card values will be accepted for the following parameters:

- Providing organization name
- Project organization name
- Project name
- Station contact organization name
- Station name
- Sampling organization name

When the user wishes to use wild cards for a request, this must be indicated by means of the “%” character. The user may use these characters to request “begins with”, “ends with” and “contains” queries. The Node request processor will parse the incoming query parameters for the “%” character and will replace them with the appropriate wild card character for the target database platform.

Paging

Paging through result record sets will be accomplished using the starting row and maximum number of records parameters.

The query will order the returned records by:

- Providing organization name
- Project name
- Station name
- Analyte name
- Taxon name

Error Handling

The following standard Network Node generic fault messages may be generated by this method.

Error Code	Description
E_UnknownUser	User authentication failed
E_Query	The supplied database logic failed
E_TransactionId	A transaction ID could not be found
E_UnknownMethod	The requested method is not supported
E_ServiceUnavailable	The requested service is unavailable
E_AccessDenied	The operation could not be performed due to lack of privilege
E_InvalidToken	The securityToken is invalid
E_TokenExpired	The securityToken has expired
E_FileNotFound	The requested file could not be located
E_ValidationFailed	DET validation error
E_ServerBusy	The service is too busy to handle the request at this time, please try later
E_RowIdOutOfRange	The rowId parameter is out of range
E_FeatureUnsupported	The requested feature is not supported
E_VersionMismatch	The request is a different version of the protocol
E_InvalidFileName	The name element in the nodeDocument structure is invalid
E_InvalidFileType	The type element in the nodeDocument structure is invalid or not supported
E_InvalidDataFlow	The dataflow element in a request message is not supported
E_InvalidParameter	One of the input parameters is invalid
E_InternalError	An unrecoverable error occurred during processing the request
E_InvalidSQL	Syntax error in the SQL statement
E_AuthMethod	The authentication method is not supported
E_AccessRight	User privilege is insufficient for the operation

Flows

The following two information management flows have been defined for the Exchange:

- PNWHostAppend
- PNWDCRefresh

An additional flow will be defined in the future when the transfer of information from the Exchange to the EPA STORET data warehouse is designed and constructed.

These flows will only be supported by the standard Network Node RPC interface. No equivalent Document Literal interface will be provided.

Load Water Quality Data to Host Database

Flow Management and Business Workflow

The objective of this flow is to enable a data provider to upload water quality monitoring data to the central Exchange Host database administered by the Oregon DEQ. This same process might also be used by a data provider to submit data files to other partner nodes as appropriate although the following discussion does not include this additional usage.

The following workflow will occur:

- 1) Data provider compiles the data to be uploaded into one XML document formatted according to the *PNWWQX_Measurements_v.1.3.xsd* schema.
- 2) Data provider invokes the Submit method on the Oregon DEQ node using the “dataflow” argument, “*PNWWQX.PNWHostAppend*” with the “document” argument referencing one XML document attached to the SOAP message.
- 3) If the submission is successful, the Oregon DEQ node will return a transaction id for reference purposes. If not successful, the appropriate standard Network error message will be returned.
- 4) The *PNWWQX.PNWHostAppend* flow processor processes the submitted XML document according to the following logic. This will be scheduled for later execution in a nightly batch process.
 - a. Validate that the organization identifier included in the XML document is known to the Exchange. If the identifier is not valid, the submission will be rejected. If the identifier is valid, the submission will proceed. The Exchange Host database administrator will issue organization identifiers and establish and update providing organization information in the host database.
 - b. For each submitted project, determine whether that project has already been created in the host database. If the project does not exist, a new project record will be created using the data element values included in the XML document. If the project does already exist, then the host database record will be updated using the data element values included in the XML document.
 - c. For each submitted station, determine whether that station has already been created in the host database. If the station does not exist, a new station record will be created using the data element values included in the XML document. If the station does already exist, then the host database record will be updated using the data element values included in the XML document.

- d. For each submitted field activity and measurement result, new records will be created in the host database. No attempt will be made to identify already existing field activities or results for updates.
 - e. If any errors are encountered during processing, the entire submission will be discarded with any updates to the host database being rolled back.
- 5) Messages regarding errors encountered during this processing will be passed back to the data provider via email directly by the *PNWWQX.PNWHostAppend* flow processor.
- 6) At any time, the data provider can confirm the status of the Submit operation using the GetStatus method for the supplied transaction id.

Network Exchange Business Processes

The following Network Exchange Business Processes will be involved in this flow.

Submit

a) Parameters

Standard Submit method arguments with an array of attached documents containing the data sets being submitted each formatted according to the *PNWWQX Measurements v.1.3.xsd* schema.

securityToken: The authentication ticket issued by the node or a trusted security provider enabling the data provider to invoke methods on the Oregon DEQ host node.

transactionId: n/a.

dataflow: The data flow parameter will be set to “*PNWWQX.PNWHostAppend*”.

documents: An array of documents of type nodeDocument. Each nodeDocument structure describes a single XML document attachment.

b) Business Rules

Used for periodic or ad-hoc submission of a water quality data sets to the Exchange host database or potentially to a partner node for population of a partner data source. Use of Submit method will return a transaction id to enable the provider to check the status of processing, or an error if problems are encountered receiving the submission.

c) Payload

Attached payload document formatted according to the following schema:

PNWWQX_Measurements_v.1.3.xsd

Only one document will be attached to the submission. This submitted document will be referenced by a nodeDocument object included in the SOAP message.

d) Flow Timing/Initiation

Ad hoc submissions determined by provider business practices and according to arrangements documented in the Trading Partner Agreement.

e) Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXUpdate*.

GetStatus

a) Parameters

- securityToken: The authentication ticket issued by the node or a trusted security provider enabling the data provider to invoke methods on the host node.
- transactionId: the transaction identification returned by the Submit method.

b) Business Rules

Used to validate the progress of the Submit process using the supplied transaction id.

c) Payload

n/a

d) Flow Timing/Initiation

As deemed appropriate by data provider during the invocation of a node method.

e) Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXUpdate*.

Load Data Source Metadata to Data Catalog

Flow Management and Business Workflow

The objective of this flow is to enable a data provider to refresh the information about their available data sources that is recorded in the Exchange Data Catalog.

The following workflow will occur:

- 1) Data provider compiles the data to be uploaded into one XML document formatted according to the *PNWWQX_DataCatalog_v.1.3* schema.
- 2) Data provider invokes the Submit method on the Oregon DEQ node using the “dataflow” argument, “*PNWWQX.PNWDCRefresh*” with the “documents” argument referencing the XML document which will be attached to the SOAP message.
- 3) If the submission is successful, the Oregon DEQ node will return a transaction id for reference purposes. If not successful, the appropriate standard Network error message will be returned.
- 4) The *PNWWQX.PNWDCRefresh* flow processor processes the submitted XML document according to the following logic. This will be scheduled for later execution in a nightly batch process.
 - a. Validate that the organization identifier included in the XML document is known to the Exchange. If the identifier is not valid, the submission will be rejected. The Data Catalog database administrator will establish and update providing organization and data access information in the host database.
 - b. Delete all records belonging to this providing organization from the Data Catalog database.
 - c. Create new records in the Data Catalog database for each data source included in the submitted XML document.

- d. If any errors are encountered during processing, the entire submission will be discarded with any updates to the host database being rolled back.
- 5) Messages regarding errors encountered during this processing will be passed back to the data provider via email directly by the *PNWWQX.PNWDCRefresh* flow processor.
- 6) At any time, the data provider can confirm the status of the Submit operation using the GetStatus method for the supplied transaction id.

Network Exchange Business Processes

The following Network Exchange Business Processes will be involved in this flow.

Submit

a) Parameters

Standard Submit method arguments with an array of attached documents containing the data sets being submitted each formatted according to the *PNWWQX_DataCatalog_v.1.3* schema.

- securityToken: The authentication ticket issued by the node or a trusted security provider enabling the data provider to invoke methods on the Oregon DEQ host node.
- transactionId: n/a.
- dataflow: The data flow parameter will be set to “*PNWWQX.PNWDCRefresh*”.
- documents: An array of documents of type nodeDocument. Each nodeDocument structure describes a single XML document attachment.

b) Business Rules

Used for periodic or ad-hoc submission of a water quality data sets to the Exchange host database or potentially to a partner node for population of a partner data source. Use of Submit method will return a transaction id to enable the provider to check the status of processing, or an error if problems are encountered receiving the submission.

c) Payload

Attached payload document formatted according to the following schema:

PNWWQX_DataCatalog_v.1.3

Only one document will be attached to the submission. This submitted document will be referenced by a nodeDocument object included in the SOAP message.

d) Flow Timing/Initiation

Ad hoc submissions determined by provider business practices and according to arrangements documented in the Trading Partner Agreement.

e) Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXUpdate*.

GetStatus

a) Parameters

securityToken: The authentication ticket issued by the node or a trusted security provider enabling the data provider to invoke methods on the host node.

transactionId: the transaction identification returned by the Submit method.

b) Business Rules

Used to validate the progress of the Submit process using the supplied transaction id.

c) Payload

n/a

d) Flow Timing/Initiation

As deemed appropriate by data provider during the invocation of a node method.

e) Security

Each user must have a valid User Account for NAAS authentication.

For authorization the user must have access to the NAAS User Role of *PNWWQXUpdate*.

2) Node Specific Flow Implementation Sheet Template

No Node specific implementation considerations identified at the time of writing.

3) Generic Diagnostics/Feedback for the Entire Exchange

None identified at the time of writing.