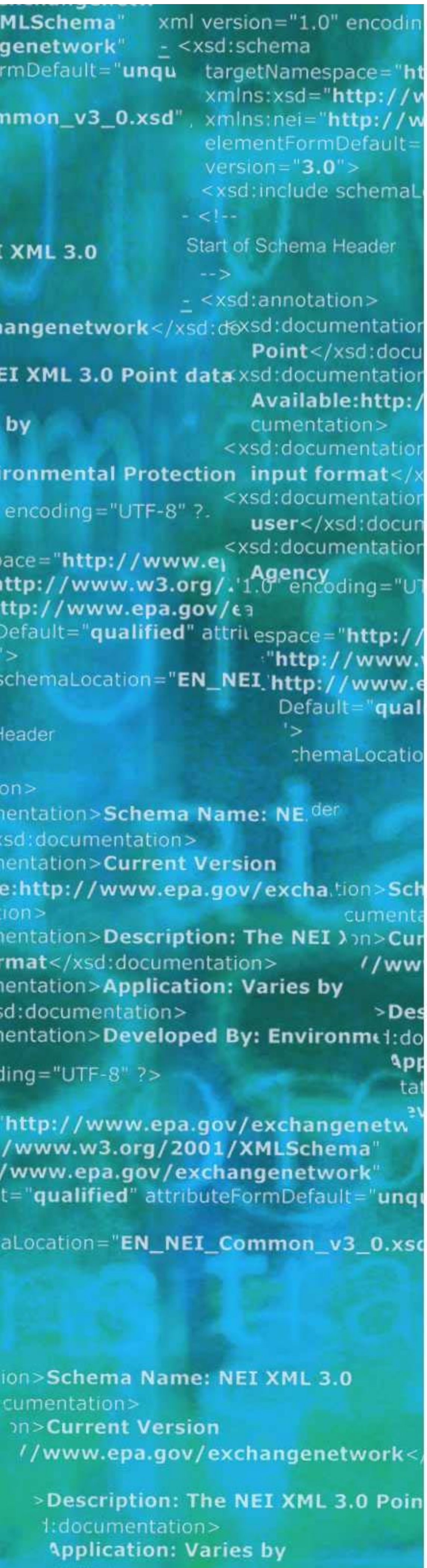

Node 2.0 FCD Compatibility Guideline Development

Approach and Methodology

Version: 1.0

Revision Date: 2/3/2009

Prepared for: Environmental Council of States (ECOS)

Prepared by: Windsor Solutions, Inc.

Version and Component Alignment

This document describes the approach and methodology used to develop compatibility guidelines for adapting legacy Exchange Network data exchanges to work with a Specification v2.0 nodes.

Document Version	Explanation (optional)
v1.0	Original Release

1 Introduction

With the release of the Node Specification v2.0, Exchange Network partners are beginning to plan the implementation of upgraded node software. While new data exchanges will be written to take advantage of the new features in Node v2.0, there are many legacy flows that need to continue to operate into the future. Until such time as all existing exchanges are rewritten to target the Node 2.0 platform, each must be adapted to operate in a Node v2.0 environment. The Node v2.0 Compatibility Guidelines were written to provide instructions for bridging legacy flows onto the Node v2.0 platform.

Note that no new functionality has been introduced to any exchange; the goal of the addenda is to provide a set of rules for implementing each exchange to a Specification v2.0-compliant node without changes to functionality. It is expected that future versions of each exchange will take advantage of the capabilities offered by the v2.0 Specification.

1.1 Background

The Node Specification v2.0, released in June 2008, extends the services and capabilities provided in the original v1.1 specification. Before this time, each FCD targeted Specification v1.1 nodes and, as such, does not supply the needed detail of how the exchange should be implemented on Specification v2.0-compliant nodes.

See the Exchange Network website located at http://www.exchangenetwork.net for more information on Node Specification v2.0, Node 2.0 Frequently Asked Questions, and other information relevant to the transition to the new specification.

1.2 Approach

The two specifications offer the same services (web service methods) such as Query, Solicit, and Submit; however, the details of each service have changed slightly in the new version. Since all data exchanges are built on the basic set of core services, the approach taken by this effort is to describe how each legacy data exchange must be modified to work with the specification changes at the service level.

The advantage to this approach is that it breaks down the changes in simple terms without special considerations of the exchange’s business processes. It also provides a consistent framework for describing the changes across exchanges, allowing each flow addendum to retain the same basic format.

2 Web Service Methods

This section describes the main differences between the services offered in the v1.1 and v2.0 specifications. For each method, the implications of adapting existing services to the new node specification are described.

2.1 Authenticate

The “Authenticate” method allows a node or node client to authenticate to the National Authentication and Authorization Service (NAAS). According to network policy, all Node 2.0 interactions must use the NAAS v3.0 endpoint for authentication and for validating tokens. Note that NAAS v2.0 and NAAS v3.0 share the same data store, so accounts do not need to be migrated.

2.2 Submit

Specification v2.0 introduces three new parameters to the Submit operation: *flowOperation* (required), *recipients* (optional), and *notificationURI* (optional). How each new parameter should be implemented for WQX v2.0 Submit operations is described below.

2.2.1 dataflow Parameter

The *dataflow* parameter existed in Specification v1.1 and is unchanged in v2.0. Typically the value provided in the *dataflow* parameter is a short acronym describing the type of data being exchanged. It is used to indicate the family of data and data services to which a message relates. For continuity, the value provided in the dataflow parameter is unchanged between the Specification v1.1 and v2.0 implementations for the purposes of porting legacy exchanges.

2.2.2 flowOperation Parameter

The *flowOperation* parameter is new for Specification v2.0. It is intended to allow a submitter to tell the receiving node how to process the file. Since the *flowOperation* value is required, the compatibility guidance will universally recommend that the value “default” be used, even though it cannot and will not be evaluated by the receiving node.

2.2.3 recipients Parameter

The *recipients* parameter is new for Specification v2.0. It is intended to allow a submitter to specify zero or more email addresses and/or node URLs to receive the submitted file. In essence, this informs the receiving node to relay the submitted file to some other node or person. The use of the *recipients* parameter is optional.

Since legacy exchange processing software was not designed to consider this parameter, exchange partners should not expect that a receiving node will act upon any *recipients* information if it is provided.

2.2.4 notificationURI Parameter

The *notificationURI* parameter replaces the *returnURL* parameter in Specification v1.1. While only present on the Solicit operation in Specification v1.1, the revised specification extends this parameter to the Submit operation. The purpose of this optional parameter is to allow the submitter to provide a set of zero or more email addresses and/or node URLs to which a notification may be sent when the receiving node changes the processing status of a submitted file (in the case of Submit) or data request (in the case of Solicit).

There is also a difference in how this information is used by the node in Specification v2.0. In the previous version, the receiving node can respond by submitting the resulting document(s) to the *returnURL* specified once processing is complete. In Specification v2.0, the node is required to send a notification to the sender any time the submission status changes. The intent is that the sender would wait for a notification that processing was complete and then download the resulting processing reports.

Legacy data exchanges are not designed to handle *notificationURI* information, so while Specification v2.0 requires nodes to support this, ported legacy flows will ignore *notificationURI* data if it is provided.

2.2.5 Exchange Network Header Usage

Since legacy flow processing software is not being upgraded to adapt legacy exchanges to the Node v2.0 Specification, there will not be any changes to the content of any XML documents, including changes to the Header. Exchanges that implement the Exchange Network Header v0.9 will continue to use it until such time as the exchange undergoes a full redevelopment, at which time Header v2.0 will be required.

2.2.6 Submit Response

Specification v1.1 only returned a Transaction ID to the submitter upon receiving a new submission. Specification v2.0 extends this to also return a status (from an enumerated list) and status detail text.

Nodes that support the receipt and processing of inbound WQX XML files must return a status and status text as required by the Specification v2.0. How the status is determined shall be a decision for the flow implementer. The node may simply return a status of “received” or may dynamically update the status based on immediate response from the flow processing software.

2.3 GetStatus

Specification v1.1 nodes only returned a simple status text string in response to a GetStatus request. Specification v2.0 extends this to also return a status (from an enumerated list), status detail text, and the Transaction ID being referenced.

It is assumed that the response to a GetStatus request will simply replicate the v1.1 response string (such as “received”) into the enumerated status response field and also the status detail text field.

2.4 Download

The behavior of the Download method is largely unchanged between Specification v1.1 and v2.0.

2.5 Query/Solicit

Specification v2.0 introduces two major changes to the Query operation: the introduction of a required “dataflow” parameter and the introduction of a strongly typed parameter list. The implication of each change is discussed below.

2.5.1 dataflow Parameter

Specification v1.1 only allowed for the request name to be provided by a data requestor when a Query or Solicit operation was performed. The unintended consequence of this was that all Query/Solicit operations must be uniquely named across all exchanges on the Exchange Network. This was remedied in Specification v2.0 by the introduction of the *dataflow* parameter, allowing for a request name to be further qualified with the name of a specific exchange. Now, many different flows can support a request named “GetFacilityByName”, for example. Furthermore, the version number has been appended to the dataflow name to gain processing efficiencies as described in section 2.2.1.

The *dataflow* parameter value used in Query and Solicit operations will be the same as is used for Submit and Download operations.

2.5.2 parameters Parameter

Specification v1.1 required the query/solicit parameter to be expressed as an unstructured list of values (array of strings). Specification v2.0 replaces this with a list of structured *parameter* objects, each with a specific name (required), data type (optional), and value. This allows for parameters to be provided in any order and to provide criteria in a much more explicit format.

For the development of compatibility guidelines, the only major change is to ensure that the parameter names are explicitly provided in the addenda. In all cases, the parameter names specified in the FCD were simply converted to UpperCamelCase to ensure a single value with no whitespace is used.