

Demonstrated Node Configuration for the Central Data Exchange Node

DRAFT

May 30, 2003

Task Order No.: T0002AJM038

Contract No.: GS00T99ALD0203



Environmental Information
exchange
Network

Abstract

The Environmental Protection Agency (EPA) selected Computer Sciences Corporation (CSC) as the primary contractor to build the Central Data Exchange (CDX) Node on the Environmental Information Exchange Network (Exchange Network). This Demonstrated Node Configuration (DNC) leverages the experience gained by CSC during the development of the CDX Node and provides installation and configuration instruction for reference and use by developers and system administrators building a node within the Exchange Network.



Table of Contents

1.0	PURPOSE	1
2.0	INTRODUCTION	2
2.1	Terminology	2
3.0	OVERVIEW OF THE DEMONSTRATED NODE CONFIGURATION	4
4.0	CDX NODE OVERVIEW	5
4.1	CDX Node Architecture	5
4.1.1	CDX Node Services	6
4.1.2	CDX Node Middleware.....	6
4.2	Hardware Requirements	7
4.3	Software Requirements	7
4.3.1	General Software Requirements	7
4.3.2	Tool Requirements.....	8
4.3.3	CDX Node ZIP File.....	8
5.0	PRE-INSTALLATION	9
5.1	Preparation for Software Installation	9
6.0	INSTALLATION AND CONFIGURATION INSTRUCTIONS	10
6.1	Software Installation and Configuration	10
6.1.1	Software Development Kit (SDK) Installation	10
6.1.2	WinZip Installation	11
6.1.3	Tomcat Installation	11
6.1.4	CDX Node Web Services Tier Installation	11
6.2	Software Testing	12
6.3	State Node Implementation	13
7.0	POST-INSTALLATION	15
8.0	NODE MANAGEMENT TOOLS	16
8.1	Node Logging Mechanisms	16
9.0	CONCLUSION	17



Table of Tables

Table 1. Terminology	3
Table 2. CDX Node Services.....	6

Table of Figures

Figure 1. Network Overview	4
Figure 2. CDX Network Node Architecture.....	5
Figure 3. Java Installation	10
Figure 4. Node Configuration.....	11
Figure 5. CDX Web Services	12
Figure 6. CDX Node WSDL	13
Figure 7. State Node Architecture.....	14



1.0 Purpose

This document describes the DNC implemented for EPA as part of the Network Node 1.0 project. This DNC is intended to serve as a primer for future Exchange Network participants. The goal is to decrease the time to market for participants by reducing complexity and costs. The strategy is to provide installation software with instructions to expedite the implementation and configuration of a Node. Instructions provided are based on the CDX Node configuration.

The Exchange Network provides a standard mechanism for data exchange. Participants in the Network Node 1.0 project include several State nodes and the EPA CDX Node. State nodes exchange data with each other according to the Network Node Functional Specification Version 1.0 and individual Trading Partner Agreements (TPA). They also similarly exchange data with CDX. The primary difference between State nodes and CDX is that CDX ultimately delivers and collects data from a series of separately governed and maintained EPA program areas (e.g., Facility Registry System [FRS]). In essence, it serves as a funnel to the corresponding EPA repository. As such, the CDX architecture is quite different from that of the typical State node. It contains several additional services (e.g., archival, transformation, distribution, etc.) that are not necessarily relevant for a State node. Therefore, this document focuses on the portion of CDX that can be best leveraged by new Exchange Network participants: the Web services tier.

The Web services tier encapsulates the receiving, sending, and parsing of Simple Object Access Protocol (SOAP) messages. This DNC includes a generalized solution for handling the Web services calls that can be used by any node on the Exchange Network. This solution allows future node developers to focus on the actual implementation of the node, which will always have proprietary components based on business logic and infrastructure, rather than the messaging layer. By using this document and the associated software, node developers will jumpstart the implementation process. They will be free to center on the additional infrastructure and database requirements that are specific to the node of interest.

In addition to the DNC, node implementers should be familiar with the following documents:

- Exchange Network Node Implementation Guide v1.0
- Network Exchange Protocol Version 1.0
- Network Node Functional Specification Version 1.0
- Network Security Guidelines and Recommendations
- Node Reference WSDL Version 1.0

These artifacts, which describe the goals and guidelines of the Network Node 1.0 project, can be found at www.exchangenetwork.net. See the *Exchange Network Node Implementation Guide v1.0* for essential steps for building a node.



2.0 Introduction

The EPA has established a single portal on the Web for environmental data entering the EPA - called the CDX. The CDX offers companies, States, Tribes, and other entities a faster, easier, more secure reporting option compared to previous alternatives. CDX provides built-in data quality checks, Web forms, standard file formats, and a common, user-friendly approach to reporting data across vastly different environmental programs. A cornerstone of EPA's e-government initiative, CDX currently accepts data for certain air, water, waste and toxics programs, and will gradually expand to support all Agency environmental reporting. Although its current focus is electronic, CDX will eventually incorporate a facility that centralizes paper data collections as well. CDX is part of a broader effort to integrate environmental data, reduce the burden of reporting, and improve data quality.

The goal of the Exchange Network is to foster standardization and information sharing. Common to all Exchange Network participants is the need to establish secure points of exchange or "nodes". CDX is EPA's "Agency Node." EPA, in collaboration with other Exchange Network members, has identified and prioritized Network dataflows for inclusion in the deployment and implementation of CDX and all nodes within the Exchange Network.

Members of the Network Node 1.0 project have developed demonstrated node configurations in order to assist prospective participants with the implementation activities. This document presents the hardware and software requirements along with the pre and post-installation activities. New Network participants can use this material along with the installation files to create a node.

2.1 Terminology

Table 1 defines common terms that are used throughout this document.

Term	Definition / Clarification
CDX	Central Data Exchange
CSC	Computer Sciences Corporation
DIME	Direct Internet Message Encapsulation
DMZ	Demilitarized Zone
DNC	Demonstrated Node Configuration
EJB	Enterprise JavaBeans
EPA	Environmental Protection Agency
Exchange Network	Environmental Information Exchange Network
FRS	Facility Registry System
HTTP	Hyper Text Transfer Protocol - The set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the Web
HTTPS	Hyper Text Transfer Protocol Secure Sockets



Term	Definition / Clarification
IP	Internet Protocol
IT	Information Technology
J2EE	Java 2 Enterprise Edition – Component-based Java architecture
JAR	Java Archive – Library of Java components
JDBC	Java Database Connectivity
JDK	Java Development Kit – Includes Java Virtual Machine that executes node implementation
JMS	Java Messaging Services
JRE	Java Runtime Environment
NAAS	Network Authentication and Authorization Services – The centralized Web services that provide user authentication and access control
Node	Participant on the Exchange Network
SDK	Software Development Kit
SOAP	Simple Object Access Protocol – Provides interoperability across operating systems
SSL	Secure Sockets Layer – Provides encryption for secure data exchanges
TPA	Trading Partner Agreement – Defines node exchanges between trading partners
UI	User Interface
URL	Uniform Resource Locator
WAR	Web Archive – Contains Web components such as servlets, JSPs, HTML, images, and JSP tag libraries. WAR files are deployed to the Web server
WSDL	Web Service Description Language - An XML-based language used to describe the available Web services
XML	Extensible Markup Language
XML Schema	XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content, and semantics of XML documents

Table 1. Terminology

3.0 Overview of the Demonstrated Node Configuration

The Exchange Network is a new approach for exchanging environmental data between EPA, States, and other partners that uses the Internet and standardized data formats. As illustrated in Figure 1, the Exchange Network consists of data exchanges between nodes or portals maintained individually by participating Partners (initially envisioned as State environmental departments and EPA). Once established, these data exchanges will replace and complement the traditional approach to information exchange that currently relies upon States feeding data directly to multiple EPA national data systems. Specifically, CDX will act as a funnel that allows Partners to feed data to the multiple EPA systems in a standard manner. Partner C (e.g., EPA) in Figure 1 represents the CDX Network Node.

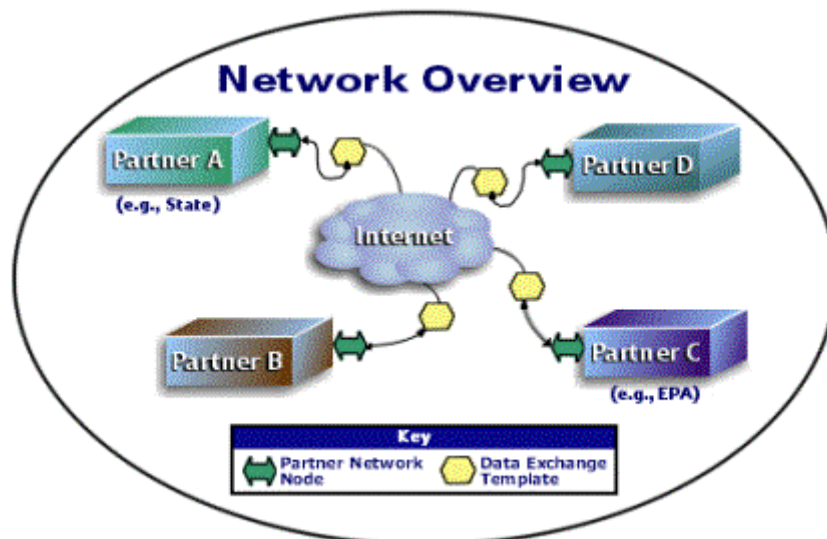


Figure 1. Network Overview

Each node on the Exchange Network will use Web services to exchange information. The core Web services (e.g., Submit(), Download(), etc.) will be based on the Network Node 1.0 Functional Specification, and will support standard interactions across nodes. The use of Internet standards (e.g., SOAP, WSDL, XML) enables hardware and software independent exchanges. This DNC describes the Web services tier for CDX.

4.0 CDX Node Overview

For background purposes, the next two sections briefly describe the entire CDX Node architecture. Additional details are documented in the *CDX System Design Document*, *Users Manual*, and *Administrators Manual*, which can be provided by the EPA. Recall that this DNC will provide the hardware, software, and tool requirements necessary to install and run the CDX Web services tier only.

4.1 CDX Node Architecture

The CDX Network Node is a Web services-based application that leverages a Web server, Web clients, and standard Internet protocols. Figure 2 shows the architecture of the system from a component standpoint. At the heart of the application framework is a single, unifying Java-based programming model for building the CDX Network Node. The Web services toolkit from Apache Axis is a key component serving as the preferred request handler and response mechanism, which includes industry standards such as SOAP, UDDI, and WSDL.

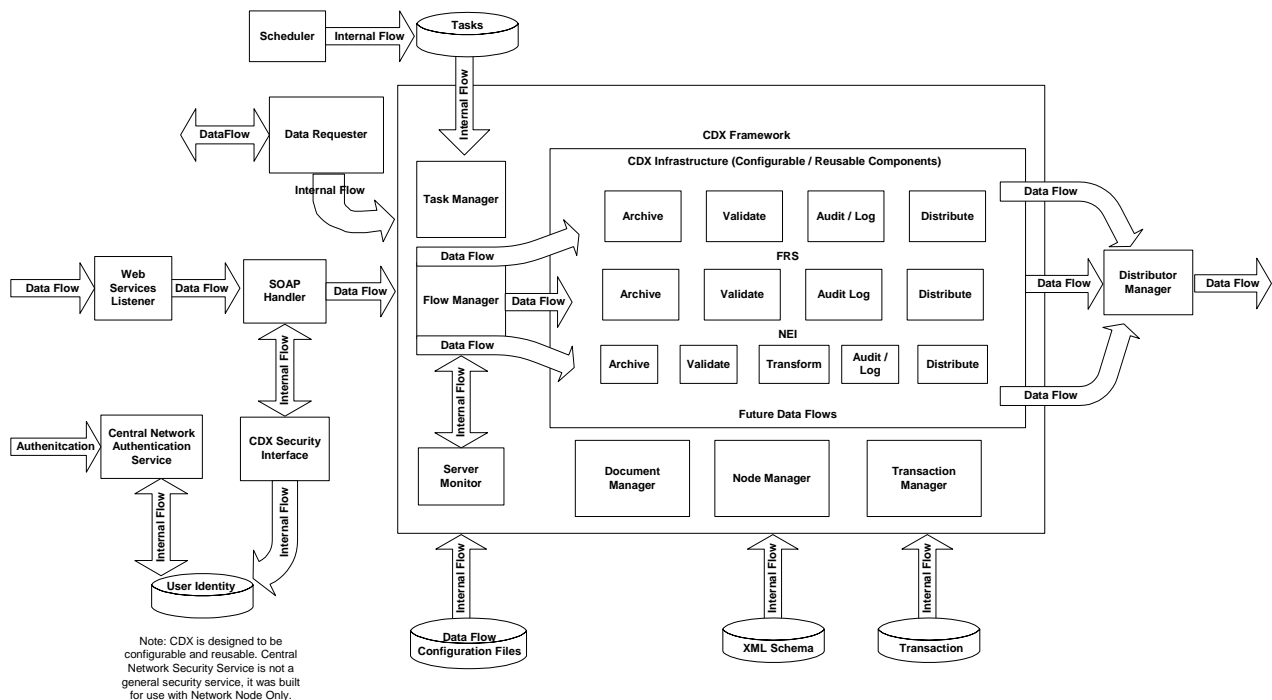


Figure 2. CDX Network Node Architecture

The CDX Network Node is a Java 2 Enterprise Edition (J2EE)-based, message-driven interactive system. It consists of the following: user interface (UI), Web and application server, SOAP, WSDL, Java Messaging Services (JMS), Enterprise JavaBeans (EJB), Java Database Community (JDBC), and database. The UI allows authorized users to schedule data exchanges via Web browser. The Web server, which handles the user requests from the scheduler, as well as other Network nodes, interacts with the Web services framework of the system to deliver available services to the Network. The Web services listener forwards all requests to the SOAP



Handler (e.g., Axis), which then parses the incoming SOAP request and translates it into a Java call. A series of other J2EE components perform the remainder of the overall service. Refer to Section 4.1.1 for a brief description of each component. These components connect to an Oracle database through a J2EE application server using JDBC.

4.1.1 CDX Node Services

Table 2 describes the services that are currently available within the CDX Network Node.

Service	Description
Archive	Provides the ability to manage, store, retrieve, and validate documents in various formats (XML, Flat, Bin, ZIP) in persistent data storage (Oracle Database)
Validate	Validates XML documents against a schema
Audit	Records each significant operation performed; provides the capabilities to track, search and manage all CDX activities
Log	Logs system-level CDX events primarily for debugging purposes
Distribute	Dispenses processed documents to participating dataflows (e.g., FRS)
Scheduler	Allows the CDX administrator to schedule and execute various tasks such as data submission and retrieval
Web Services Listener/SOAP Handler	Exposes CDX operations as Web services and translates incoming/outgoing SOAP requests/responses
Central Network Authentication Service	Provides the capability to Authenticate and Authorize (future) incoming and outgoing requests
Task Manager	Queries internal task table, and manages scheduled tasks
Document Manager (Archiver)	Manages storage and retrieval of the documents
Node Manager	Controls validation and management of Network nodes
Transaction Manager	Creates, validates, and manages CDX transactions, (i.e., the association of transactions with stored documents)
Server Monitor	Monitors server State and provides status

Table 2. CDX Node Services

4.1.2 CDX Node Middleware

The J2EE application server that hosts CDX is WebLogic Version 7.01 from BEA Systems, Inc. This serves as the application server and Web server. WebLogic was chosen due to the unique EPA requirements of CDX in terms of scale and diversity of transactions. WebLogic is very stable and has excellent support and proven scalability. This platform, which has full open standards compliance with J2EE support, is consistently one of the fastest to support the latest



specifications. With extensions for Web services development, security, and enterprise application integration, BEA provides industry-leading performance.

One of the primary advantages of a J2EE solution is that applications are portable across platforms. Although CDX is integrated with WebLogic, the Web services tier can be hosted as-is in any Java environment (Refer to section 4.3.1 for software requirements). In order to assist the widest audience, the Web services tier distribution has been generalized for deployment on any Java environment through the DNC. As such, the host Web server for the DNC will be the freely available Apache Tomcat. This Web server can host the Web services tier, parse incoming SOAP messages, and forward requests as desired by the State node.

4.2 Hardware Requirements

We recommend reviewing other DNCs for advice on hardware requirements for hosting a full State node. However, the minimum recommended hardware for the Apache Tomcat Web services tier is as follows:

Description	Minimum Requirements
Processor	450 MHz Intel Pentium – compatible CPU
Memory	128 MB of RAM
Disk Space	110 M hard disk space

Note that since the DNC is Java-based, it can run on a variety of platforms and operating systems.

Additional consideration may be required for load-balancing in the event of high Network traffic volume.

4.3 Software Requirements

4.3.1 General Software Requirements

The following software is required for the DNC implementation:

- JDK 1.3.x - This is not provided as part of the DNC distribution. It can be downloaded from: <http://java.sun.com/j2se/1.3/download.html>.
- WinZip - This is not provided as part of the DNC distribution. It can be acquired from: <http://www.winzip.com>.
- Apache Tomcat 4.0.6 - The Windows version of Tomcat is included in the DNC distribution. If it is needed for another platform see: <http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.6/bin/>.
- SSL Certificate - All operational nodes require SSL encryption. Node implementers need to acquire and install an SSL certificate. Instructions for configuring an SSL Certificate with Tomcat are available at: <http://jakarta.apache.org/tomcat/tomcat-4.0-doc/ssl-howto.html>.



4.3.2 Tool Requirements

No toolsets are required for the CDX implementation.

4.3.3 CDX Node ZIP File

In addition to this document, a zip file (`CDX_NODE_DNC.zip`) containing the node software, third party tools, Axis configuration files, and node system configuration files are provided.



5.0 Pre-Installation

This section outlines the activities that need to be considered before installing the node. These activities include:

- Determining any information that will be required before installing the node.
- Determining any Information Technology (IT) infrastructure or security issues that will need to be addressed or resolved as part of the node installation (i.e., permissions, firewall configuration settings, administration issues).

5.1 Preparation for Software Installation

The node Web services tier can be deployed on an Apache Tomcat Web server. The following conditions need to be validated before deployment:

- The Web services tier on which the State node will be deployed is accessible via the public Internet over a defined Uniform Resource Locator (URL), which either has a registered domain name address, or is defined by an Internet Protocol (IP) address and port number.
- The node server has access to the data source(s) that house the State EPA data.
- Network nodes should be hosted in a Demilitarized Zone (DMZ).
- It is strongly recommended to use 128-bit Secure Sockets Layer (SSL) on Apache Tomcat.



6.0 Installation and Configuration Instructions

This section outlines instructions for installation and configuration of the node.

These instructions include:

- Description of the software installation and configuration steps.
- Description of the software testing steps.
- Description of the how to implement the State node specific business logic.

6.1 Software Installation and Configuration

The software installation steps should be completed in the order listed below. They assume the installation occurs using the Windows operating system.

6.1.1 Software Development Kit (SDK) Installation

In order to run the CDX DNC, SDK 1.3.x must be installed on the machine (Note: JRE is NOT sufficient). If SDK 1.3.x is not installed perform the following steps:

- Download SDK from the following site: <http://java.sun.com/j2se/1.3/download.html>.
- Choose appropriate platform link and follow the installation.

Define the JAVA_HOME system variable. Go to Windows **Start->Settings->Control Panel** then click on **System**. Select the **Advanced** tab and click the **Environment Variables...** button. Add a **New...** variable that points to the location where Java SDK is installed as in Figure 3.

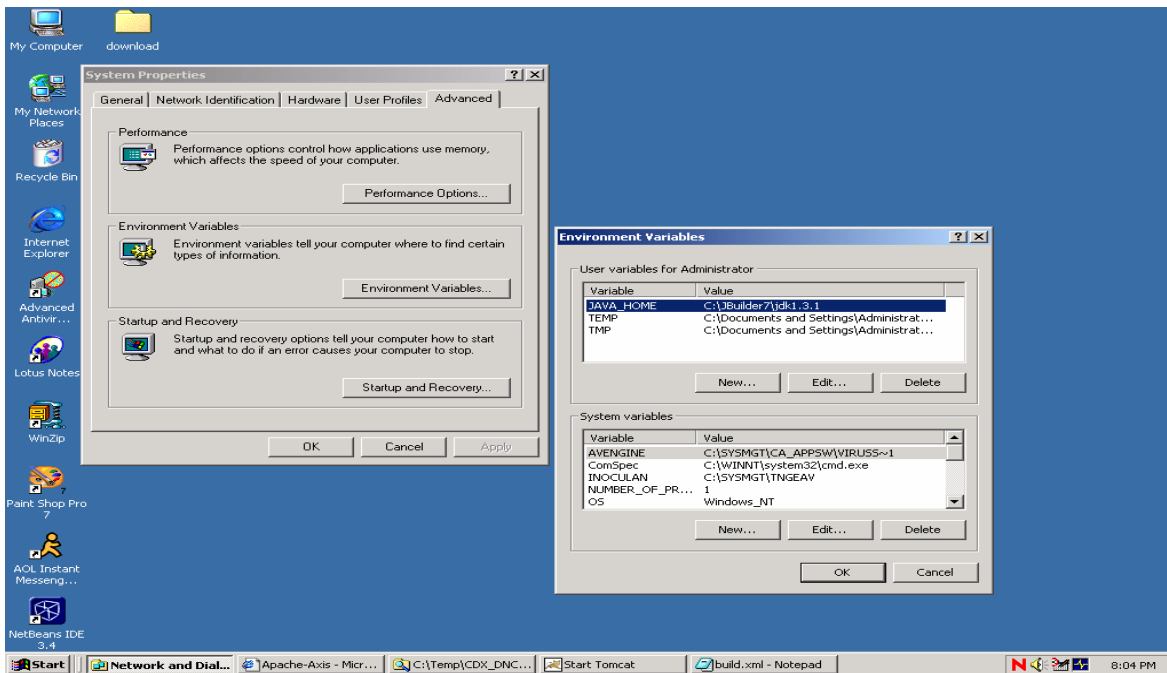


Figure 3. Java Installation



6.1.2 WinZip Installation

In order to unzip the CDX DNC, WinZip must be installed on the machine. Download WinZip from the following site: <http://www.winzip.com> and follow the installation instructions.

6.1.3 Tomcat Installation

Unzip the CDX_NODE_DNC.zip file, choose C:\ as the root directory. The following directory structure will be created: C:\DNC

```
| -Tomcat_Dist  
| -CDX_DNC
```

Install the Tomcat Web server. Run the jakarta-tomcat-4.0.6.exe file located under C:\DNC\Tomcat_Dist\directory. When running the Tomcat installer, perform the following steps:

- Click on **OK** if SDK was found on the machine. (Otherwise follow step 1).
- Agree to the license terms.
- On the Installation Checklist window, check only two options: Tomcat 4.0 and Tomcat 4.0 Start Menu Group.
- Accept Installation in the following location: C:\DNC\Tomcat.
- Install the Tomcat and press **Close**.

6.1.4 CDX Node Web Services Tier Installation

Build CDX_DNC WAR files and complete configuration.

- Go to C:\DNC\CDX_DNC directory and edit the build.properties file:
 - Modify the deploy.host variable to contain the IP address of the machine that the node will be running on. Do not change the default port of 8080. See Figure 4.

```
# =====  
# Build properties file for environment-specific settings.  
# Please modify this file instead of the build.xml script!  
# =====  
# Project root directory  
#  
proj.root=./  
#change the deploy.host value to something reachable from the internet  
#if you want this node to be accessible. This will become part of your cdx.wsd1 file, so other nodes  
#will know how to reach you.  
deploy.host=127.0.0.1:8080
```

Figure 4. Node Configuration

- Run the build.cmd file.
- Copy axis.war and schema.war from C:\DNC\CDX_DNC\dist to C:\DNC\Tomcat\webapps. Note that when Tomcat starts, these WAR files are automatically extracted to disk under C:\DNC\Tomcat\webapps\axis and C:\DNC\Tomcat\webapps\schema. These subdirectories must be manually deleted when subsequently copying the WAR files for redeployment. Remember to restart Tomcat after copying the WAR files.
- Copy C:\DNC\CDX_DNC\conf\properties\user_properties.xml to C:\DNC\Tomcat directory.

6.2 Software Testing

Run the Tomcat Web server. Select **Start->Programs->Apache Tomcat->Start Tomcat**.

Verify the available Web services and WSDL.

- Run a Web browser.
- Go to the URL: <http://localhost:8080/axis/servlet/AxisServlet>. A window similar to Figure 5 should appear.

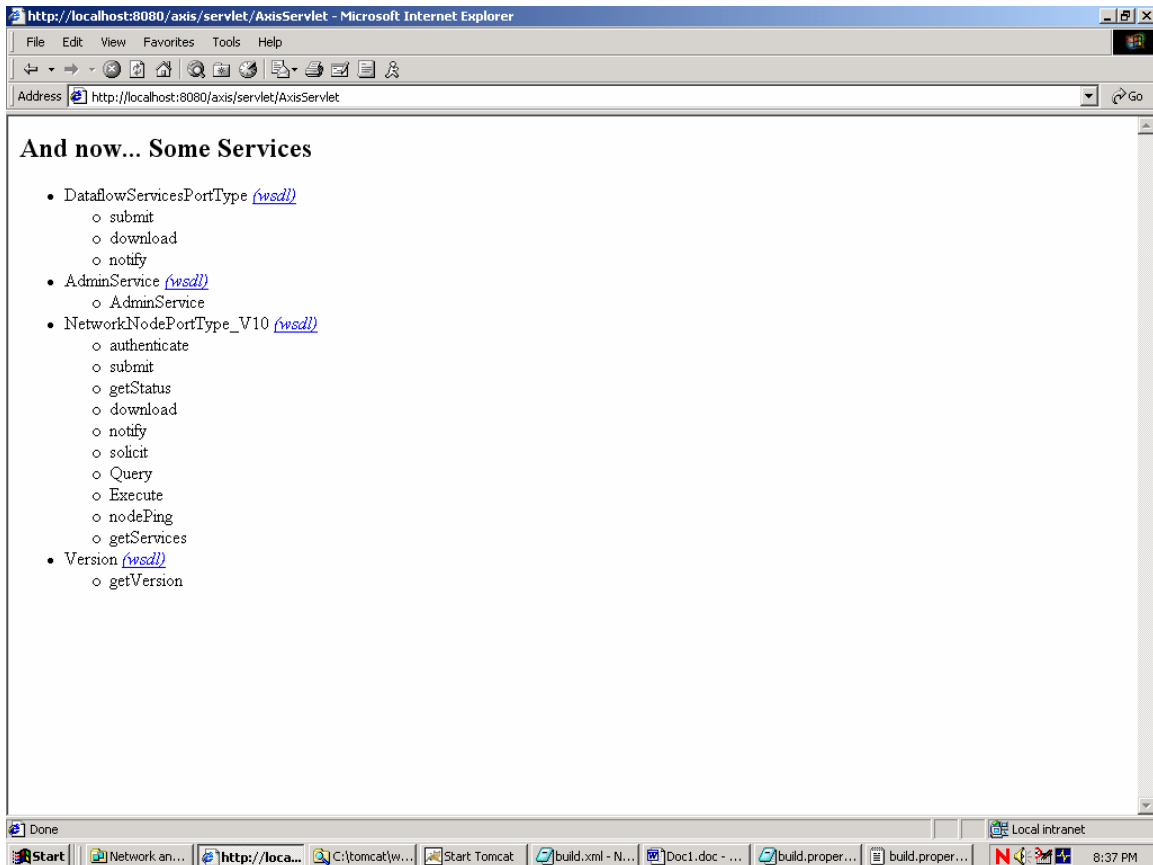


Figure 5. CDX Web Services

- In order to display the WSDL file, specify the following URL: <http://localhost:8080/schema/v1.0/cdx.wsdl>. A window similar to Figure 6 should appear. Verify that the IP address at the bottom is correct: `<soap:address location="http://<IP>:8080/axis/services/NetworkNodePortType_V10" />`.


```

<?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="cdx" targetNamespace="http://www.ExchangeNetwork.net/schema/v1.0/node.wsdl"
  xmlns:tns="http://www.ExchangeNetwork.net/schema/v1.0/node.wsdl" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:typens="http://www.ExchangeNetwork.net/schema/v1.0/node.xsd" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:dime="http://schemas.xmlsoap.org/ws/2002/04/dime/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/">
- <types>
- <schema targetNamespace="http://www.ExchangeNetwork.net/schema/v1.0/node.xsd"
  xmlns="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:SOAP-
  ENC="http://schemas.xmlsoap.org/soap/encoding/" elementFormDefault="qualified">
- <complexType name="NodeDocument">
- <sequence>
  <element name="name" type="xsd:string" default="Your document name" />
  <element name="type" type="xsd:string" />
  <element name="content" type="xsd:base64Binary" />
</sequence>
</complexType>
- <complexType name="ArrayOfstring">
- <complexContent>
  <restriction base="SOAP-ENC:Array">
    <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="string[]" />
  </restriction>
</complexContent>
</complexType>
- <complexType name="ArrayOfDoc">
- <complexContent>
  <restriction base="SOAP-ENC:Array">
    <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="typens:NodeDocument[]" />
  </restriction>
</complexContent>
</complexType>
- <complexType name="QueryResult">
- <sequence>
  <any />
</sequence>
</complexType>

```

Figure 6. CDX Node WSDL

If each step occurred as described, the software installation and testing have been completed successfully.

6.3 State Node Implementation

After installing, configuring, and testing the software, the State node specific business logic must be added. The architecture of this node is displayed in Figure 7. Note that the Web services tier software that is being hosted by the Apache Tomcat Web server consists of the following components:

- HTTP - This is part of the Apache Tomcat Web server. By default, the HTTP listener is listening on port 8080. Refer to Section 4.3.1 for instructions for upgrading to SSL for secure data exchanges.
- Apache Axis Web Services Toolkit - This toolkit translates the incoming SOAP messages into Java requests. The associated Java Archive (JAR) files are included in `axis.war`.
- Web Services Adapter - This is a set of Java classes that provide the binding to the State node plug-in. The adapter receives inbound Java requests from Axis and redirects them to the State node plug-in. The adapter automatically wraps and unwraps the DIME attachments in order to abstract this complexity away from the State node plug-in. For convenience, the Web services adapter also contains access methods for outbound Network requests. These outbound requests include each of the Exchange Network Web

services (e.g., Submit(), Download(), etc.). The Authenticate() Web service makes calls to the Network Authentication and Authorization Services (NAAS) for user authentication. This allows the State node plug-in to make outbound calls without building its own Web services interface. See the `C:\DNC\CDX_DNC\client\UserManual.doc` for instructions and an example client for performing outbound calls. The adapter components are included in the `cdx_ws.jar` within `axis.war`.

- State Node Plug-in – This is the class that will need to be rewritten by the State node developers. By default, it contains a simple implementation that allows the State node to pass the Automated Test tool validation (See Section 7). Essentially, stubs are provided for each method in the *Network Node Functional Specification Version 1.0*. These methods must be replaced with State node business logic. This typically involves JDBC calls to the State node database. The precise name of this class is:

`gov.epa.state.sample.axis.DNCSampleImpl`

- After adding the implementation, simply repeat the steps in Section 6.1.4. Note that if additional classes are required, place these in the `C:\DNC\CDX_DNC\src` directory prior to running `build.cmd`. If additional JAR files are needed, place them in `C:\CDX_DNC\lib\commons`.

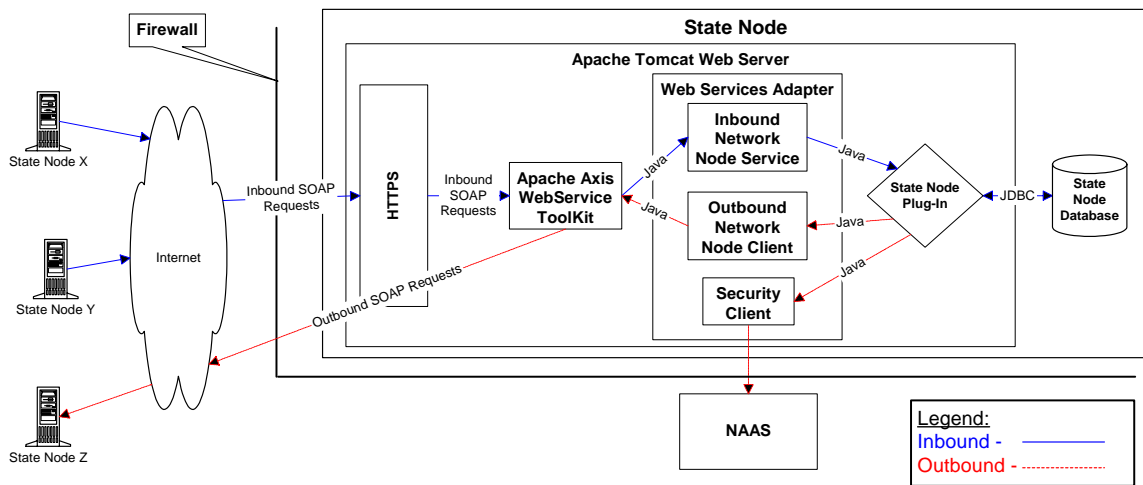


Figure 7. State Node Architecture



7.0 Post-Installation

Note: The tool described below is being replaced by more rigorous tool. The link to it will be made available soon on ExchangeNetwork website. (contact CSC if you have questions)

This chapter outlines the activities that need to be performed in order to test the node and communicate with other nodes on the Exchange Network.

CSC has provided a node test tool available over the Internet at <http://epacdxnode.csc.com/>, which allows node testing in one of two (2) modes:

- Manual testing – By entering the address of a WSDL file, represented as a URL, the test tool provides an entry form that allows each node service to be tested with user-provided input and allows the viewing of both the request and response SOAP messages.
- Automated testing – By entering the address of a WSDL file, represented as a URL, the test tool runs a set of pre-defined tests against each node service, returning either a “Passed” or “Failed” indication. It also allows the viewing of both the request and response SOAP messages.

In order to communicate with other nodes on the Network, it will be necessary to have the node address, as well as a client application which can generate SOAP request messages based on the nodes WSDL file.

The location of the WSDL file required to perform testing against this DNC node is <http://<IP>/schema/v1.0/cdx.wsdl>. The <IP> should be replaced by the IP address of the machine where the node will be running.



Node Management Tools

This section outlines activities necessary to manage and maintain the node on an ongoing basis.

7.1 Node Logging Mechanisms

- The CDX Network Node generates an entry in a log file for each operation that it performs (e.g., each time a service is requested).
 - `<IP>_access_log.<date>.txt` is the access log file located in `C:\DNC\Tomcat\logs`.
 - `<IP>_log.<date>.txt` is the transaction log file located in `C:\DNC\Tomcat\logs`.



8.0 Conclusion

This document outlined the steps required to setup the Web services tier for a Network node. States that wish to adopt a similar model are encouraged to use this document as an aid in their implementation efforts.

- This installation was based on the inclusion of a generic Java-based infrastructure. Note that the Apache Tomcat solution can be used to host many node implementations. If full transaction management is required, a more robust application server should be considered (e.g., WebLogic, JBoss, WebSphere) to replace the Apache Tomcat Web server. Any J2EE-compliant application server could host the Web services tier that is provided in this DNC.

Those with an interest in understanding more about this node configuration model are encouraged to contact the Network Steering Board (refer to the *Contacts* page at www.exchangenetwork.net).