

## **CCDWS**

### Common Communication Driver for Web Services Test Plan

## Table of Contents

1. Introduction.....	3
2. Related Documents .....	3
3. Testing Configuration .....	3
3.1. Operating Environment.....	3
3.2. Test Data .....	3
4. Test Cases.....	3

## Revision History

Date	Revision	CCDWS Version	Changes
2011-10-25	1.0	0.9.4 (beta)	❑ Initial revision
2012-05-30	1.1	0.9.6 (beta)	<ul style="list-style-type: none"> <li>❑ Must set DetectParseErrors=1 in CCDWS-0050 as default is now 0 (do not detect parse errors).</li> <li>❑ Added additional tests to handle WS-Addressing errors and retries.</li> <li>❑ Updated tests to reflect additional retry conditions (such as connection failures).</li> </ul>
2012-05-31	1.2	0.9.6 (beta)	❑ Added Prefix test CCDWS-0150
2012-06-05	1.3	0.9.6 (beta)	❑ Added non-default installation location test CCDWS-0160
2014-0922	1.4	1.0.1.0	<ul style="list-style-type: none"> <li>❑ Changed CCDWS-0100 to reflect new certificate store handling.</li> <li>❑ Changed CCDWS-0110 from use of invalid certificate to certificate that is included in the CRL.</li> </ul>

# 1. Introduction

This document provides a test plan for the Common Communications Driver for Web Services (CCDWS), sponsored by the Health Industry Electronic Commerce (HIEC) associations, and designed by Harder Software. CCDWS is designed to act as a communications interface between dental offices and healthcare insurers.

CCDWS is designed to run as a service application, polling for input request files. Upon discovery of an input file it will perform all tasks necessary to submit the claim and receive the adjudication response or acknowledgement in real time. It will operate in a parallel with the original CCD application, so that both carriers that accept dialup claims and those that accept web service claims can be supported simultaneously without modification to the dental software.

## 2. Related Documents

*CCDWS Requirements* was authored by Alberta Blue Cross to provide the requirements and overall parameters for development of CCDWS.

*CCDWS – Technical Design* provides information on design and programming of the CCDWS application. This document was authored by Harder Software.

*CCDWS – Technical Reference* can be used by software vendors to configure and troubleshoot CCDWS when integrated with their dental or other practice software. This document was authored by Harder Software.

*Point of Service (PoS) Application Services Transport Specification, Draft 1.9* defines much of the underlying transport specific specifications used for SOAP transactions. This document is authored by HIEC.

## 3. Testing Configuration

### 3.1. Operating Environment

CCDWS is currently compiled to execute in recent versions of 32 or 64-bit Windows, including Windows XP, Windows Vista, and Windows 7. It can be tested in any of these environments.

### 3.2. Test Data

Adjudication engines are particular about the content of the requests that are received. This test plan assumes that a set of data are published and available for testing. Within the test cases, exceptional conditions related to the data are described. Complete test data instances are not provided.

## 4. Test Cases

As additional test cases become available, and may be inserted at any point in the document, each case is provided a unique value rather than using an automated numbering scheme.

Unless specified, the log level should be configured to 3 (warning) or greater for referencing logged warnings and errors.

When configuration file changes are required, the CCDWS service must be restarted for them take effect.

<b>Test Case</b>	CCDWS-0010
<b>Description</b>	CCDWS must be capable of coexisting with existing CCD installation
<b>Configuration</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Ensure both CCDWS and CCD are running</li> <li><input type="checkbox"/> Ensure that the CCD network location corresponding to the CCDWS destination is disabled (it may have been automatically set to XXX_DISABLED) during the CCDWS installation).</li> <li><input type="checkbox"/> Create a valid input file for a configured CCD network.</li> <li><input type="checkbox"/> Create a valid input file for a configured CCDWS destination (e.g. ABC), while the CCD is still reading and submitting its input.</li> </ul>
<b>Input</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> CCD input contains any request that will result in an expected response</li> <li><input type="checkbox"/> CCDWS input contains any request that will result in an expected response</li> </ul>
<b>Expected Results</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> CCDWS output should be a valid response for its input</li> <li><input type="checkbox"/> CCD output should be a valid response for its input</li> </ul>

<b>Test Case</b>	CCDWS-0020
<b>Description</b>	Server is not accepting connections, or Endpoint host or port is incorrect, resulting in a connection attempt retry.
<b>Configuration</b>	<input type="checkbox"/> Change the port number of the EndPoint configuration entry for the destination to a value that not monitored.
<b>Input</b>	<input type="checkbox"/> Contains any request
<b>Expected Results</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> CCDWS will attempt to connect, fail, then retry once as per the default value of MaxRetries configuration option, resulting in replication of the error.</li> <li><input type="checkbox"/> Output status value is 1026 (No answer)</li> <li><input type="checkbox"/> Message in log file is "Error (7028): SOAP: A connection error occurred."</li> <li><input type="checkbox"/> Message in log file is "SOAP fault message: connect failed in tcp_connect()."</li> </ul>

<b>Test Case</b>	CCDWS-0030
<b>Description</b>	Endpoint path incorrect.
<b>Configuration</b>	<input type="checkbox"/> Change path of the destination to some invalid value
<b>Input</b>	<input type="checkbox"/> Contains any request
<b>Expected Results</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Output status value is 1001 (General Error)</li> <li><input type="checkbox"/> Message in log file is "Error (7404): HTTP: 404 Not Found."</li> </ul>

<b>Test Case</b>	CCDWS-0040
<b>Description</b>	Action is incorrect
<b>Configuration</b>	<input type="checkbox"/> Add (or change) the Action entry in the configuration file to an invalid value. Note that if it doesn't exist the default value will be used.
<b>Input</b>	<input type="checkbox"/> Contains any request

<b>Expected Results</b>	<ul style="list-style-type: none"> <li>❑ Output status value is 1001 (General Error)</li> <li>❑ Message in log file contains "SOAP fault message : " followed by a fault, determined by the actual server fault returned (may vary).</li> <li>❑ Message in log file is "Error (7001): SOAP: The service returned a client fault." (may vary)</li> </ul>
-------------------------	---

<b>Test Case</b>	CCDWS-0050
<b>Description</b>	CDAnet input payload contains invalid fields
<b>Configuration</b>	<ul style="list-style-type: none"> <li>❑ Standard configuration for sending a CDAnet message to a destination</li> <li>❑ DetectParseErrors is set to 1</li> </ul>
<b>Input</b>	❑ A CDAnet message, either with one or more invalid fields as defined by the CDAnet standard
<b>Expected Results</b>	<ul style="list-style-type: none"> <li>❑ Output status value is 1043 (Invalid characters)</li> <li>❑ Message in log file is "Error (6012): Payload parse error."</li> <li>❑ Each detected field error will also be recorded beneath a line entitled "Input request errors."</li> </ul>

<b>Test Case</b>	CCDWS-0060
<b>Description</b>	CDAnet input payload is invalid
<b>Configuration</b>	<ul style="list-style-type: none"> <li>❑ Standard configuration for sending a CDAnet message to a destination</li> <li>❑ Set DetectParseErrors=1 in main section</li> </ul>
<b>Input</b>	❑ In invalid CDAnet request, such as CPhA3 or other.
<b>Expected Results</b>	<ul style="list-style-type: none"> <li>❑ Output status value is 1043 (Invalid characters)</li> <li>❑ Message in log file is "Error (6012): Payload parse error."</li> <li>❑ Message in log file contains the message "Invalid payload."</li> </ul>

<b>Test Case</b>	CCDWS-0070
<b>Description</b>	Parse error reporting can be ignored in case where variation from standard is required.
<b>Configuration</b>	<ul style="list-style-type: none"> <li>❑ Standard CDAnet configuration</li> <li>❑ Set DetectParseErrors=0 in main section or remove option completely as this is the default.</li> </ul>
<b>Input</b>	❑ A CDAnet message, either with one or more invalid fields as defined by the CDAnet standard
<b>Expected Results</b>	❑ Successful transaction (response may be rejection due to invalid fields)

<b>Test Case</b>	CCDWS-0080
<b>Description</b>	Detect invalid remote certificate
<b>Configuration</b>	❑ Change destination EndPoint to be remote website not signed by CA certificate (e.g. <a href="https://www.hardersoft.com">https://www.hardersoft.com</a> )
<b>Input</b>	❑ Any valid request for the destination

<b>Expected Results</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Output status value is 1045 (Server disconnect)</li> <li><input type="checkbox"/> Message in log file is "Error (7030): SOAP: An SSL error occurred."</li> <li><input type="checkbox"/> Message in log file begins with "SSL verify error" followed by certificate details</li> <li><input type="checkbox"/> Message in log file is "SOAP fault message: SSL_connect error in tcp_connect()."</li> </ul>
-------------------------	--

<b>Test Case</b>	CCDWS-0090
<b>Description</b>	Ignore invalid remote certificate, as workaround for certificate problems
<b>Configuration</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Change destination EndPoint to be remote website not signed by CA certificate (e.g. https://www.hardersoft.com)</li> <li><input type="checkbox"/> Add SSLVerify=0 in destination section</li> </ul>
<b>Input</b>	<input type="checkbox"/> Any valid request for the destination
<b>Expected Results</b>	<input type="checkbox"/> Successful transaction

<b>Test Case</b>	CCDWS-0100
<b>Description</b>	Client certificate is not accessible.
<b>Configuration</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Modify the input file so that the CDA provider number (field B01) and/or provider office number (field B02) does not match any certificates installed in the certificate store. Specifically, the contents of the Subject field should not contain an ID OID that ends in the provider number, and/or an ID SITE OID should not contain a SITE ID OID that ends in the provider office number.</li> </ul>
<b>Input</b>	<input type="checkbox"/> Any valid request for the destination
<b>Expected Results</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Output status value is 1045 (Server disconnect)</li> <li><input type="checkbox"/> Message in log file contains a message of the form "Could not find certificate for provider ID xxxxxxxxx, and office id yyyy".</li> <li><input type="checkbox"/> CCDWS still attempts to make the connection.</li> <li><input type="checkbox"/> Message in log file is "Error (7030): SOAP: An SSL error occurred", and prior to that there will be a message indicating a handshake error.</li> </ul>

<b>Test Case</b>	CCDWS-0110
<b>Description</b>	Client certificate is in CRL
<b>Configuration</b>	<input type="checkbox"/> Install and use a client certificate that is included in a test CRL. CDAnet should be able to provide such a certificate
<b>Input</b>	<input type="checkbox"/> Any valid request for the destination
<b>Expected Results</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Output status value is 1001 (General Error)</li> <li><input type="checkbox"/> Message in log file should indicate certificate is not valid.</li> <li><input type="checkbox"/> Note: this condition was not available for testing at the time of writing, so actual SSL message is currently unknown.</li> </ul>

<b>Test Case</b>	CCDWS-0120
<b>Description</b>	WS-Addressing not enabled, resulting in invalid (incomplete) SOAP header information.

<b>Configuration</b>	<ul style="list-style-type: none"> <li>❑ In the appropriate destination section in the configuration file, set the following option: WS-Addressing=0</li> </ul>
<b>Input</b>	<ul style="list-style-type: none"> <li>❑ Any valid request for the destination</li> </ul>
<b>Expected Results</b>	<ul style="list-style-type: none"> <li>❑ Results may vary depending on endpoint</li> <li>❑ Output status value is 1001 (General Error)</li> <li>❑ Message in the log file contains "SOAP fault actor: Client"</li> <li>❑ Additional SOAP fault informational messages should occur in the log file indicating the structure is missing header information or is otherwise invalid.</li> <li>❑ Message in log file is "Error (7012): An exception raised by the service."</li> </ul>

<b>Test Case</b>	CCDWS-0130
<b>Description</b>	Read timeout results in single retry as per default number or retries.
<b>Configuration</b>	<ul style="list-style-type: none"> <li>❑ In the configuration file, set ReadWriteTimeout=1 to force a timeout that CCDWS will attempt to recover from.</li> <li>❑ Note that if the server responds within 1 second that this test will fail (i.e. the response will be received). In this case it may be necessary to test with another server.</li> </ul>
<b>Input</b>	<ul style="list-style-type: none"> <li>❑ Any valid request for the destination</li> </ul>
<b>Expected Results</b>	<ul style="list-style-type: none"> <li>❑ CCDWS will send a message, receive a timeout, and resend once as per the default value of MaxRetries configuration option, resulting in replication of the error.</li> <li>❑ Output status value is 1042 (Server timeout)</li> <li>❑ Message in the log file contains "SOAP fault actor: Client"</li> <li>❑ Message in the log file contains "SOAP fault message: End of file or no input: Operation interrupted or time out..."</li> <li>❑ Message in log file is "Error (7098): SOAP: Read timeout."</li> </ul>

<b>Test Case</b>	CCDWS-0140
<b>Description</b>	Test the MaxRetries option greater than 1 to see that the number of retries is configurable.
<b>Configuration</b>	<ul style="list-style-type: none"> <li>❑ Change the port number of the EndPoint configuration entry for the destination to a value that not monitored.</li> <li>❑ Set MaxTries=2 in the same destination</li> </ul>
<b>Input</b>	<ul style="list-style-type: none"> <li>❑ Any valid request for the destination</li> </ul>
<b>Expected Results</b>	<ul style="list-style-type: none"> <li>❑ CCDWS will attempt to connect, fail, then retry twice as per the default value of MaxRetries configuration option, resulting in three instances of the error in the log file.</li> <li>❑ Output status value is 1026 (No answer)</li> <li>❑ Message in log file is "Error (7028): SOAP: A connection error occurred."</li> <li>❑ Message in log file is "SOAP fault message: connect failed in tcp_connect()."</li> </ul>

<b>Test Case</b>	CCDWS-0150
------------------	------------



<b>Description</b>	Test the Prefix configuration option to automatically populate the CDAnet A01 Transaction Prefix field when submitting requests.
<b>Configuration</b>	<input type="checkbox"/> Add a Prefix=<Prefix> option to the Destination section in the configuration file, where <Prefix> can be any string up to 12 characters in length. Some destinations will require specific prefix values.
<b>Input</b>	<input type="checkbox"/> Any valid dental request for the destination, with the A01 field either empty or populated with some other value than defined in the configuration file.
<b>Expected Results</b>	<input type="checkbox"/> CCDWS will replace anything in the Transaction Prefix field A01 with the value specified in the configuration file before sending, regardless of whether it is empty or not. <input type="checkbox"/> CCDWS output should be a valid response for its input

<b>Test Case</b>	CCDWS-0160
<b>Description</b>	Test installation within non-default directory.
<b>Configuration</b>	<input type="checkbox"/> Install CCDWS either manually or with automated installation application. <input type="checkbox"/> Ensure path is not c:\ccd <input type="checkbox"/> Ensure drive is local to the host, not a network drive
<b>Input</b>	<input type="checkbox"/> Any valid dental request for the destination
<b>Expected Results</b>	<input type="checkbox"/> CCDWS should pick up the request <input type="checkbox"/> CCDWS output should be a valid response for its input

<b>Test Case</b>	CCDWS-0170
<b>Description</b>	Test download of CRL
<b>Configuration</b>	<input type="checkbox"/> Remove any *.crl file located in the main CCD directory
<b>Input</b>	<input type="checkbox"/> Any valid dental request for the destination
<b>Expected Results</b>	<input type="checkbox"/> CCDWS should download the CRL from the default location, verified by looking for a *.crl file in the main CCD directory upon completion of the transaction.

<b>Test Case</b>	CCDWS-0180
<b>Description</b>	Test download of CRL from secondary location
<b>Configuration</b>	<input type="checkbox"/> Remove any *.crl file located in the main CCD directory <input type="checkbox"/> Add an entry in the [Main] section of the configuration file: CrlDistributionPoint=
<b>Input</b>	<input type="checkbox"/> Any valid dental request for the destination
<b>Expected Results</b>	<input type="checkbox"/> CCDWS should download the CRL from the default secondary location, verified by looking for a *.crl file in the main CCD directory upon completion of the transaction.