Australian Government
Australian Digital Health Agency

Clinical Package Validator
User and Configuration Guide

22 December 2020 v3.2

Approved for external use

Document ID: DH-3350:2020

Australian Digital Health Agency ABN 84 425 496 912, Level 25, 175 Liverpool Street, Sydney, NSW 2000
Telephone 1300 901 001 or email help@digitalhealth.gov.au
www.digitalhealth.gov.au

Acknowledgements

Council of Australian Governments
The Australian Digital Health Agency is jointly funded by the Australian Government and all state and territory governments.

Document information

Key information

Owner Director, Connectivity & Informatics

Contact for enquiries Australian Digital Health Agency Help Centre
Phone 1300 901 001
Email help@digitalhealth.gov.au

Product or document version history

Product or document version	Date	Release comments
2.0	22 Aug 2014	Major release.
2.1	20 Jan 205	Feature and usability enhancements.
2.2	--	Not published.
2.3	17 Jul 2015	Software enhancements (see release note for details).
2.4	20 May 2016	Software enhancements (see release note for details).
2.5	29 May 2017	Software enhancements (see release note for details).
2.6.1	05 Oct 2017	Software enhancements (see release note for details).
2.7	16 Feb 2018	Software enhancements (see release note for details).
2.8	20 Jan 2020	Software enhancements (see release note for details).
2.9 – 3.1	-	Not published.
3.2	08 Dec 2020	Configuration guide moved from Installation and Configuration Guide to this document; Information related to HL7v2 validation added.

Table of contents

1 Introduction

1.1 Purpose

This document is a guide for developers and testers ("users") who use the Clinical Package Validator ("the Validator") to test whether their healthcare software system is producing messages, clinical packages and documents that conform to the relevant specifications for the messages and document types.

To the extent users wish to use the Validator to test conformance, users should not rely on the Validator alone to determine conformance, and independent testing will be required. The Validator is a tool to assist users only. The Validator does not test all conformance specifications and users should carefully read the *Clinical Package Validator Product Data Sheet*. For further guidance, contact the Agency Help Centre on 1300 901 001 or help@digitalhealth.gov.au.

1.2 Intended audience

This document is intended for:

- testers who use the Validator to assess whether their healthcare software system produces clinical documents/messages and clinical packages that conform to the relevant specifications; and
- developers who use the Validator to provide quick feedback for software development.

1.3 Scope

This document describes the use of the Validator to test both CDA and HL7v2 payloads

- whether the syntax and structure of a clinical document / message are conformant with specifications, and
- whether codes within a clinical document / message can be found in code systems supported by the Validator.

1.4 Overview

The Validator is a tool for testing whether a healthcare software system is producing certain (but not all) clinical documents that conform to the relevant document CDA implementation guide, structured content specification, and the My Health Record conformance profile specification.

In addition, the Validator tests certain (but not all) codes in clinical documents, and tests clinical packages to determine whether they conform to the packaging specifications.

The tool also tests HL7 v2 Diagnostic and Referral Messages that meet the Australian Localisation based on the work done by the Orders and Observations Working Group [HL7OOWG].

The Validator does not test all conformance specifications and users should carefully read the *Clinical Package Validator Product Data Sheet* to determine what the Validator cannot be used for, and for further guidance contact the Agency on 1300 901 001 or help@digitalhealth.gov.au.

2 Tests performed by the Validator

The Validator performs the following tests:

- package validation;
- template validation;
- HL7 v2 message validation for specific message types
- XDS metadata validation.

It can be used to assist a vendor testing the following types of files:

- a clinical document;
- a clinical package containing a clinical document, eSignatures, and other documents (e.g. attachments); and
- HL7 v2 Diagnostics and Referral messages

Please refer to the *Clinical Package Validator Product Data Sheet* for further details of tests in scope.

2.1 CDA validation

2.1.1 Package validation

2.1.1.1 Package requirements

The clinical package requirements describe how to construct a conformant ZIP file containing a clinical document, create and include eSignatures, and properly reference attachments from that clinical document.

The requirements for a clinical package are described in the CDA Package specification [ADHA2011a] and the *Clinical Documents - Common Conformance Profile* [ADHA2015c]. Requirements for referencing attachments are listed in the *CDA Package Specification* [ADHA2011a], *CDA Rendering Specification* [ADHA2012a], and the *Clinical Documents - Common Conformance Profile* [ADHA2015c].

Additional requirements for clinical packages apply, depending on whether the My Health Record (MHR) or provider to provider (P2P) context is selected by the user. The *PCEHR Document Exchange Service Logical Service Specification* [ADHA2014b] and the *PCEHR Document Exchange Service Technical Service Specification* [ADHA2014c] list requirements that apply to clinical packages sent to the My Health Record system (referred to as the My Health Record context). Requirements for clinical packages sent from one healthcare provider to another healthcare provider (referred to as the P2P context) are listed in the *P2P Document Delivery Technical Service Specification* [ADHA2012b].

The P2P context imposes fewer constraints on a clinical package than the My Health Record context. For example, a general practitioner may attach a report from a pathology laboratory to an eReferral that is to be sent to a specialist. If, however, the eReferral is to be sent to the My Health Record system, the general practitioner cannot attach the pathology report to the

eReferral, because only the author of a document can upload that document to the My Health Record system.

2.1.1.2 Assessing a clinical package using the Validator

To determine whether a clinical package conforms to some packaging requirements, the Validator applies a non-exhaustive set of tests. The tests that are applied will depend on whether the user has selected the MHR or P2P context (Section 3.2.4) and on whether the Validator is able to perform the tests. You should refer to the *Clinical Package Validator Product Data Sheet* or contact the Agency Help Centre on 1300 901 001 or help@digitalhealth.gov.au to confirm that the Validator may be used to carry out the required tests. If the user selects the My Health Record context, only the set of tests for the My Health Record context is applied. Similarly, if the P2P context is selected, only the set of tests for the P2P context is applied.

Some of the clinical package test cases applied by the Validator for both the My Health Record and P2P contexts are described in the *Conformance Test Specification for CDA Packaging* [ADHA2015b].

2.1.1.3 Limitations of package validation

The package validation function of the Validator has the following limitations:

1 In both the My Health Record and P2P contexts:

- a The tests that are applied to an attachment to a clinical document do not work if the attachment is compressed.
- b The Validator checks the dates and times within a Public Key Infrastructure (PKI) certificate to make sure that the certificate was valid when it was used to sign a clinical document. It does not, however, check whether the certificate was on the revocation list when the clinical document was signed.
- c The Validator does not check whether national healthcare identifiers in the eSignature are registered in the national healthcare identifier service. Nor does it determine whether there is any relationship between the identifier for the approver, the identifier in the certificate, and the identifiers in the clinical document.
- d Some types of clinical documents may contain a reference to a file (e.g. an image) on a network location such as a website. The Validator does not verify the value of the integrityCheck attribute that may be associated with such a reference.[1]

2 In the P2P context:

- a Package validation tests are not applied to an attachment to a clinical document if that attachment is a clinical package. In this case, the tester must extract the attachment from the clinical package and manually apply the Validator to it.
- b The tests that are applied to an attachment to a clinical document do not work if the attachment and clinical document are in different folders.
- c A clinical package may contain multiple eSignature files; typically, one for each person who approved the clinical document and the clinical information within it. The primary eSignature file has the filename CDA_SIGN.XML and this is the only eSignature validated

[1] To perform this verification, a tester may access tools or websites that apply the SHA-1 calculation to the referenced file. It is important to note that the Agency's specifications require the inclusion of the integrity check value in a base64 format rather than in a hexadecimal format.

by the Validator. The Validator will report if it finds other eSignatures, but it will not validate them.

d A clinical package may contain a package index, but the Validator does not examine the package index when validating a clinical package.

2.1.2 Template validation

The Validator can apply two templates to a clinical document:

- A My Health Record template which is applied by the My Health Record system to a clinical document uploaded to it; and
- an additional template containing additional validation rules.

By default, a common set of the latest template packages are included as part of the installation.

A My Health Record template exists for each type of clinical document at each level of conformance. When a clinical document is being validated, the Validator will apply the template that the user has decided is relevant to that document including user-generated templates.

An additional template containing additional validation rules may also be applied if additional quality criteria apply in a particular context.

My Health Record templates may be obtained from a number of sources:

- template package libraries published on the Agency website[2], which contain a set of clinical document template packages; and
- template packages included in the My Health Record software vendor test environment.

Additional template packages are available from the Agency.

A template consists of the Australian CDA schema and Schematron rules. The Australian CDA schema constrains the set of data elements that may be used in a clinical document. The Australian CDA schema also contains rules that define the data types that are allowed for a data element, their attributes, and the allowed parent/child relationships. Schema rules are not specific to a type of clinical document or a conformance level, but are derived from the *HL7 Clinical Document Architecture, Release 2.0* specification [HL72004] and the *Australian CDA Schema Extension* [ADHA2014a].

Schematron is a rule-based validation language that is capable of expressing constraints that cannot be expressed in a CDA schema. My Health Record templates contain Schematron rules developed for each type of clinical document at each level of conformance. These rules describe the required cardinality of data elements in a clinical document and the relationships between these data elements. They also check whether a data element value is an allowed value.

Schematron rules are derived from the My Health Record conformance profile for each type of clinical document, the relevant structured content specification, the relevant CDA implementation guide, and specifications referred to by these documents.

[2] These can be derived from the relevant clinical document end product at http://www.digitalhealth.gov.au/implementation-resources/clinical-documents.

The Validator applies template validation to the following:

- Clinical documents created by clinical information systems and contracted service provider systems, such as:
    - birth details, child parent questionnaires, consumer entered achievements, consumer entered measurements, discharge summaries, eHealth diagnostic imaging reports, eHealth dispense records, eHealth pathology reports, eHealth prescription records, eReferrals, event summaries, health check assessments, shared health summaries, and specialist letters.
- Clinical documents created by consumer portals, such as:
    - advance care directive custodian documents, personal health notes, and personal health summaries.
- Clinical documents created by Medicare repositories, such as:
    - Australian immunisation register reports, Australian organ donor register reports, Medicare Department of Veterans' Affairs (DVA) benefits reports, and pharmaceutical benefits reports.
- Clinical documents created by the My Health Record system, such as:
    - eHealth prescription and dispense views, health check schedule views, Medicare overviews, and observation views.

2.1.2.1 Limitations of validation by My Health Record templates

The My Health Record template validation function of the Validator has a number of limitations. These are listed in detail in the *Clinical Package Validator Product Data Sheet*. Broadly speaking, the following high-level limitations apply:

1 My Health Record templates published by the Agency have been developed for My Health Record requirements rather than P2P requirements. If the P2P context is selected and the Validator reports that the clinical document references attachments of a disallowed type, the template error message may be ignored because the error message only applies to the My Health Record context.

2 My Health Record template validation only checks whether the minimum requirements for a conformance level have been met. No checks are applied to those data elements in a clinical document that are not required for that conformance level. For example, because the body of a conformance level 2 document need not contain structured data, the template for a level 2 document does not contain any checks for structured data. The Validator will ignore any structured data that may be present in a document that is being assessed for level 2 conformance. To assess this structured data, a template for level 3A conformance may be applied by the Validator, even though the document is only being assessed for level 2 conformance.

3 Extensions to a clinical document are only subjected to Australian CDA schema checks. The My Health Record templates contain no Schematron rules for these extensions. Extensions may be additional sections or data elements that have not been defined in the relevant CDA implementation guide. A clinical document may also be extended by incorporating information from any detailed clinical model published by the Agency.

4 My Health Record templates do not include rules for every conformance requirement listed in the specifications. They are only an aid for assessing the conformance of clinical documents. Validation using My Health Record templates must be accompanied by manual

inspection of clinical documents and the application of the relevant conformance test specification for the selected type of clinical document.

An additional template with additional rules developed by the user, or the Information Quality Rules ("IQ Rules") developed by the Australian Digital Health Agency, may be applied to assist a user in overcoming these limitations.

2.1.3 XDS Metadata validation

Additional rules are applied when uploading a document to the My Health Record system. These rules test data that are not only inside the CDA Package but accompany the CDA Package as what is known as XDS Metadata.

This validation takes a request SOAP message (XML) for uploading a CDA document to the My Health Record system and does a number of checks that the My Health Record system also performs but gives a more helpful response than 'Document metadata failed validation'.

It also displays all the data it finds so the user can check it was what they expected it to be.

2.1.4 Other limitations of the Validator

In addition to the limitations of package validation (Section 2.1.1) and template validation (Section 2.1.2) the following limitations also apply to the Validator:

1 Where the Validator can be used to test a clinical document (please refer to the *Clinical Package Validator Product Data Sheet*), the Validator renders that clinical document into a human-readable form. Its rendering conforms to most, but not all, of the generic requirements for the rendering of clinical documents stated in the *CDA Rendering Specification*, *Clinical Documentation* [ADHA2012a].

2 The generic requirements for rendering clinical documents do not apply to eHealth prescription and dispense views, even though these views are clinical documents. The Validator does not implement the specific requirements for rendering eHealth prescription and dispense views, but instead applies the generic CDA rendering requirements.

3 Some types of conformance tests are inherently manual and cannot be automated in the Validator.

As a result of these and other limitations, the validation of a clinical document cannot be solely performed by the Validator. It must be accompanied by manual inspection of the clinical document's XML file.

2.2 HL7 v2 validation

The Validator is able to run a number of tests over certain HL7 v2 messages as per the conformance profiles defined on the HL7 Australian Diagnostics and Referral Messaging - Localisation of HL7 Version 2.4 website [HL7OOWG]. This covers the following message types:

- ACK^I12
- ACK^O01
- ACK^R01
- ORM^O01
- ORR^O02
- ORU^R01
- REF^I12

In addition, the Validator is also able to test the following HL7 v2.3.1 message types as profiled in the Agency Messaging Specification [HL7MDM]:

- ACK^T02
- MDM^T02

3 Configuring the Validator

3.1 Launch the Validator

1. Launch the Validator by selecting the Windows icon and locate the Clinical Package Validator in the Metro view (Figure 1) or use the desktop icon.

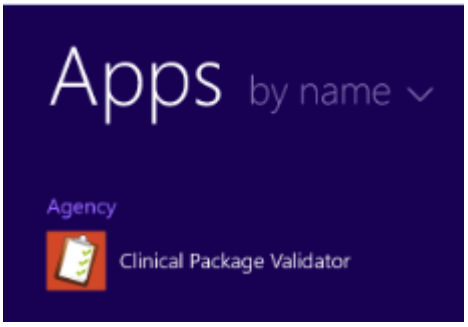


Figure 1 -Clinical Package Validator

2. By default, to assist the user, a common set of the latest template packages are included as part of the installation. A warning message is displayed if the Validator detects that a NASH organisation test certificate has not been installed. To continue select **OK** (Figure 2).

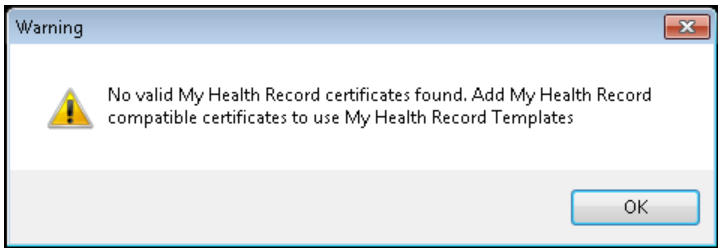


Figure 2 – Select Ok

Note: The Validator may be used without a NASH organisation test certificate if template packages are downloaded from the Agency website and imported into the Validator. The recommended approach, however, is to import template packages from the My Health Record software vendor test environment.

3. There may be a delay of several seconds before the terms and conditions message is displayed. If you accept the terms and conditions, select **Yes** to continue. If you select **No**, the Validator will close (Figure 3).

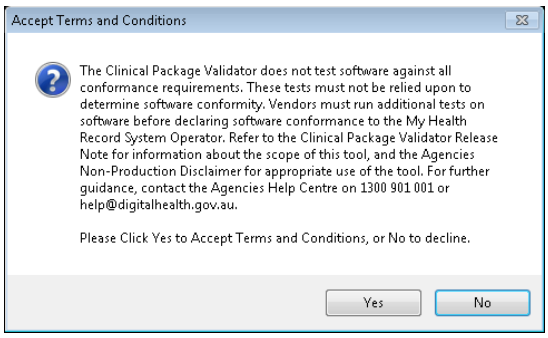


Figure 3 – Select Yes to Accept Terms and Conditions

4. The template manager will now be displayed (Figure 4)

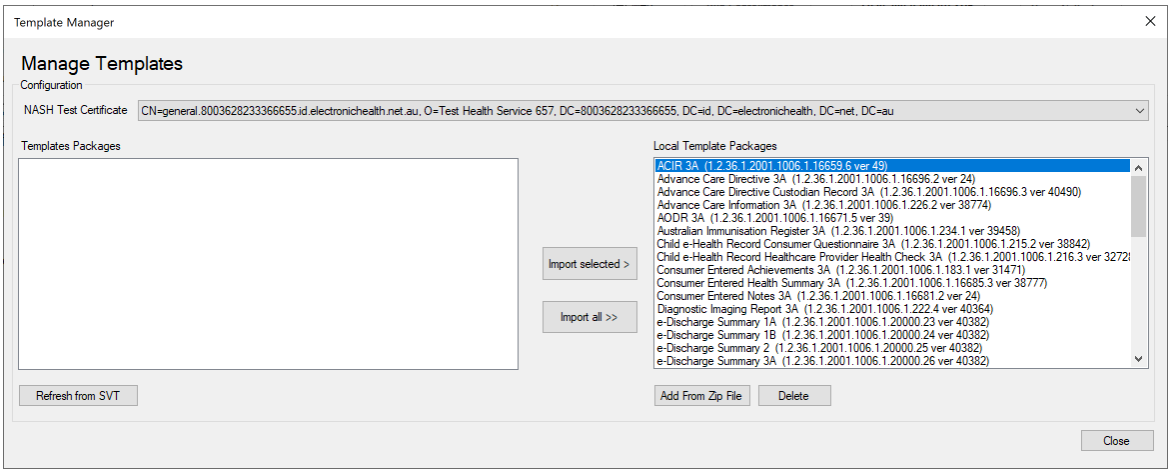


Figure 4 – Template Manager

To proceed, either import template packages from the My Health Record software vendor test environment (SVT) (Section 3.2.2) or import template packages downloaded from the Agency website (Section 3.2.1)

3.2 CDA validation

3.2.1 Select template packages

The Validator uses template packages to assess whether some clinical documents conform to published specifications. Template packages include schema and Schematron rules.

The Validator needs a template for each type of clinical document that is in scope for the Validator to test and for each level of conformance. By default, to assist the user, a common set of the latest template packages are included as part of the installation. As templates are revised, more than one template may exist for a specific clinical document type and conformance level. Testers should use the most recent version of a template unless there is a reason for choosing an earlier version.

The Validator does not test all conformance specifications. Please refer to the Validator Product Data Sheet for further information about exclusions and the Non-Production Disclaimer.

Template packages may be obtained by:

- downloading the template packages from the Agency website and importing them into the Validator; or
- importing the template packages from the My Health Record software vendor test environment into the Validator.

For either option, the template package directory on the Agency website should be viewed to determine which version of the template packages should be used. To view the clinical documents template package directory, follow the steps below.

1. Download the template package directory from the Agency website[3] and save a copy. **Note:** If the webpage for the template package directory states that a newer version is available, download the newest version (Figure 5).

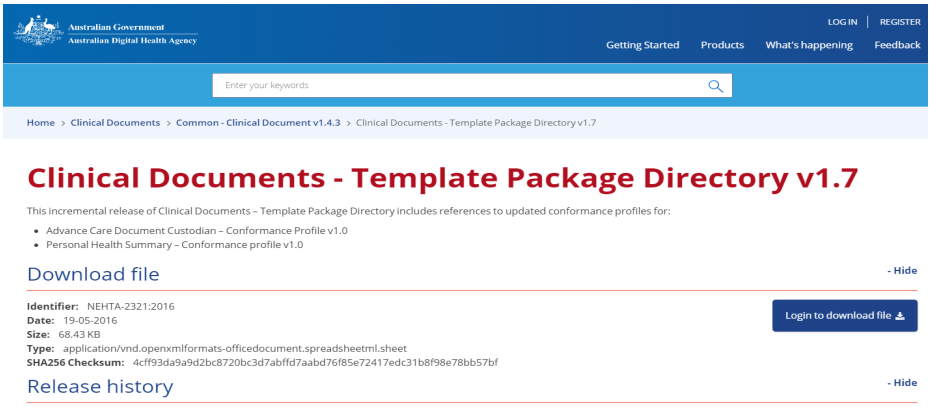


Figure 5 – Template Package Directory

2. Read the 'usage instructions' worksheet in the template package directory spreadsheet (Figure 6) and determine from the 'directory' worksheet the template package identifier and version number to use. **Note:** An 'HPIIRelaxed' template package may only be used to upload documents to the My Health Record system with the permission of the My Health Record System Operator.

[3] https://www.digitalhealth.gov.au/implementation-resources/clinical-documents/EP-2320-2016/NEHTA-2321-2016

Usage Instructions	
Purpose	The purpose of the Template Package Directory (TPD) is to provide implementers of clinical document specifications with a comprehensive overview of available template packages and their current status.
Usage	The actual content of the TPD is provided on the *directory* worksheet. Its columns are explained below.
Template Package Library	For each listed template package, a hyperlink to the download location of the Template Package Library containing the template package is provided. Template packages are published as part of a Template Package Library (TPL). TPLs are published as part of specification end products. In general, the TPL is published in the same version of the end product that contains the My Health Record Conformance Profile version that its template packages are aligned with. To access a particular template package, please download the respective TPL, unpack it and use the TPL's overview document as an index to look up the template package within the TPL.
Adoption Event Dates	The adoption lifecycle for clinical document specifications can be characterised through a series of adoption events. Each of these events has implications for CISs producing or consuming clinical documents, and the My Health Record system. The TPD lists approved adoption event dates where they have been defined.
	`specification published`: Publication date of the specification that the template package is aligned with.
	My Health Record SVT available: Template package is available in the My Health Record SVT environment.
	My Health Record SVT retired: Template package is no longer available in the My Health Record SVT environment.
	My Health Record PROD available: Template package is available in the My Health Record Production environment.
	My Health Record PROD retired: Template package is no longer available in the My Health Record Production environment.
	More information about adoption events and their associated dates can be found in the Clinical Documents Adoption Framework.

Figure 6 – Template Package Directory Spreadsheet

3.2.2 Import template packages from the My Health Record SVT

Template packages may be imported from the My Health Record software vendor test (SVT) environment as an alternative to importing them from the Agency website.

Note: A NASH organisation test certificate must first be installed before template packages can be imported from the SVT environment. To import template packages from the SVT environment, follow the steps below.

1. Select the **Configuration** menu (Figure 7)

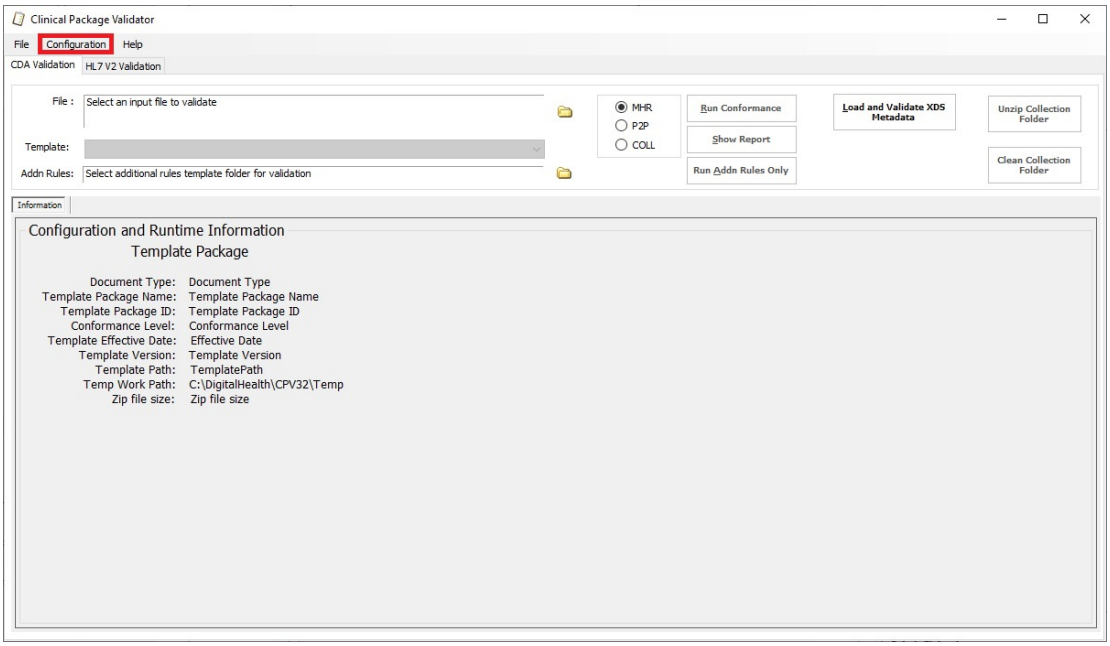


Figure 7 – Configuration Menu

2. Select **Manage Templates** (Figure 8)

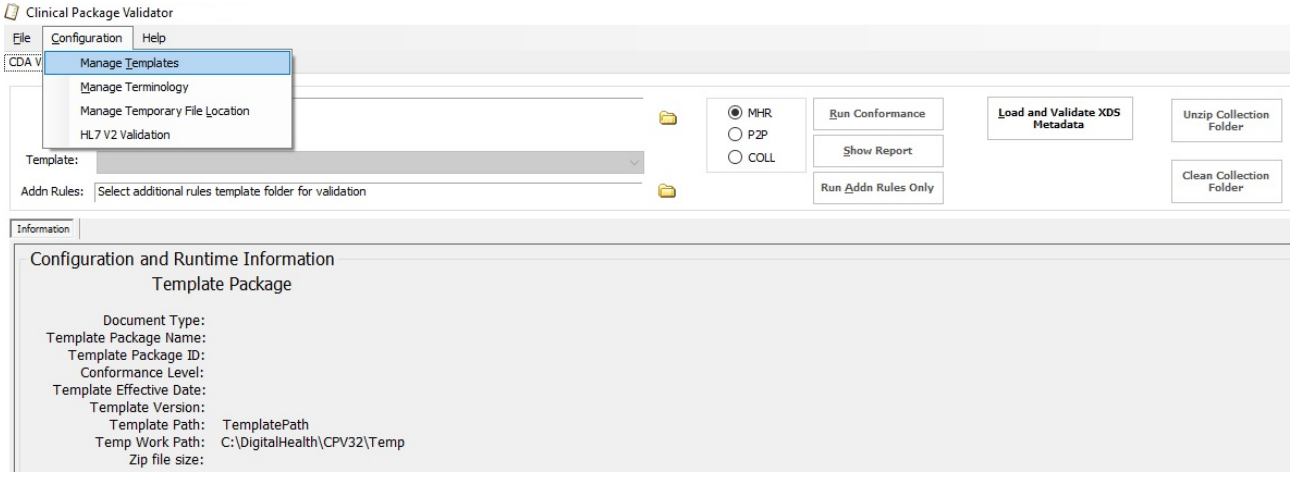


Figure 8 – Manage Templates

3. If one or more NASH organisation test certificates have been installed, the available certificates will be displayed in a pull-down list (Figure 9). Select the certificate to be used.

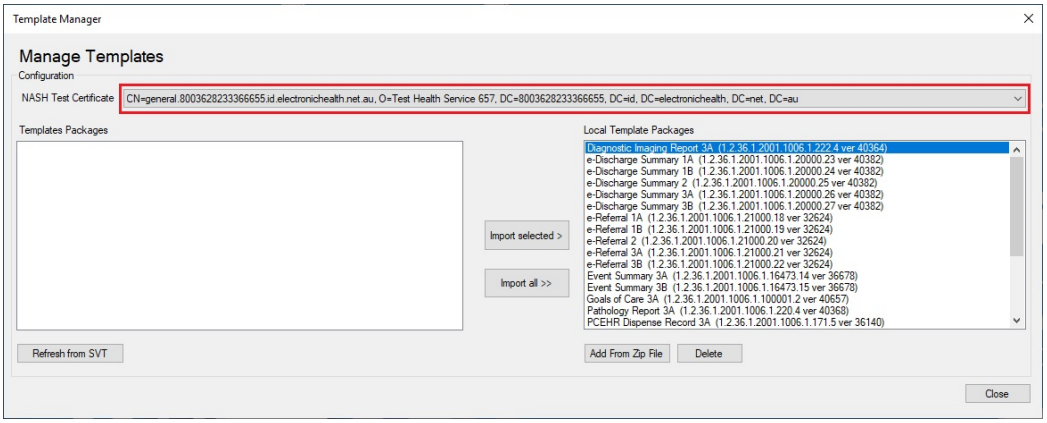


Figure 9 – NASH Test Certificate

4. To obtain the list of template packages that may be imported from the SVT environment, select **Refresh from SVT** (Figure 10)

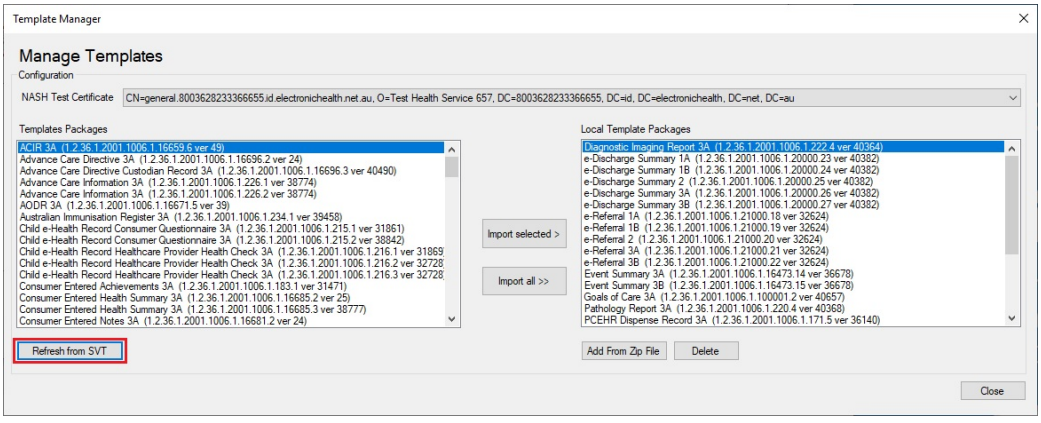


Figure 10 – Refresh from SVT

5. An error message is displayed if the SVT environment cannot be contacted. To continue, select **OK** and determine if there is a working connection to the Internet (Figure 11)

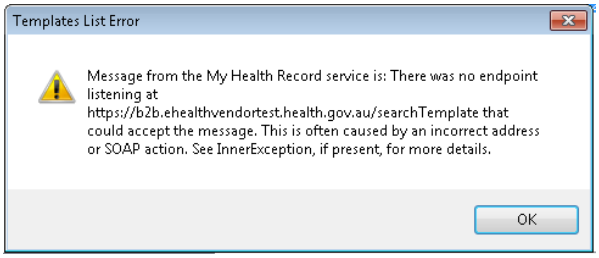


Figure 11 – Select OK

6. The list of template packages will be obtained from the SVT (Figure 12)

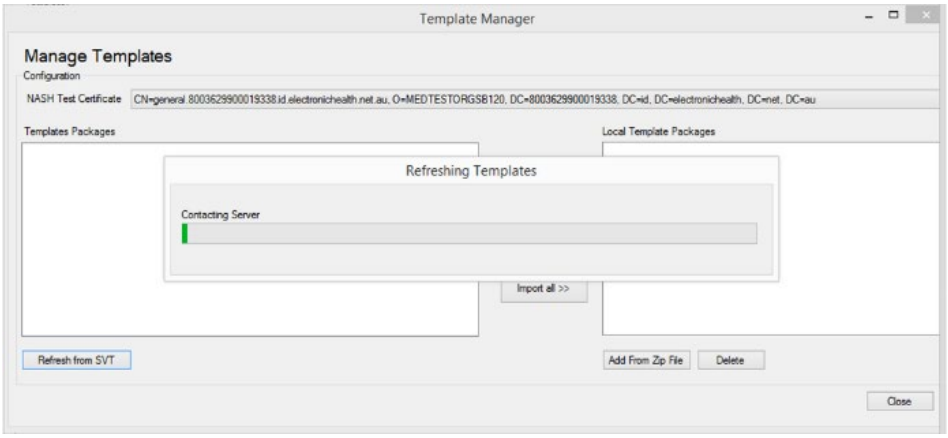


Figure 12 – Refreshing Templates

7. Choose the required template package and select **Import selected**. **Note:** An **Import all** option is also provided (Figure 13)

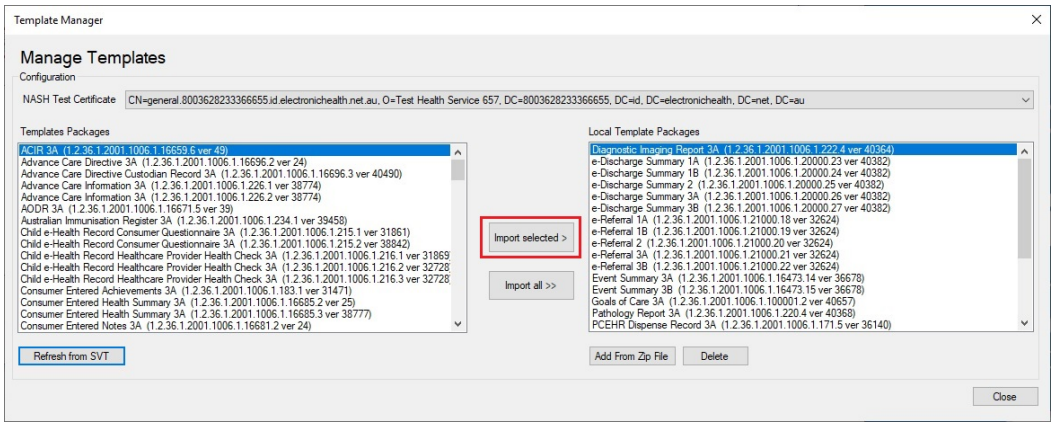


Figure 13 – Refreshing Templates

8. Verify that the template package has been imported into the Validator and select **Close** (Figure 14)

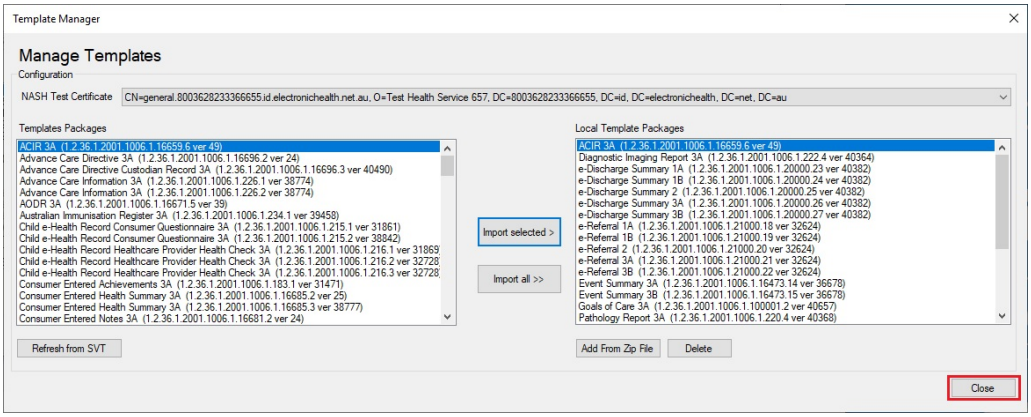


Figure 14 – Select Close button

3.2.3 Import template packages downloaded from the Agency website

Template packages may be downloaded from the Agency website and imported into the Validator as an alternative to importing them from the My Health Record software vendor test environment. To download a template package from the Agency website, follow the steps below.

1 Download the template package library (Figure 15) from the selected clinical document type end product, available from https://www.digitalhealth.gov.au/implementation-resources/clinical-documents

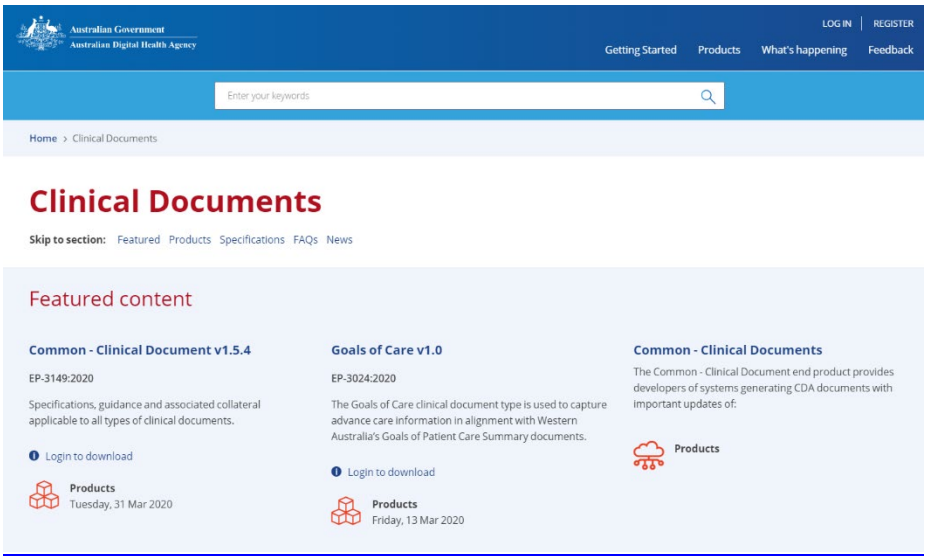


Figure 15 – Template Package Library

2 Unzip the template package library ZIP file and save the selected template packages to a preferred location. **Note:** A template package is in the form of a ZIP file within the template package library. The template packages should not be unzipped.

3 To import the downloaded template package, select the **Configuration** menu (Figure 16)

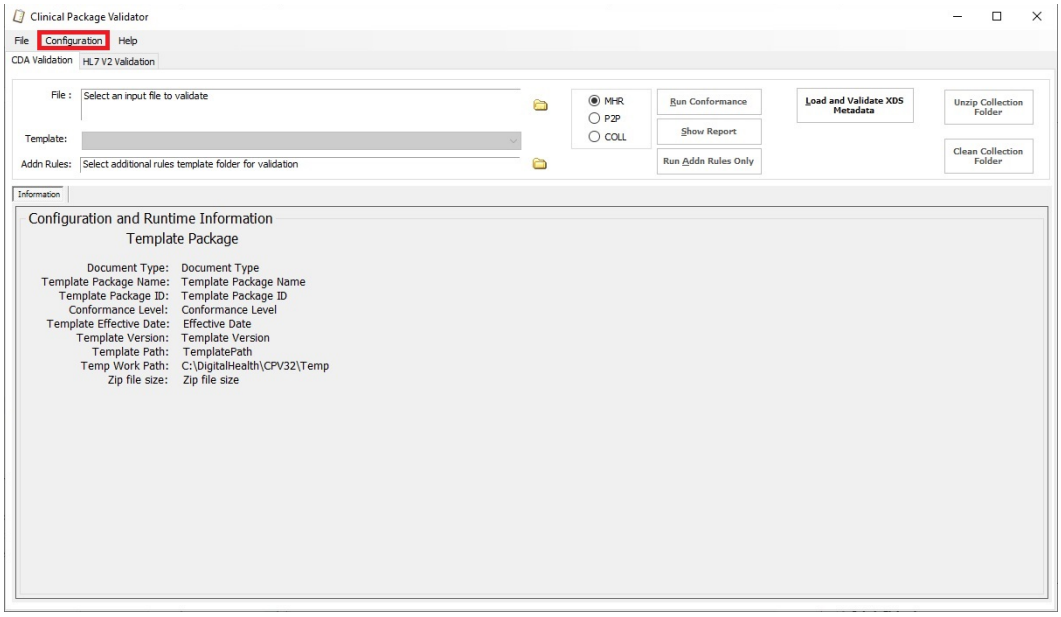


Figure 16 – Configuration Menu

4 Select Manage Templates

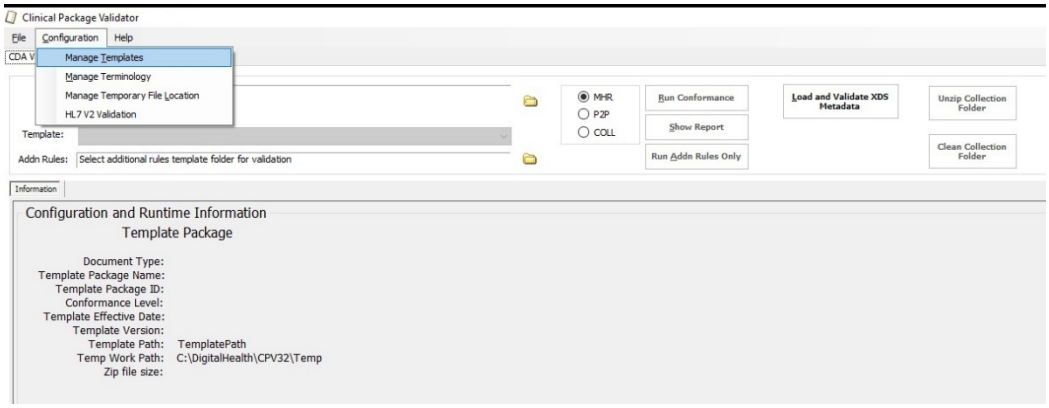


Figure 17– Manage Templates

5 Select **Add From ZIP File**

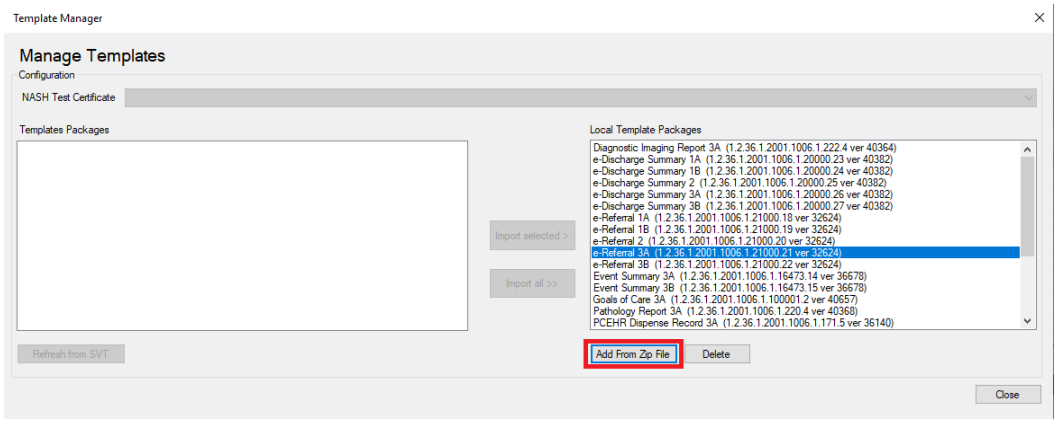


Figure 18 – Add From Zip File

6 Navigate to the directory in which the downloaded template package has been stored, locate the required template package ZIP file and select **Open** (Figure 19).

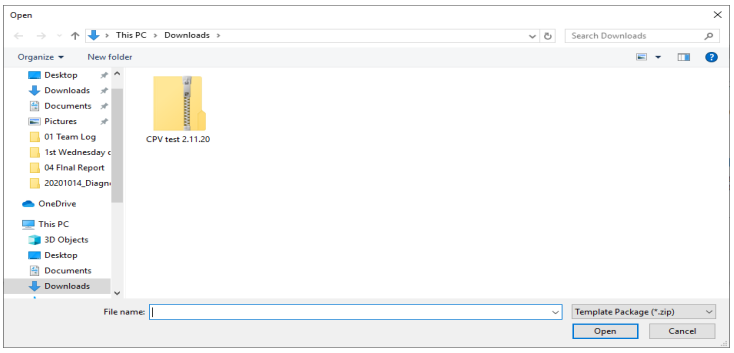


Figure 19 – Open Zip File

7 Verify that the template package has been imported into the Validator and select **Close** (Figure 20)

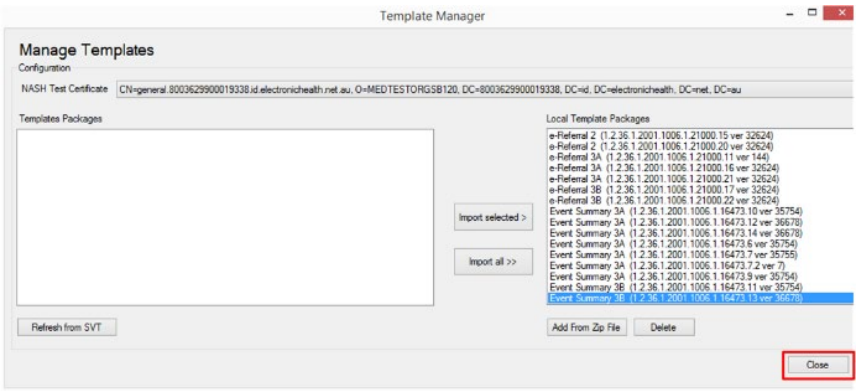


Figure 20 – Select Close

3.2.4 Delete template packages

Template packages imported into the Validator should be deleted if they are no longer needed.

1 To delete a template package from the Validator, select the **Configuration** menu (Figure 21)

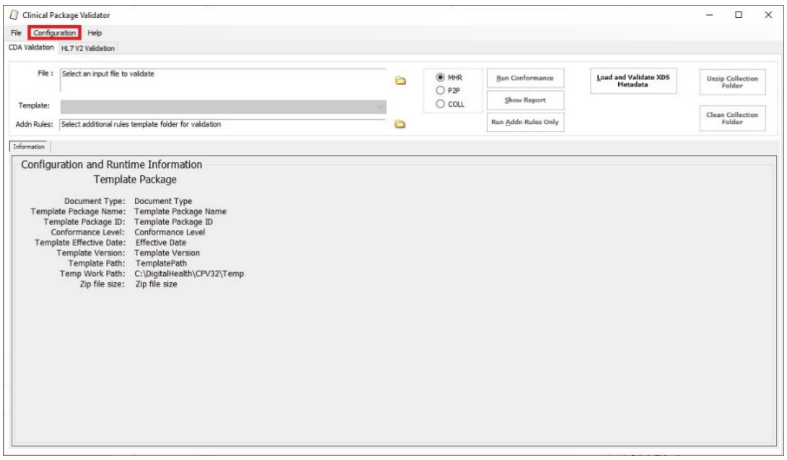


Figure 21 – Select Configuration

2 Select Manage Templates

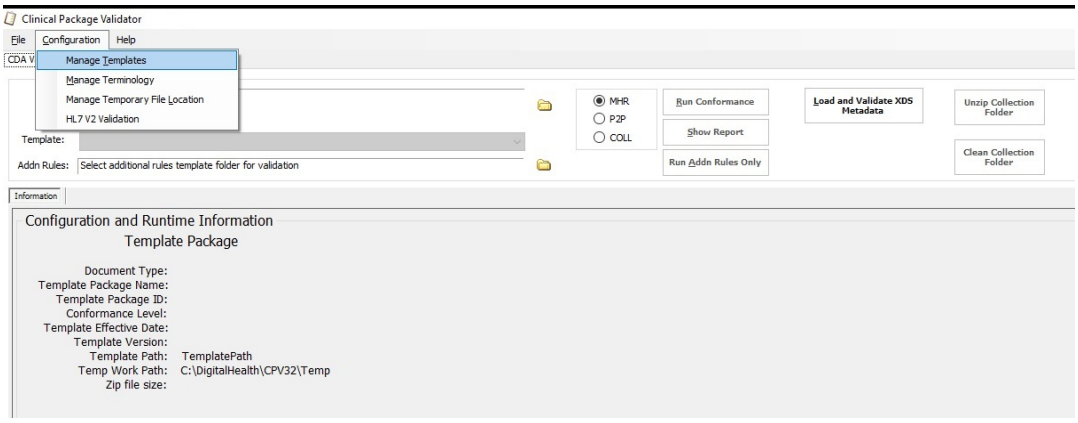


Figure 22 – Manage Templates

3 Choose the template package to be deleted and select **Delete** (Figure 23)

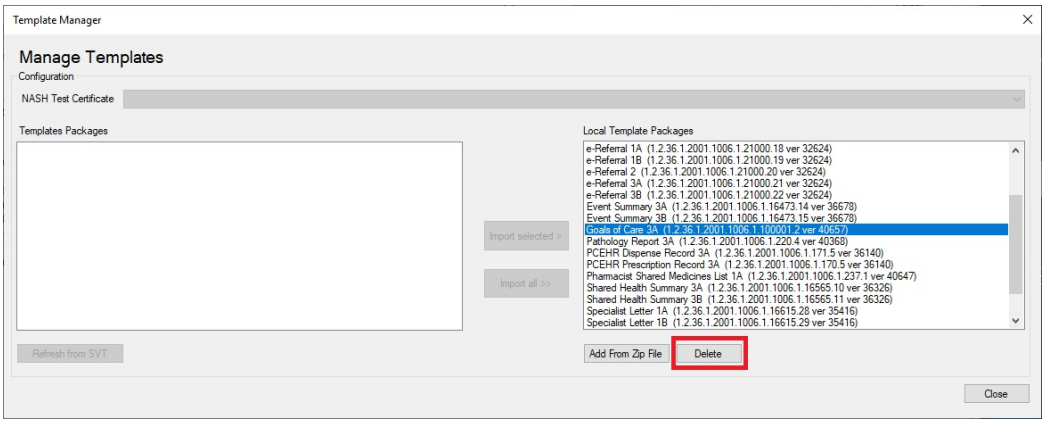


Figure 23 – Select Delete

3.2.5 Configure clinical terminology datasets

The Validator is preloaded with clinical terminology datasets. As clinical terminology is frequently revised, resulting in new versions of clinical terminology datasets being regularly published, the Validator allows an administrator or user to upload new versions of the terminology datasets listed in Table 1. It is recommended to only load one version of SNOMED CT and PBS at any one time as the SQL Lite database doesn't perform very well. When loading new versions of these terminologies, it is best to remove the previous versions first.

The Validator allows a user to import combined SNOMED CT-AU/AMT v3 and PBS clinical terminology datasets in ZIP format.

Table 1 - Code systems the user can import into the Validator

Code system name	Code system identifier
Australian Medicines Terminology (AMT) version 3	2.16.840.1.113883.6.96
PBS Item codes	1.2.36.1.2001.1005.22
PBS Manufacturer codes	1.2.36.1.2001.1005.23
Systematized Nomenclature of Medicine - Clinical Terms Australian Release (SNOMED CT-AU)	2.16.840.1.113883.6.96

3.2.5.1 Determine which terminology datasets are loaded

1 To determine which versions of terminology datasets are already present in the Validator database, select the **Configuration** menu (Figure 24).

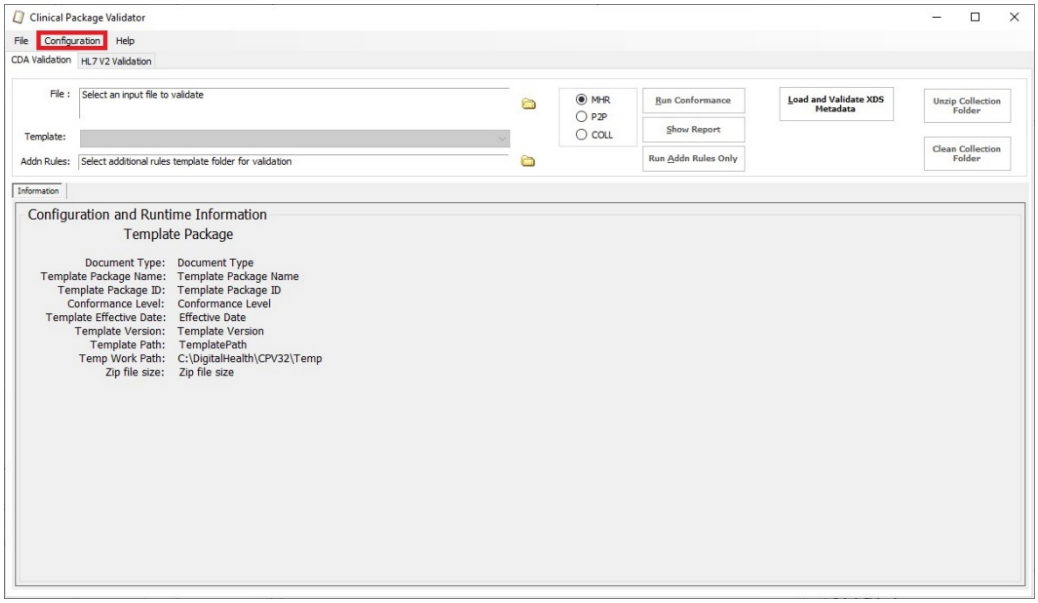


Figure 24 – Select Configuration Menu

2 Select Manage Terminology (Figure 25)

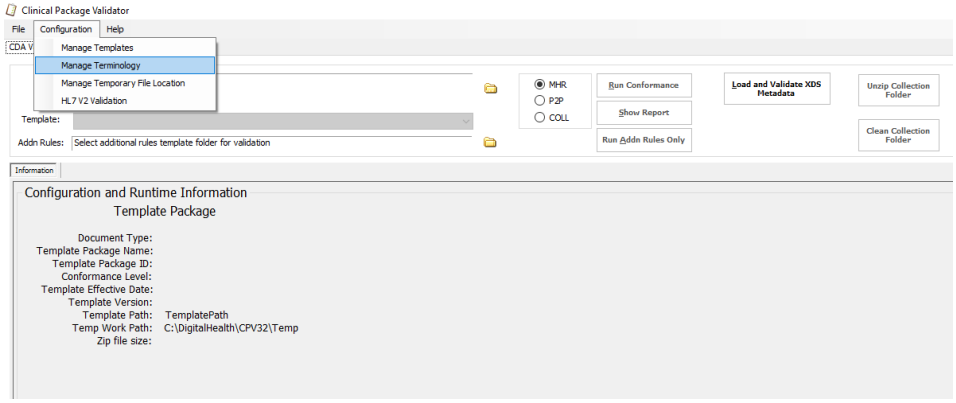


Figure 25 - Select Manage Terminology

3 A list of terminology datasets and their versions will be displayed (Figure 26)

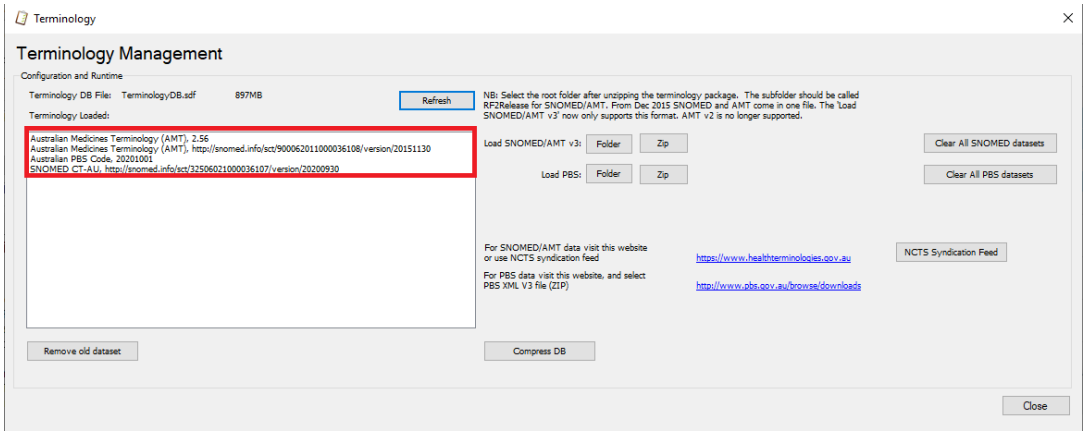


Figure 26 - Terminology Datasets

- AMT version 2's final version was 2.56. After that AMT was included under SNOMED releases.
- PBS uses the version number format 'YYYYMMDD' where 'YYYYMMDD' is the date of release of a specific PBS dataset (e.g. '20141130').
- SNOMED CT-AU uses the version number format 'http://snomed.info/sct/32506021000036107/version/YYYYMMDD' where 'YYYYMMDD' is the date of release of a specific SNOMED CT-AU dataset (e.g. 'http://snomed.info/sct/32506021000036107/version/20141130').

Note: PBS Item codes and PBS Manufacturer codes are both included in the PBS dataset.

3.2.5.2 Find and load another version of the combined SNOMED CT-AU and AMT v3 dataset

1 To obtain and load another version of the combined SNOMED CT-AU and AMT v3 dataset, select the link to the SNOMED CT-AU website (Figure 27). You would do this if your clinical document uses a terminology release that is not one of the versions included in the Validator. Since July 2016, all SNOMED/AMT terminology is now published on its own dedicated website: www.healthterminologies.gov.au

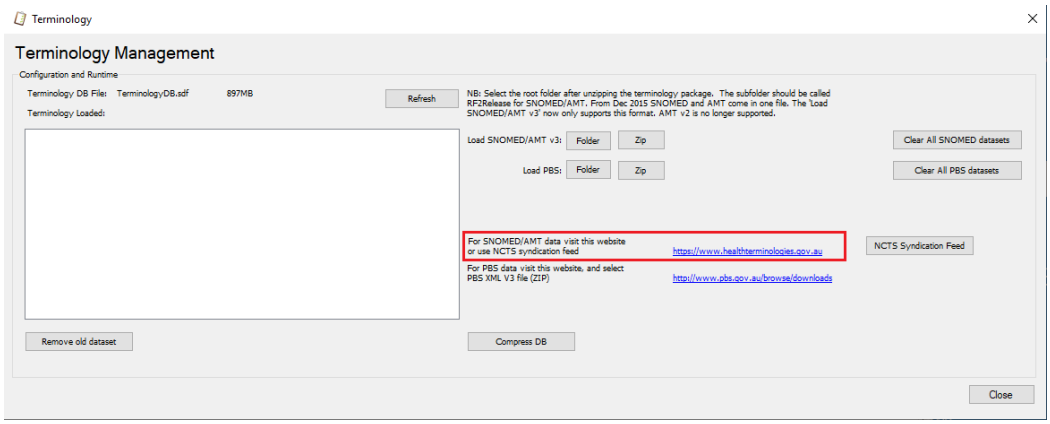


Figure 27 – Link to SMOMED CT-AU website

2 Select either **Login** or **Register** for the latest version of SNOMED CT-AU, or to select and download an earlier version (Figure 28). Note that the file includes both SNOMED CT-AU and AMT.

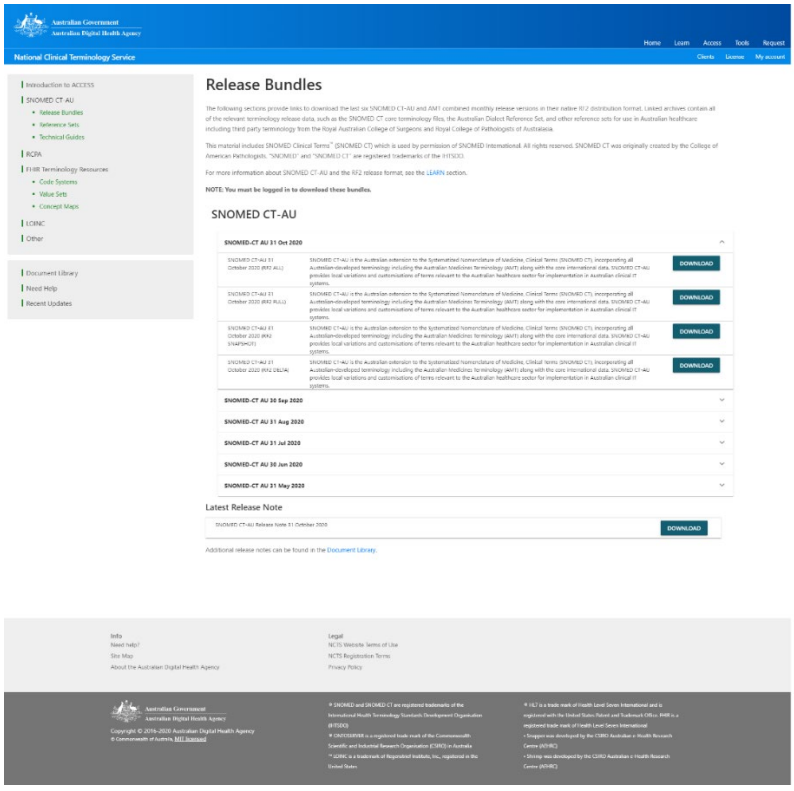


Figure 28 – SMOMED CT-AU website

Note:

- You need to be Logged in to be able to select and download the release
- Always download the SNAPSHOT version as that is what the Validator requires

3 Selecting the download icon, saves the zip file to your default Download directory. Move the file to c:\temp or another local temporary folder. The file downloaded should follow a similar pattern to this, where yyyymmdd is the monthly release date.

NCTS_SCT_RF2_DISTRIBUTION_32506021000036107-yyyymmdd-SNAPSHOT.zip

4 An alternative way to download the file from the website automatically is to get a Client ID and Client Secret from the health terminologies website. Select the Client Credentials tab below your user details and select the “Add” button. Enter the System Name and System Purpose and select Save (Figure 29). Note down the Client ID and secret:

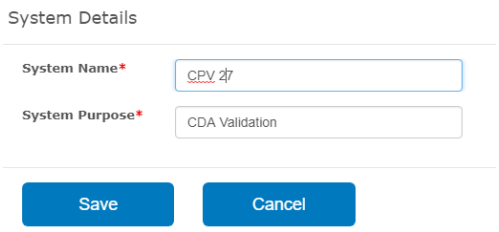


Figure 29 – System Details capture

5 Select the “NCST Syndication Feed” button (Figure 30)

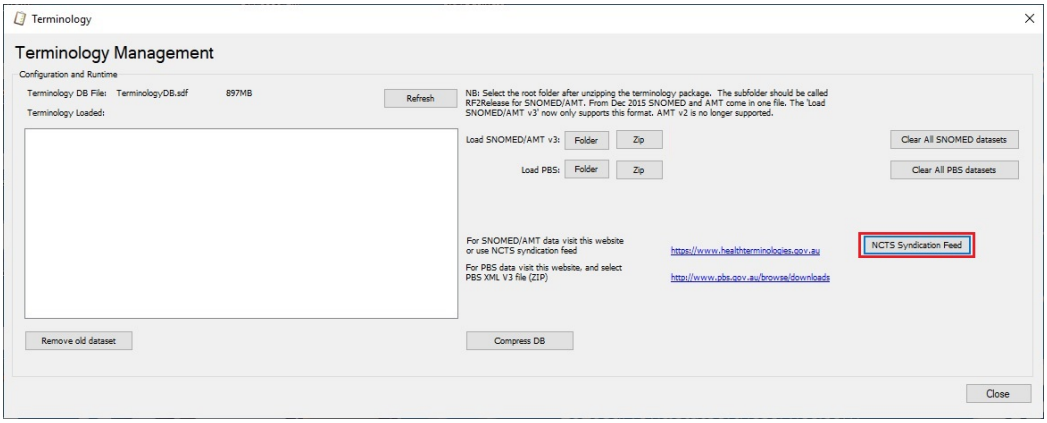


Figure 30 – NCST Syndication Feed

6 Select the folder icon , browse to a temporary folder such as c:\temp and select ok. Enter the Client ID and Client Secret (in step 4), and then select “Get Releases”. This will populate the list box with the last 6 months of SNAPSHOT files. Select the file required and “Download”. The progress bar will indicate the download speed - it may freeze during download, but when the download is complete it will update and show a full green bar (Figure 31).

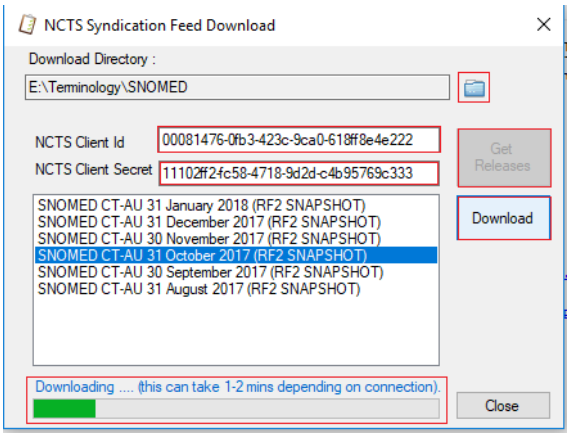


Figure 31 – Syndication Feed Download

The file downloaded should follow a similar pattern to this, where yyyymmdd is the monthly release date. NCTS_SCT_RF2_DISTRIBUTION_32506021000036107-yyyymmdd-SNAPSHOT.zip

7 Import the downloaded SNOMED CT-AU package ZIP file by selecting **Zip** (Figure 32) for "Load SNOMED".

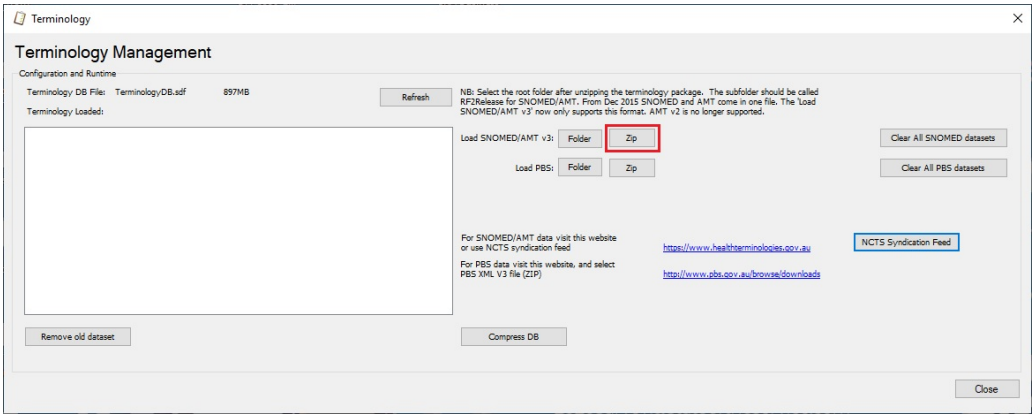


Figure 32 – Select Zip button

Note: If the SNOMED CT-AU package ZIP file was already downloaded and unzipped into a folder, the contents of the folder may be imported by selecting **Folder**.

8 Select the SNOMED CT-AU package ZIP file and select **Open (Figure 33)**

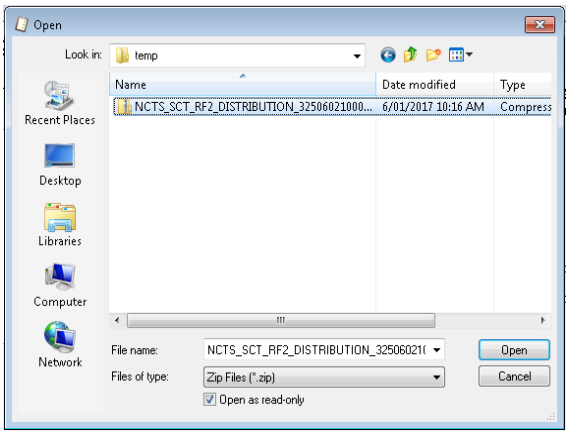


Figure 33 – Open Zip file

9 The message 'Unzipping data' is displayed while the SNOMED CT-AU data is being imported.

Note: The loading process may take many minutes to complete

10 Confirm that the SNOMED CT-AU dataset has been successfully loaded into the Validator database. The message 'Loaded data into database' (Figure 34) should appear next to the button that was selected to load the dataset and the terminology dataset should appear in the 'Terminology Loaded' list

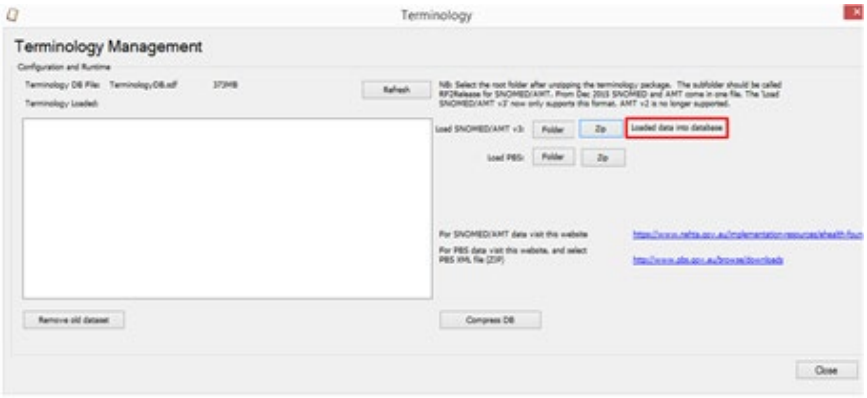


Figure 34 – Loaded data into database

3.2.5.3 Find and load another version of the AMT v3 dataset

Since Oct 2016, the Agency has stopped releasing separate releases of AMT. AMT is now only released within the SNOMED release.

3.2.5.4 Find and load another version of the PBS dataset

1 To obtain and load another version of the PBS dataset, select the link to the PBS website (Figure 35).

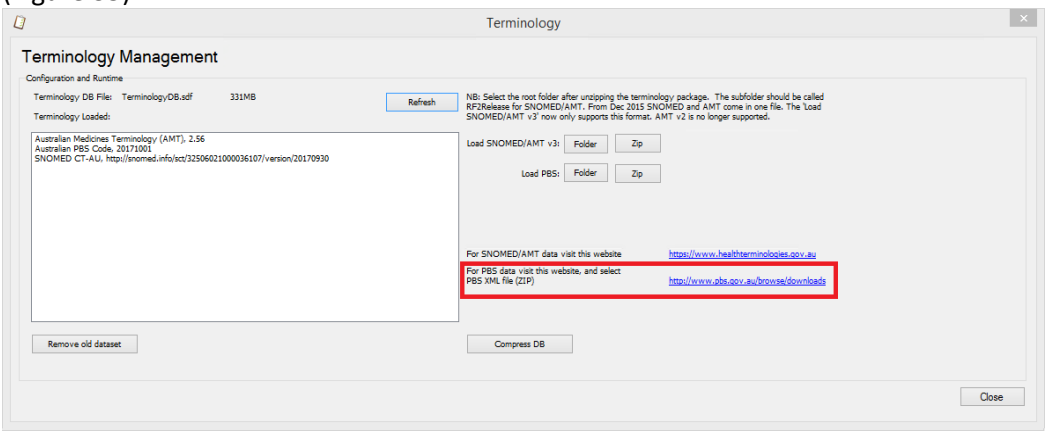


Figure 35 – Select the link to the PBS website

Note: PBS Item codes and PBS Manufacturer codes are both included in the PBS dataset.

2 Either download the latest version of the PBS dataset by selecting the **PBS XML file** link (Figure 36) or select **Publications Archive** to choose an earlier version of the PBS dataset.

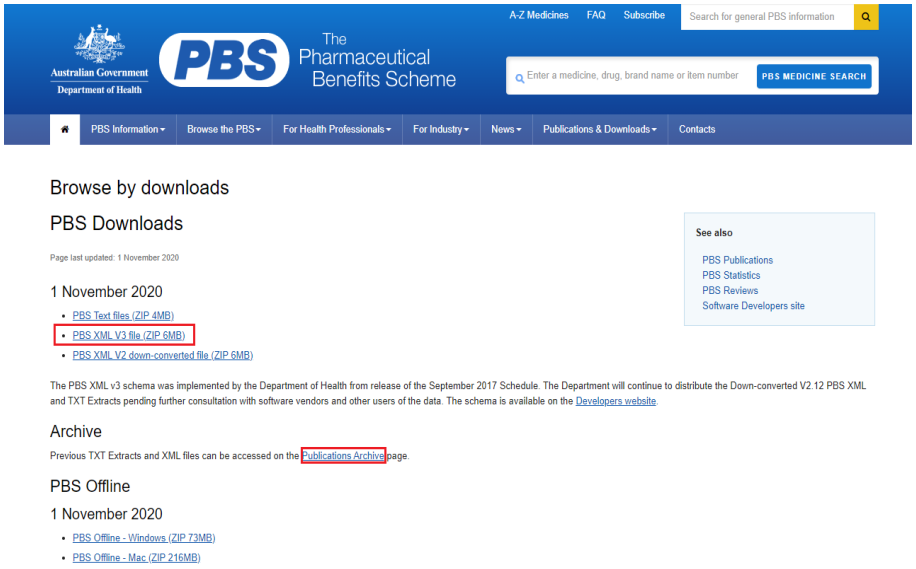


Figure 36 – Select the link to the PBS XML file

3 3. Save a copy of the selected PBS XML file in the C:\temp directory or another local directory (Figure 37)

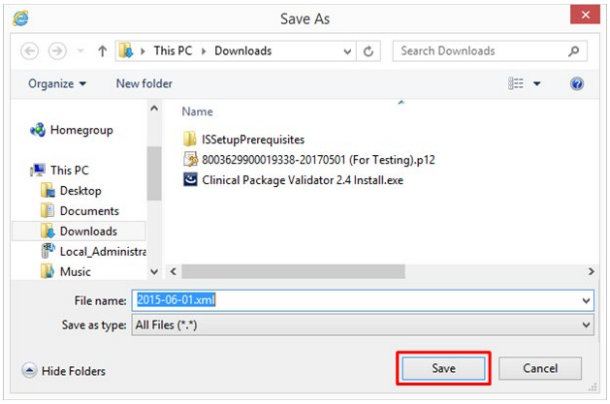


Figure 37 – Select Save

4 Import the downloaded PBS package ZIP file by selecting **Zip** for "Load PBS" (Figure 38)

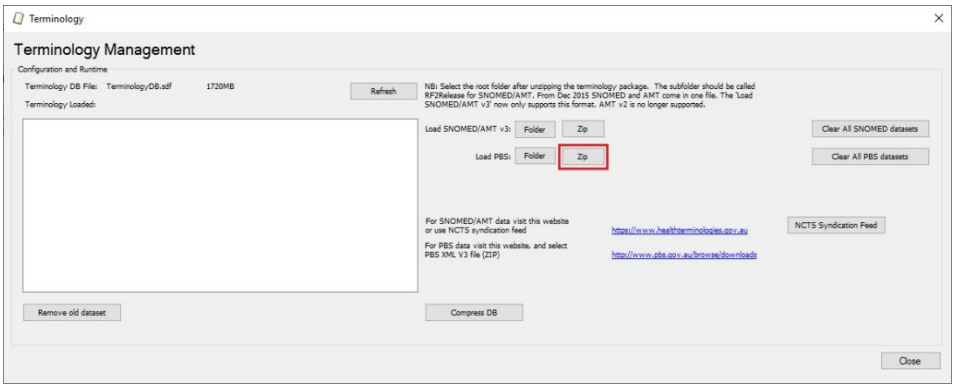


Figure 38 – Select ZIP

Note: If the PBS package ZIP file was already downloaded and unzipped into a folder, the contents of the folder may be imported by selecting Folder.

5 Select the PBS package ZIP file and select **Open** (Figure 39)

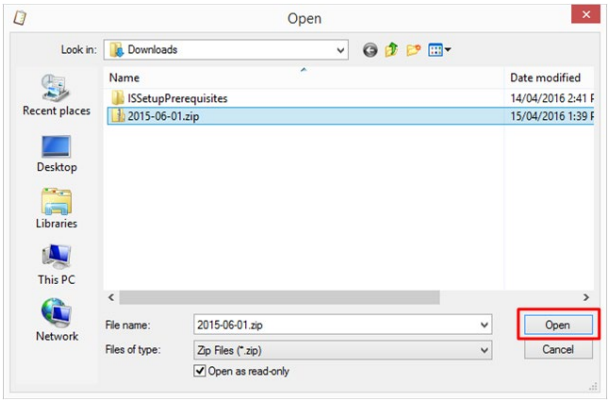


Figure 39 – Select PBS package ZIP and Open

6 Two messages (i.e. 'Unzipping data' then 'Loaded data into database') are displayed while the PBS data is being imported (Figure 40)

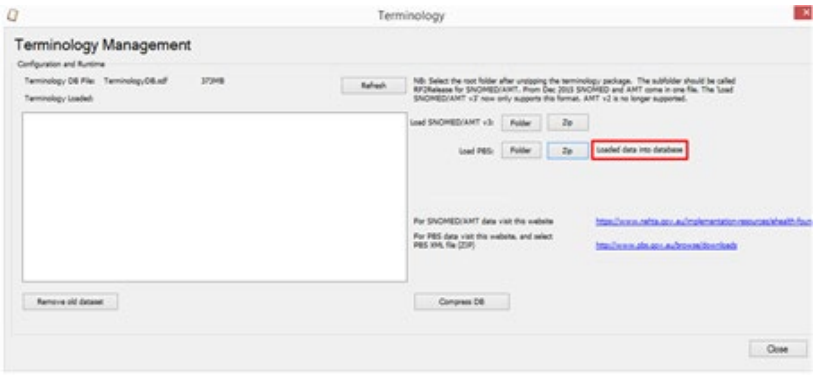


Figure 40 – Select PBS package ZIP and Open

Note: The loading process may take many minutes to complete.

7 Confirm that the PBS dataset has been successfully loaded into the Validator database. The message 'Loaded data into database' (Figure 41) should appear next to the button that was selected to load the dataset

Figure 41 – Loaded into database

3.2.5.5 Remove a terminology dataset

1 If the list of terminology datasets currently loaded into the Validator is not displaying, select **Refresh** (Figure 42)

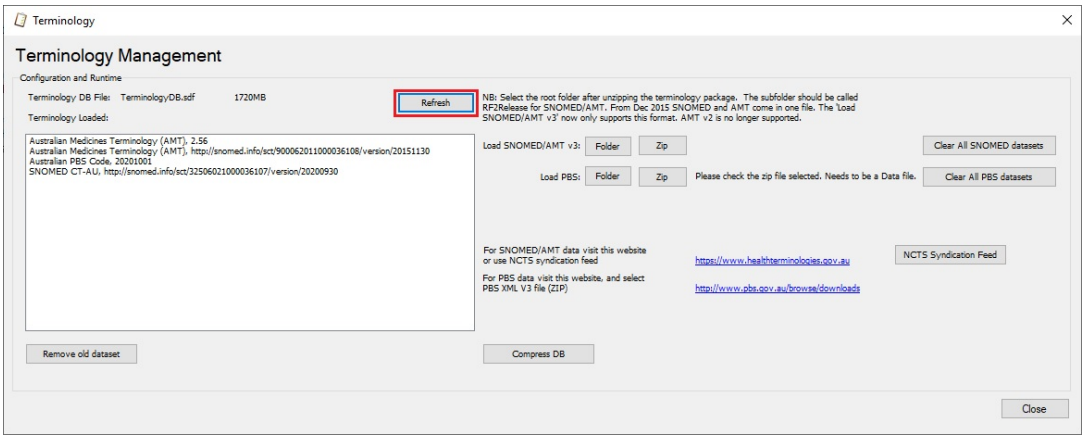


Figure 42 – Refresh for datasets

2 Choose the dataset to be removed and select **Remove old dataset** (Figure 43)**.** The removal process may take many minutes to complete

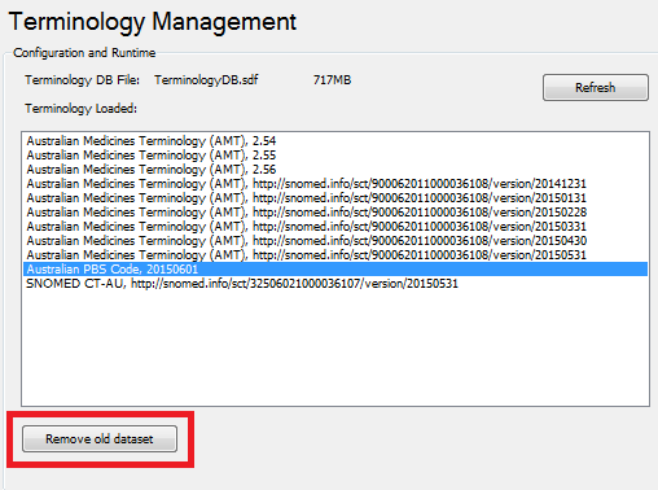


Figure 43 – Remove old dataset

3 Confirm the dataset is no longer present in the list of loaded datasets (Figure 44)

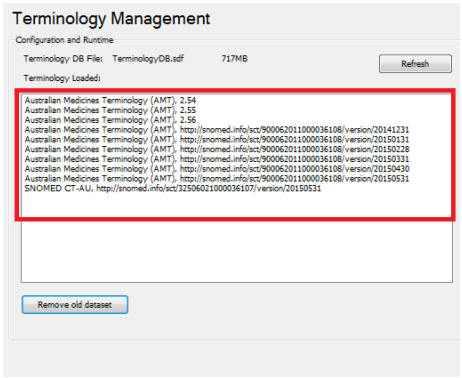


Figure 44 – Remove old dataset

4 If the database has multiple versions of a terminology, they can be cleared out with just clicking the “Clear All…” buttons on the right side (Figure 45). This is also useful if the database gets into an unstable condition.

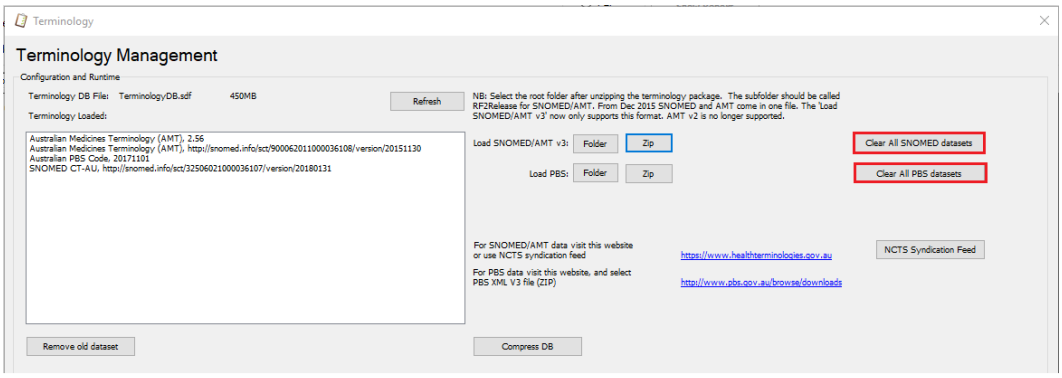


Figure 45 – Remove old dataset

3.2.6 Compress the database

The Validator’s terminology database may be compressed to reduce its size. Compression removes unused space and reduces the footprint of the database on the disk. Compression should be done after loading and/or removing datasets. Compressing the database does not affect the performance of the Validator. It is recommended to leave this setting with the default value unless there is a specific reason for doing so.

1 View the size of the database and select **Compress DB** to reduce its size (Figure 46). In this example, the size of the database before compression is 831 MB. **Note:** The compression process may take many minutes to complete.

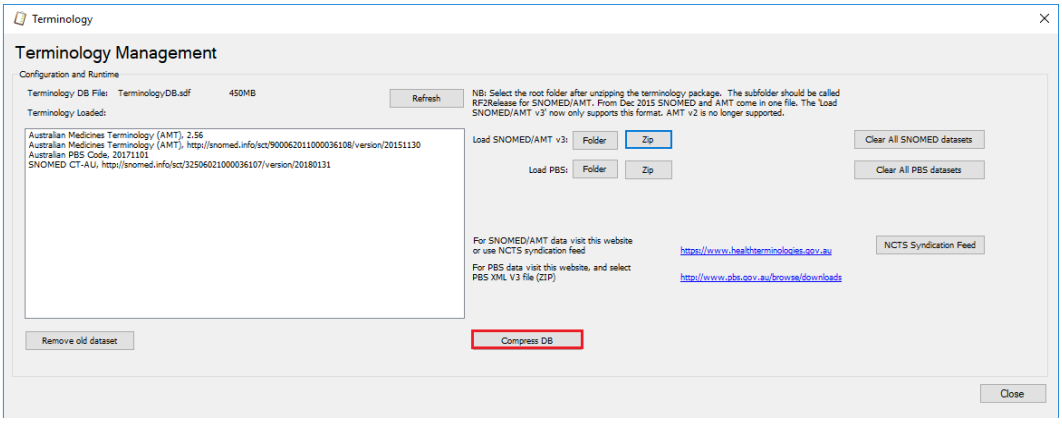


Figure 46 – Compress DB

2 View the size of the database again (Figure 47) to confirm that the compression has been successful. In this example, the size of the database after compression is 331 MB.

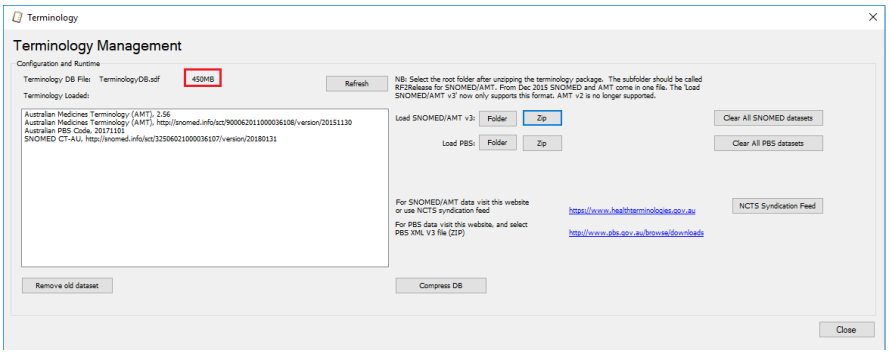


Figure 47 – Size of database

3.2.6 Configure the temporary file location

The Validator stores a set of files for a temporary period. By default, these are stored in C:\DigitalHealth\CPVxy\Temp (where “xy” refers to the Validator’s version number “x.y”) although a different location can be selected if required.

1 To change the location used by the Validator to store temporary files, select the **Configuration** menu (Figure 48)

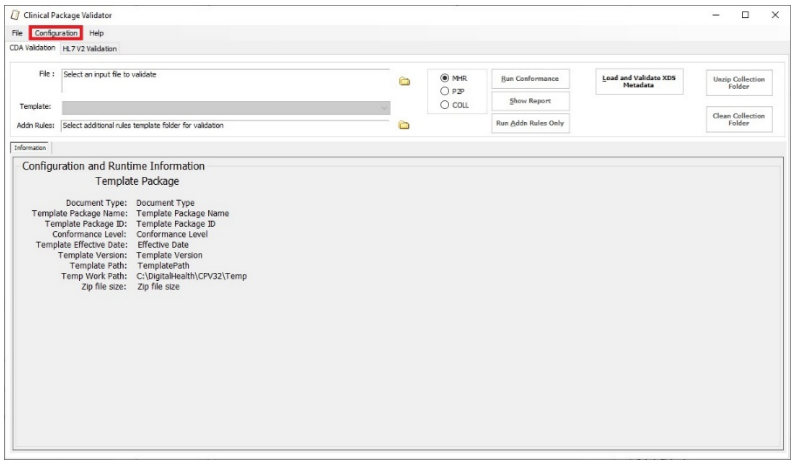


Figure 48 – Configuration Menu

2 Select **Manage Temporary File Location** (Figure 49)

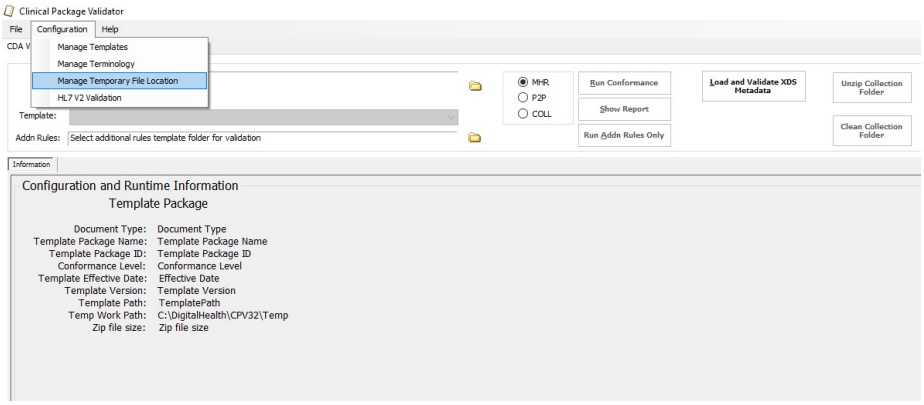


Figure 49 – Manage Temporary File Location

3 Use the folder icon to browse for the folder (Figure 50). Please note the text in RED. The Validator will delete every file and subfolder in this temporary folder, every time it runs. Do not set it to any folder that contains files you need to keep. To save the new temporary file location, select **Save** (Figure 51).

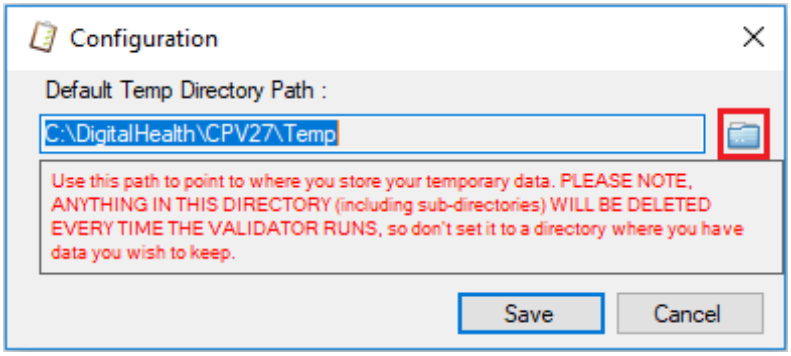


Figure 50 – use the folder icon

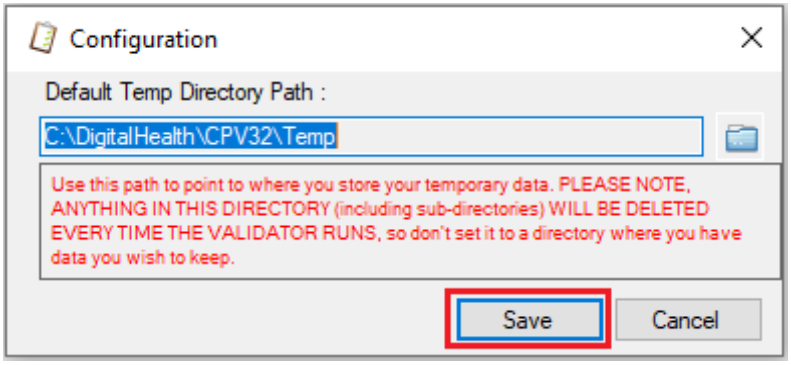


Figure 51 – select the Save button

3.2.7 Install a new version of the generic style sheet

1 The Validator is provided with version 1.6.0 of the generic style sheet (Generic CDA Style Sheet – stylesheet no CSS v1.6.0) to display clinical documents. This style sheet is available from the Agency's website.[4] To install a different version of the generic style sheet, stop the Validator if it is already running.

2 Navigate to the folder where the Validator is installed (usually **C:\DigitalHealth\CPV**xy, where "xy" refers to the Validator's version number "x.y") and select the **Stylesheets** subfolder (Figure 52).

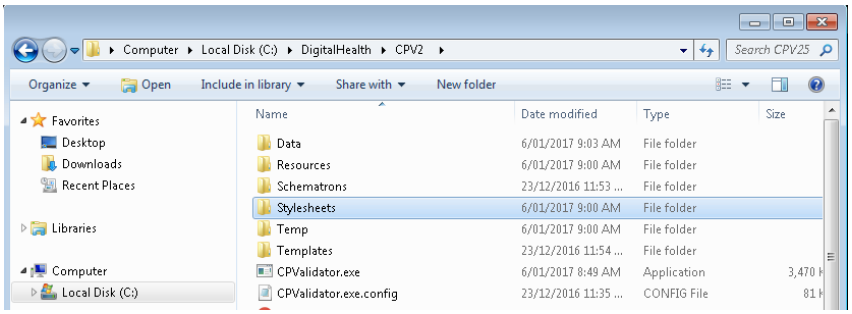


Figure 52 – The folder icon and the Save button

3 Locate and delete the **DH_Generic_CDA_Stylesheet** (Figure 53) file from this folder and replace it with the new style sheet, using the same name.

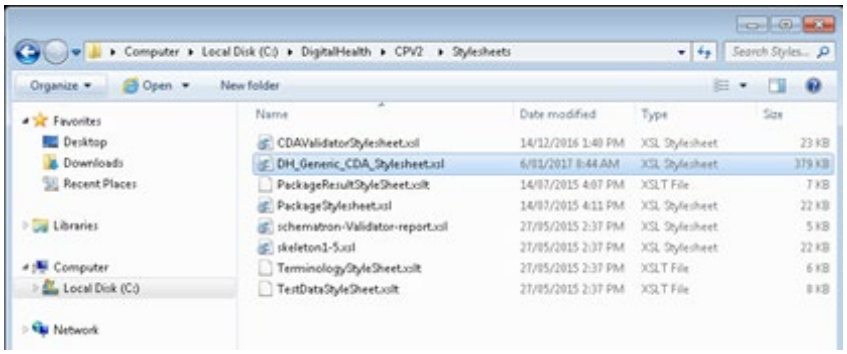


Figure 53 – DH Generic CDA Stylesheet

4 Start the Validator. The Validator will now use the new version of the style sheet.

[4] https://www.digitalhealth.gov.au/implementation-resources/ehealth-reference-platform/EP-2463-2017/DH-2467-2017

3.3 HL7 v2 validation

This section describes the HL7 v2 Validation functionality, including the location of the Profile and Coding files that are used by the Validator.

1 Go to Configuration and select HL7 V2 Validation (Figure 54)

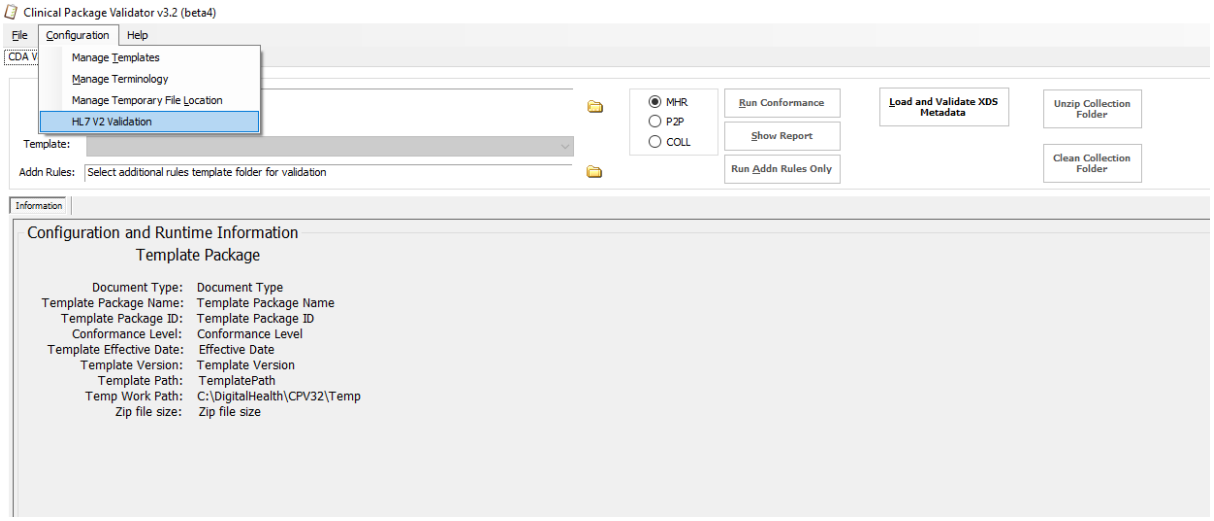


Figure 54 – select HL7 v2 Validation

2 Select the Open Folder Location buttons, to see the location of the Profile and Code files on your computer (Figure 55)

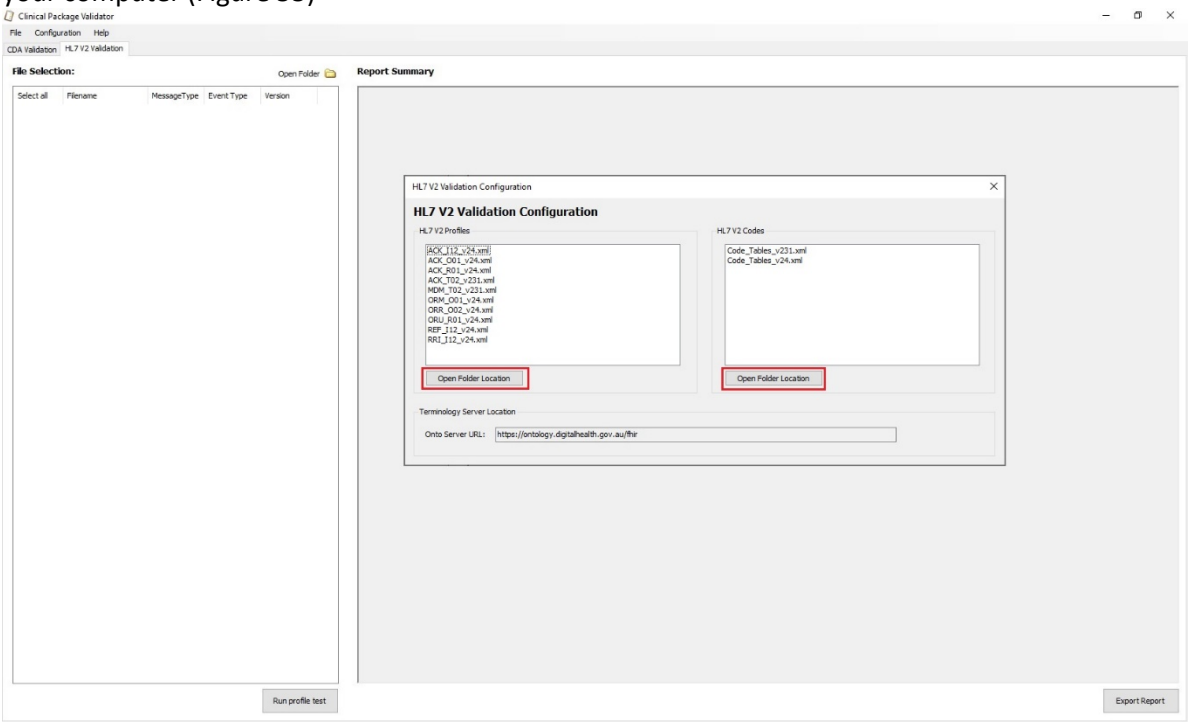


Figure 55 – Select Open Folder Location button

3 The location of the Profile files displayed (Figure 56) and the Code files displayed (Figure 57)

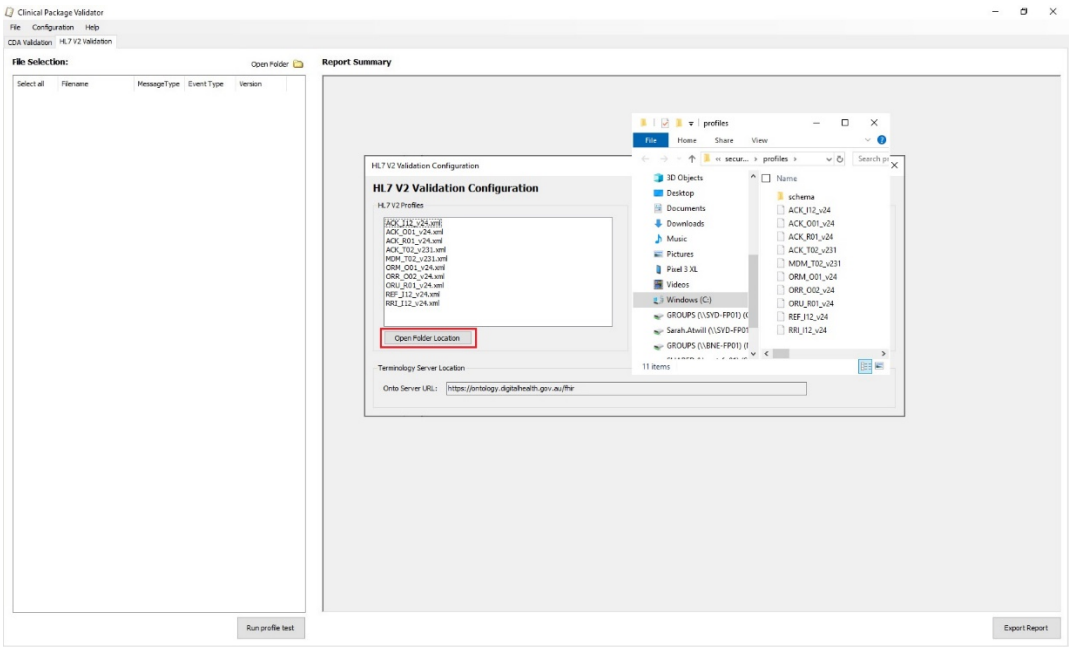


Figure 56 – location of Profile files

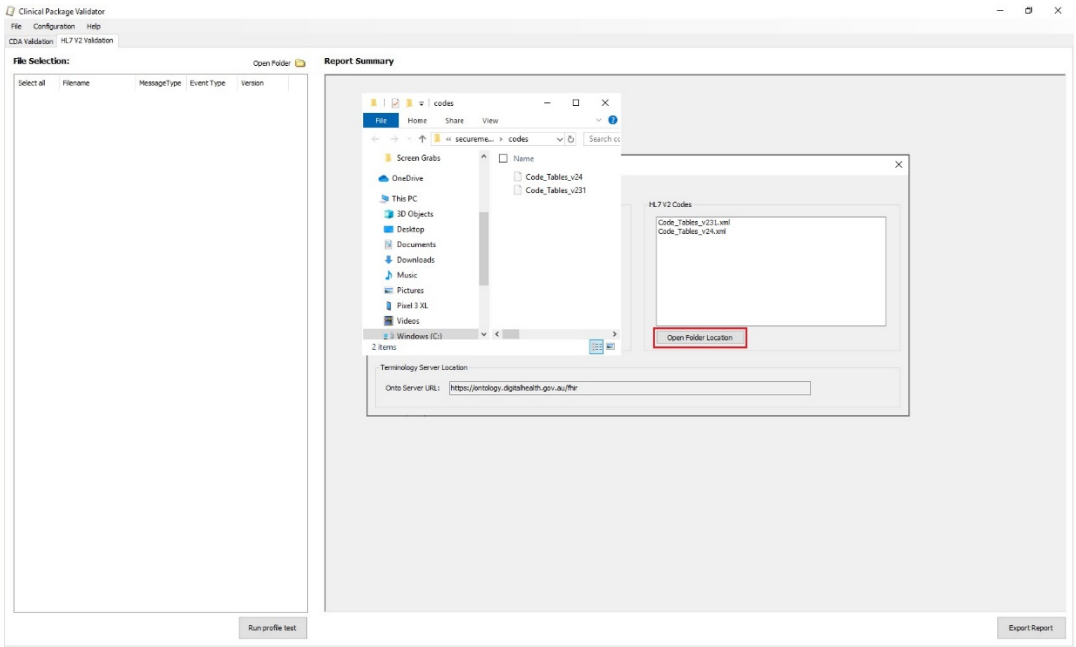


Figure 57 – location of Code files

4 The validator uses the Ontoserver for terminology checking. If the Ontoserver endpoint needs changing for any reason, please see Section 3.4 on how to change this.

3.4 Advanced Configuration

There may be cases where user need to change the ontoserver endpoint due to access restrictions, or wants to see what terminology checks are being made, or even building their own Schematron Rules and want to force compilation of these every time they are executed.

The Validator has a configuration file called CPValidator.exe.config in its running directory C:\DigitalHealth\CPV32 which allows certain changes to configuration.

Users are warned to be extremely careful when editing this file as if done incorrectly, could result in the Validator failing to run correctly. It is recommended that a copy of the config file is made before editing it, and any changes made require the Validator to be restarted to take effect.

The following table lists of the configuration changes that can be made in the <appSettings>:

Key	**Value**
tsLogging	Turns on/off Terminology logging to a file in the running directory called tsLog.txt Values can be 'true' or '**false'**
efLogging	Turns on/off Error logging to a file in the running directory called efLog.txt when errors are thrown while compiling schematron rules. Values can be 'true' or '**false'**
forceTemplatePackageSchematronRecompile	Turns on/off compiling of schematron template packages for every run. Values can be 'true' or '**false'**
forceAdditionalRulesSchematronRecompile	Turns on/off compiling of the additional rules schematron template packages for every run. Values can be 'true' or '**false'**
createAnalysisReport	Turns on/off creating the analysis.tsv file when running schematron template packages Values can be '**true'** or 'false'
ontoServerUrl	The URL of the ontoserver that is used to check terminology. Value should default to **https://api.healthterminologies.gov.au/integration/v2/fhir**

4 Using the Validator

This section describes the menus, parameters, commands and other features of the Validator. It also describes the results and reports generated by the Validator.

4.1 Validator menus

This section describes the purpose of each Validator menu option.

Note: Gaps in the file paths within the following screenshots should feature your username or equivalent when following the documented procedures.

4.1.1 File menu

The Validator will close if the Exit menu option is selected (Figure 58).

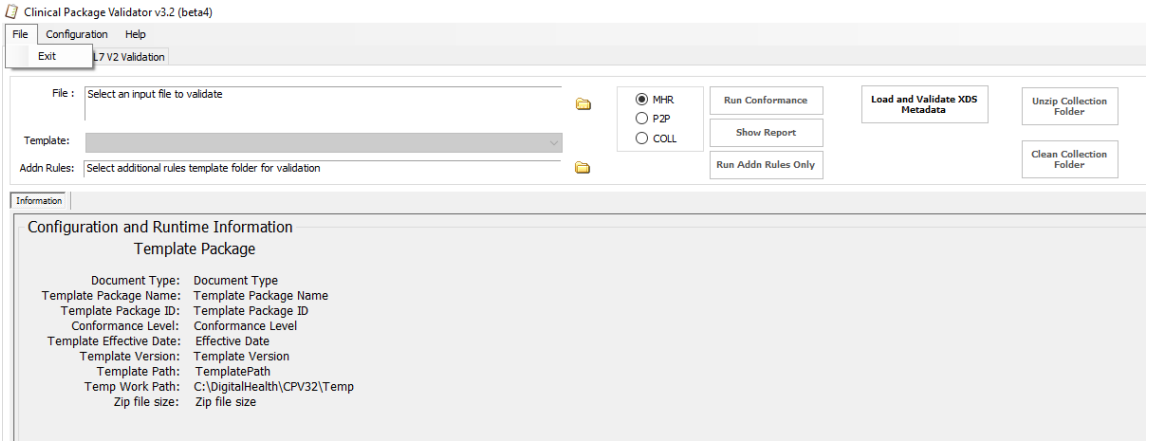


Figure 58 – File -> Exit menu option

4.1.2 Configuration menu

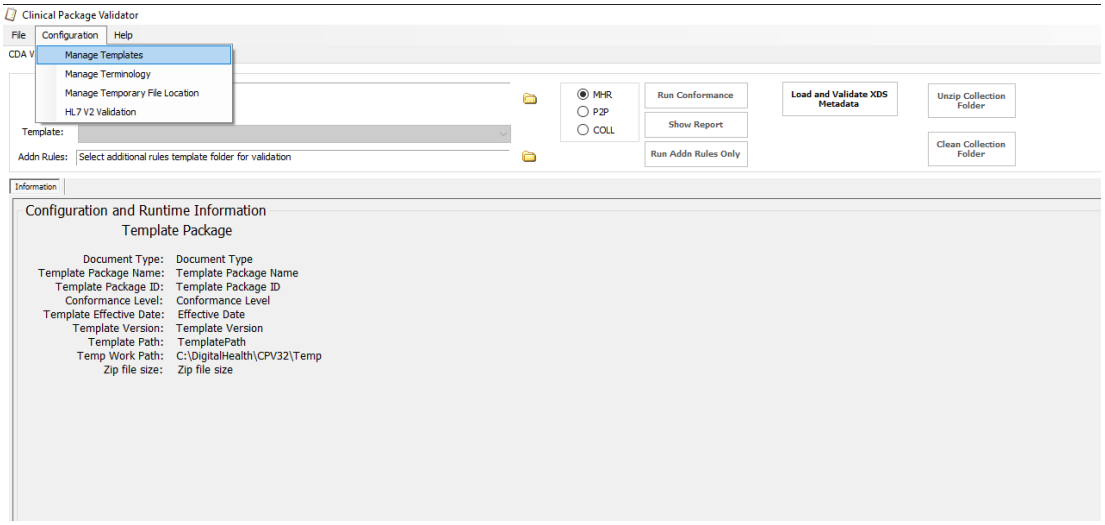


Figure 59 - Configuration menu options

4.1.3 Template Management

This menu option allows the user to import and manage My Health Record templates for validating clinical documents (Figure 60).

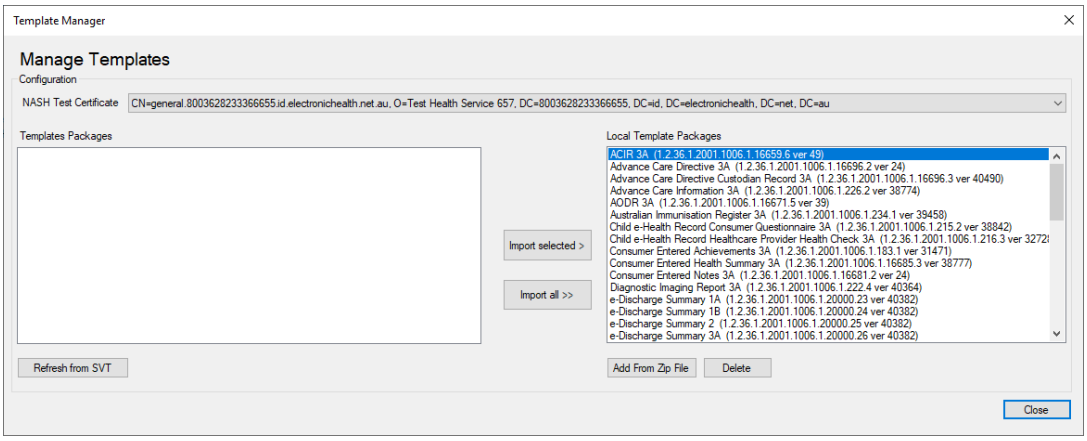


Figure 60 - Template management menu

4.1.4 Terminology Management

This menu option allows the user to see which terminology datasets are installed, load new terminology datasets, and remove terminology datasets (61).

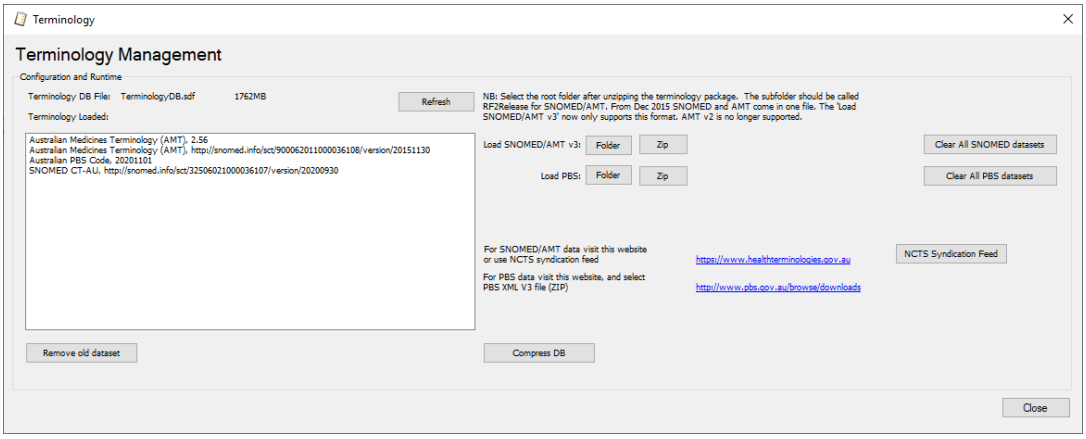


Figure 61 - Terminology management menu

4.1.5 Terminology Management

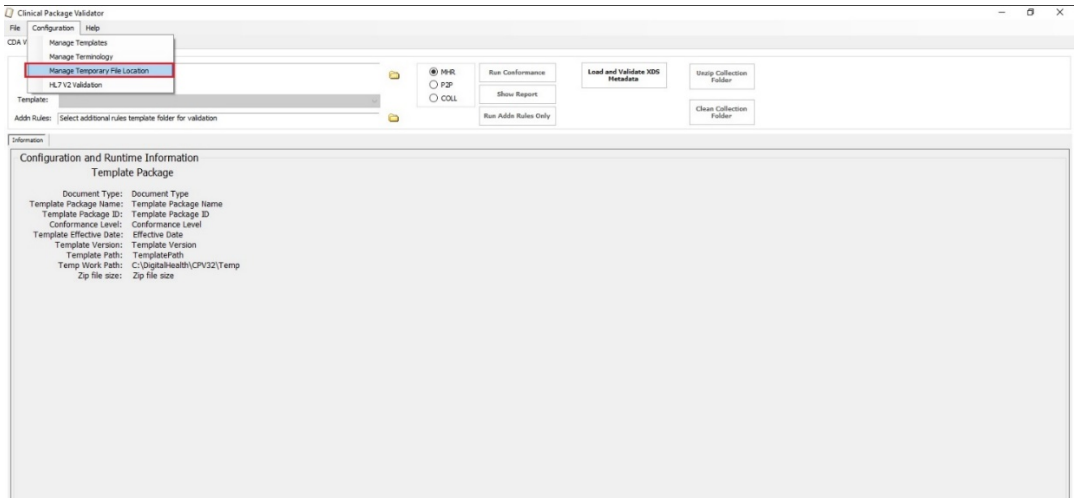


Figure 62 – Manage Temporary File Location

4.1.6 Default Temp Directory Path

This menu option allows the user to change the location used for storing temporary files (Figure 63). More information is described in the *Clinical Package Validator Installation and Configuration Guide*. Please note the text in RED. This highlights that the Validator will delete every file and subfolder in this temporary folder, every time it runs. So, do not set it to any folder that contains files you need to keep.

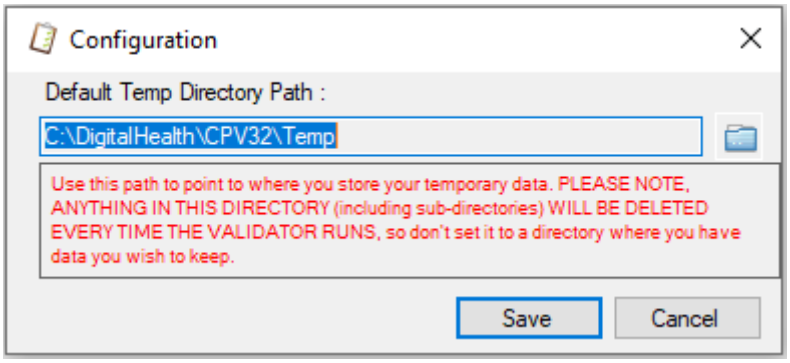


Figure 63 - Default temp directory path menu

4.1.7 HL7 V2 Validation

This menu option is described in section 3.3

4.1.8 Help menu

This menu option (Figure 64) displays the version number, copyright, and acknowledgements for the Validator

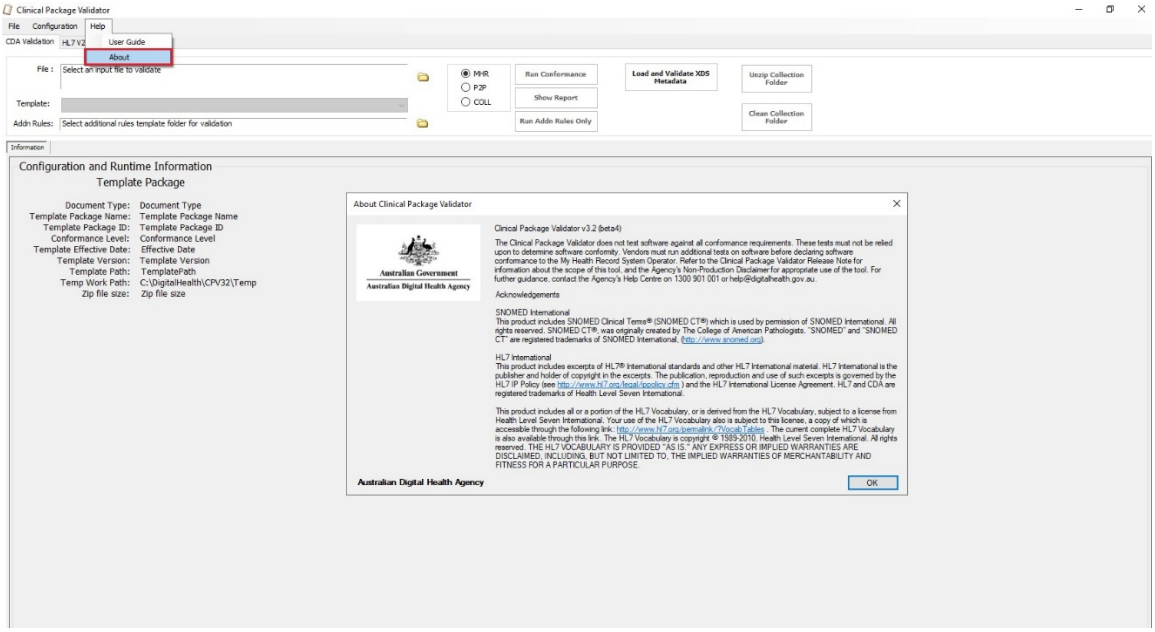


Figure 64 – Help menu

4.2 CDA validation

4.2.1 Validator parameters

Before the Validator can validate a clinical document or clinical package, some parameters must first be set. If the Validator has been installed and configured according to the *Clinical Package Validator Installation and Configuration Guide*, the only parameter that may need to be set is the clinical document or package to be validated. The full set of parameters that may be set are listed in Table 1. Please refer to the *Clinical Package Validator Product Data Sheet* for more details on the scope of template package validation.

Table 1 - Validator parameters

Parameter	When to set
Clinical document or package to be validated	Always.
Template to be applied	If there is more than one template to select from.
Additional template to be applied	If an additional template package is to be applied.
Context	If the context is not the default value of 'My Health Record (MHR)'.

4.2.1.1 File parameter

The File parameter displays the location of the document that is to be validated (Figure 65).

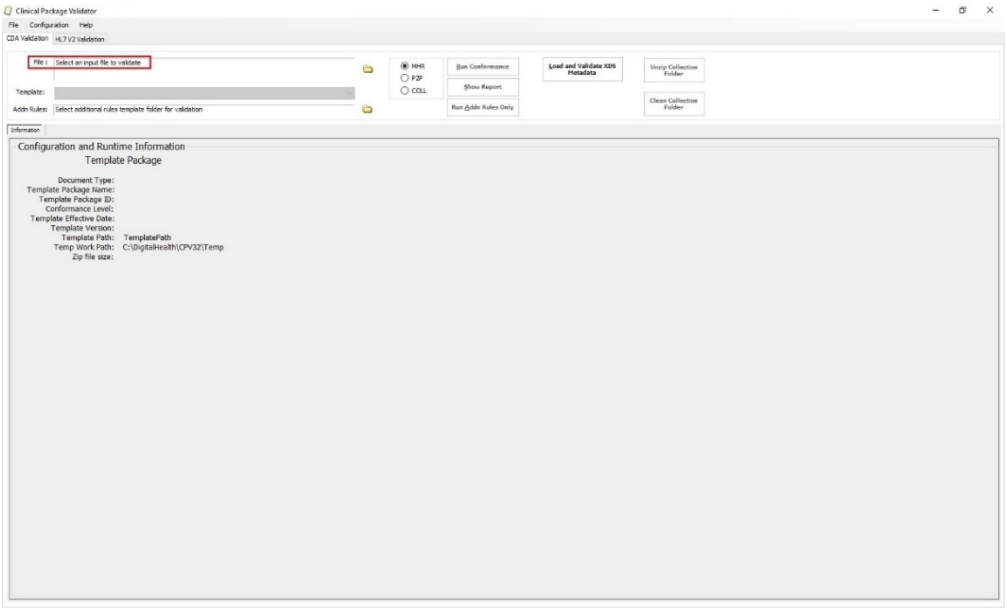


Figure 65 - File parameter

The Validator uses the filename extension to determine the type of the file to be validated:

- a file with the '.xml' filename extension is assumed to be a clinical document;
- a file with the '.zip' filename extension is assumed to be a clinical package; and
- a file with the '.hl7' or 'txt' filename extension is assumed to be an HL7v2 MDM wrapped clinical package.

Two options are provided for selecting the file that is to be validated.

Option one

The first option is to select the file by browsing. Click the **File Open** button (Figure 66).

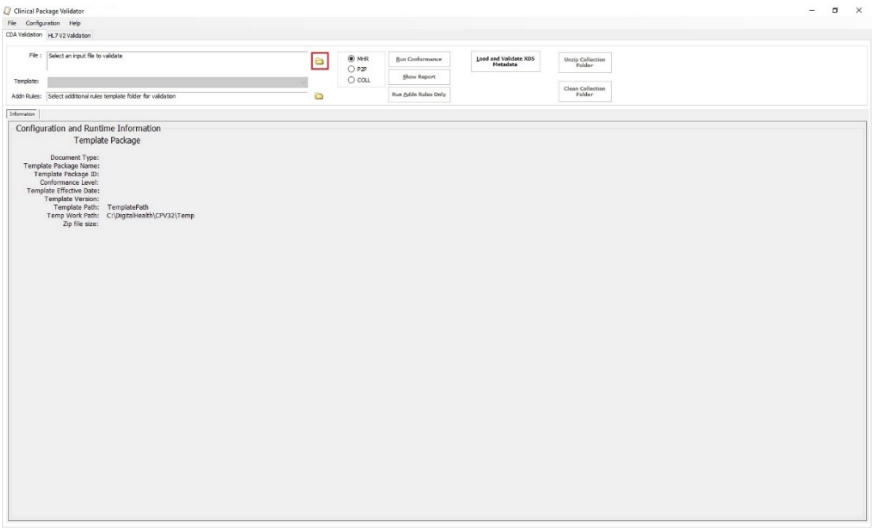


Figure 66 - Selecting the File open button

Navigate to the location of the clinical document, clinical package or HL7 MDM wrapped clinical package, select the file and click **Open** (Figure 67).

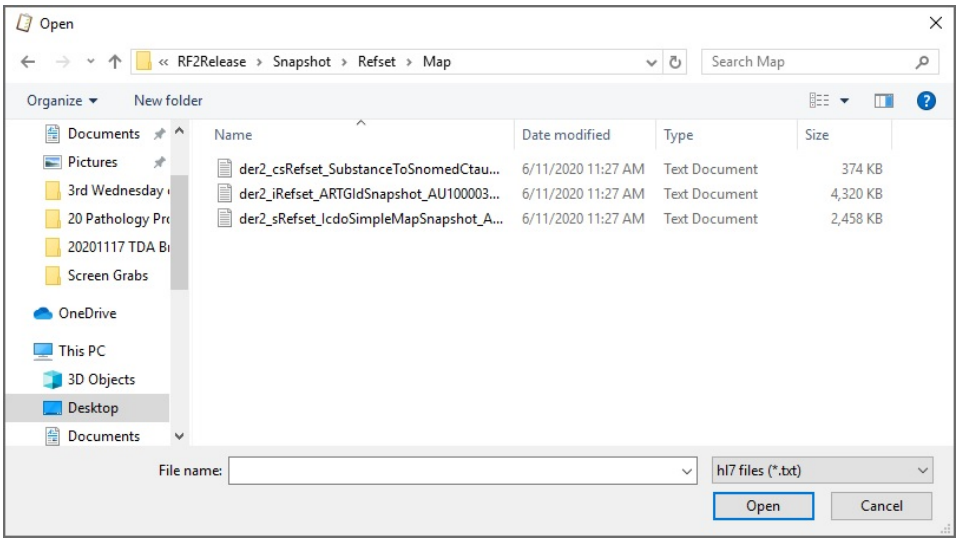


Figure 67 - Selecting the file to be validated

Option two

The second option for selecting the file to be validated is to drag the file and drop it into either the **File** field or the **Information** tab (Figure 68).

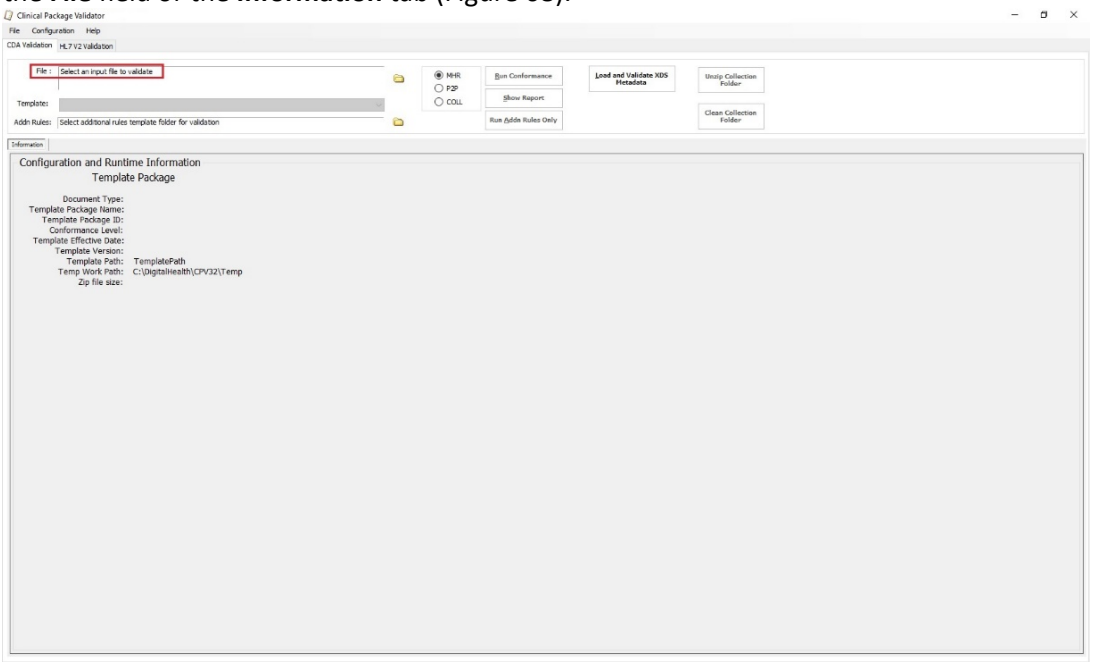


Figure 68 - Selecting the file to be validated by dragging and dropping it to the Information tab

The Validator automatically examines the content of the file to determine the type of the clinical document being validated. The types of clinical documents that are known to the Validator are listed in section 2.1.2. The clinical document type is displayed in the Document Type field in the **Information** tab (Figure 69).

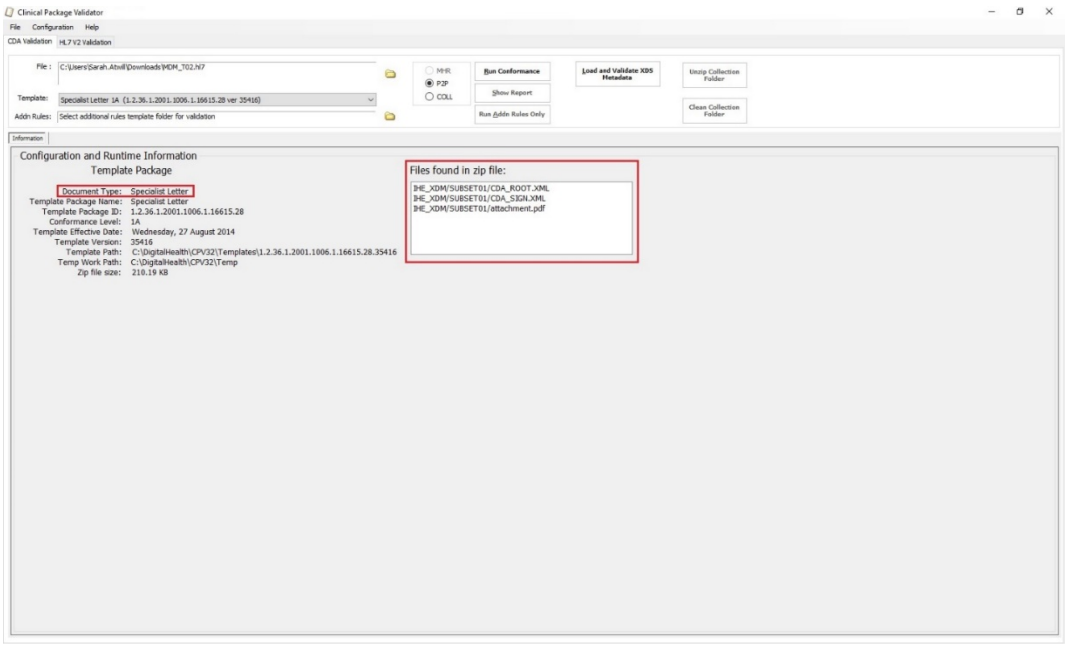


Figure 69 - Document Type field displaying the type of the clinical document being validated

If the Validator cannot determine the type of the clinical document being validated, an error message is displayed (Figure 70).

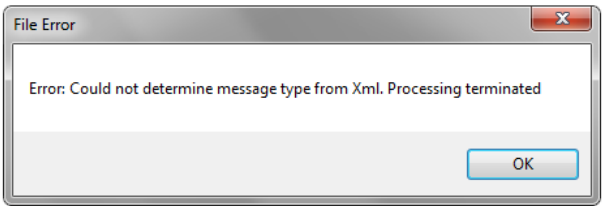


Figure 70 - Error message when the type of a clinical document is unknown

The Validator may display other error messages if a severe error prevents it from continuing (Figure 71 and Figure 72).

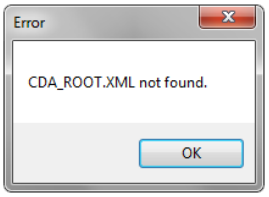


Figure 71 - Error message when opening a ZIP file that is not a clinical package

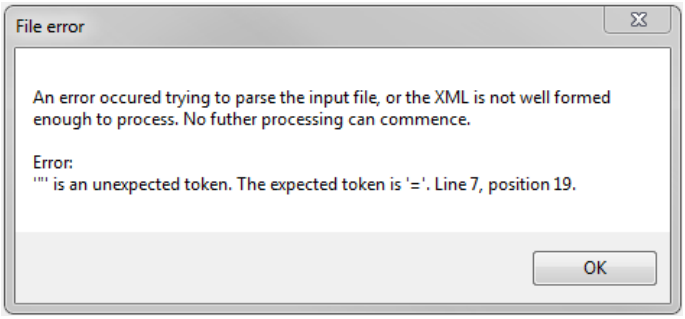


Figure 72 - Error message when opening an XML file that does not conform to the HL7 CDA R2 specification

The Validator will display a warning message if it does not have a template to validate the type of clinical document that has been imported (Figure 73).

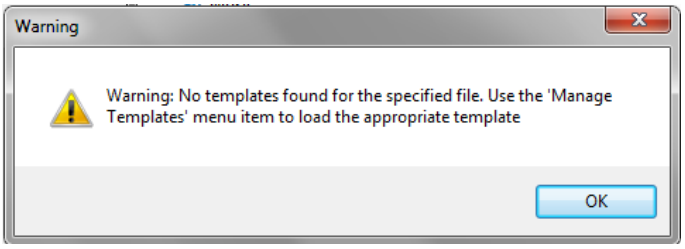


Figure 73 - Warning message when there is no template for the type of clinical document

The user can continue to still run an Additional Rules template even if a document template does not exist.

4.2.1.2 Template parameter

The **Template** parameter displays the My Health Record template to be used for validating the selected clinical document (Figure 74).

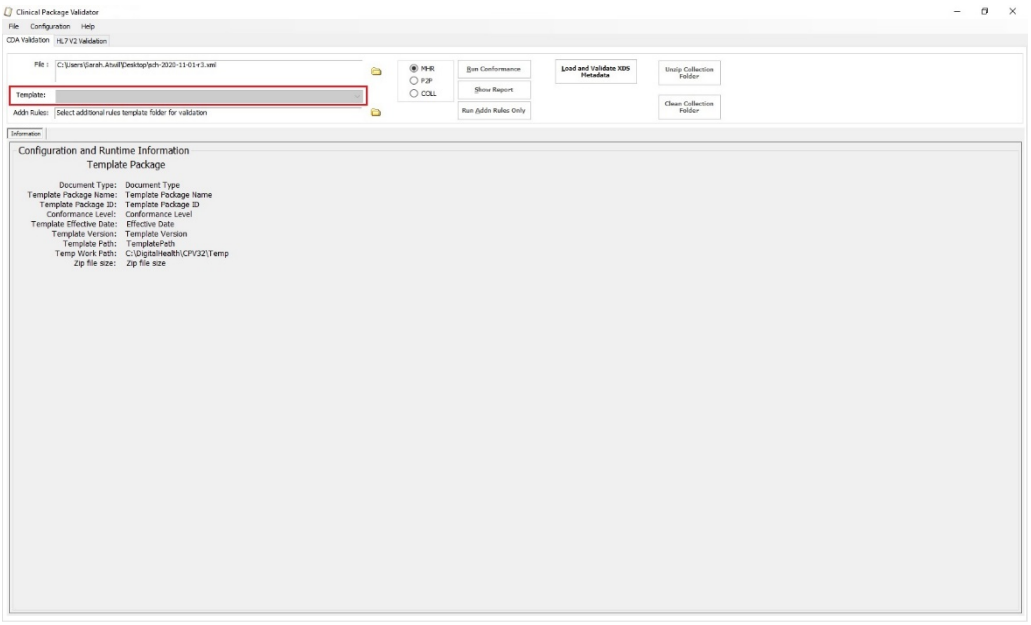


Figure 74 - The My Health Record template to be used for validation

The **Template** field has no information before a clinical document is selected. Once a document is selected, the Validator will determine the type of the selected document.

The Validator needs a My Health Record template for each type of clinical document at each level of conformance. As templates are revised, more than one template may exist for a specific clinical document type and conformance level. Testers should use the most recent version of a template unless there is a reason for choosing an earlier version. A template package directory is published on the Agency website[5].

If the Validator has more than one template for the type of document to be validated, the **Template** field will contain a drop-down list of all templates that may be applied so that the user can select a template (Figure 75).

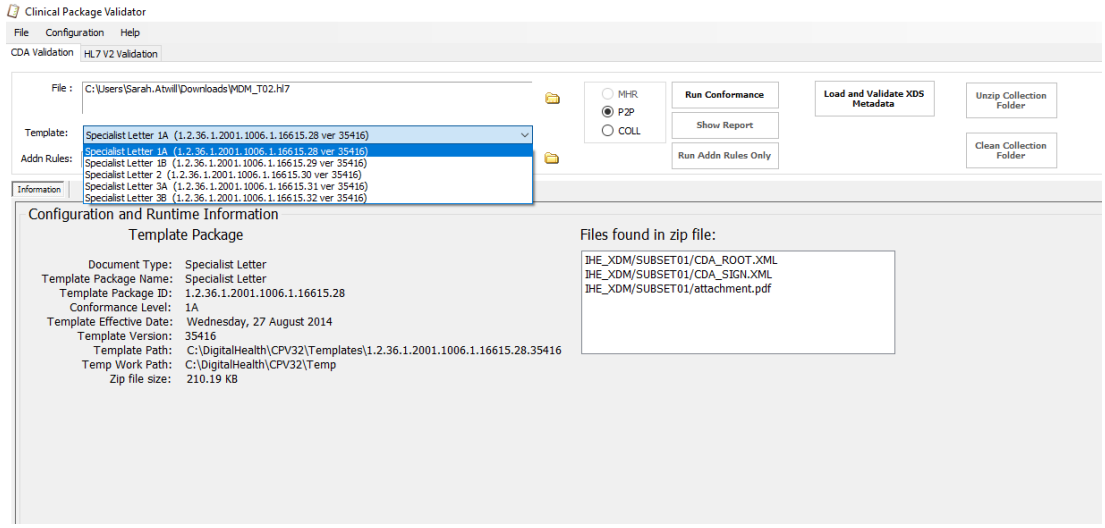


Figure 75 - Selecting a template

The **Template** field displays the document type, conformance level and template identifier for each template. Detailed information about the selected template will be displayed in the **Information** tab (Section 4.2.3).

4.2.1.3 Addn Rules parameter

The **Addn Rules** parameter displays the additional template to be used for validating a clinical document (Figure 76). Information about the role of an additional template is provided in section 2.1.2.

[5] https://www.digitalhealth.gov.au/implementation-resources/clinical-documents/EP-2320-2016/NEHTA-2321-2016

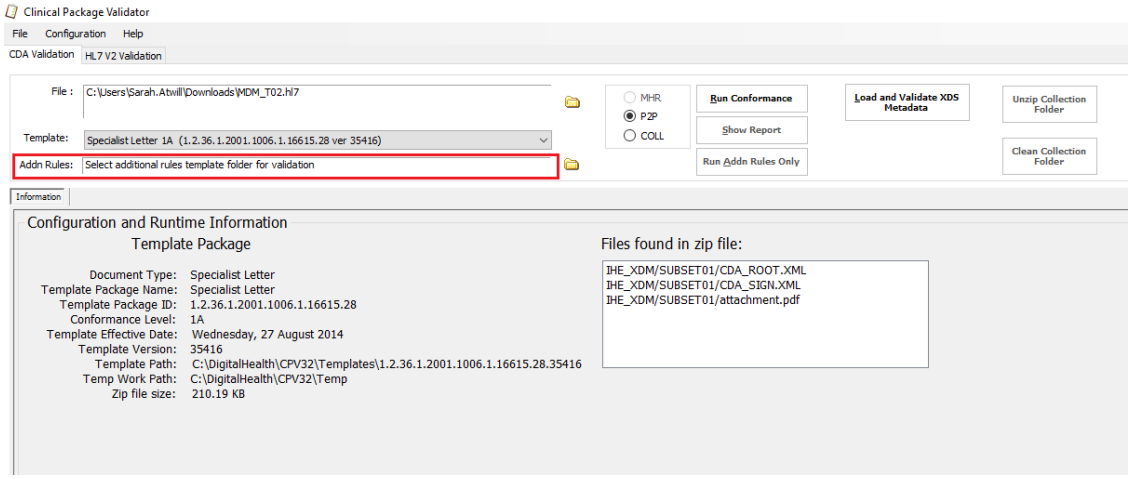


Figure 76 - The additional rules validation template

Two options are provided for selecting the location of an additional template.

Option one

The first option is to select the additional template by browsing. Click the folder button (Figure 77).

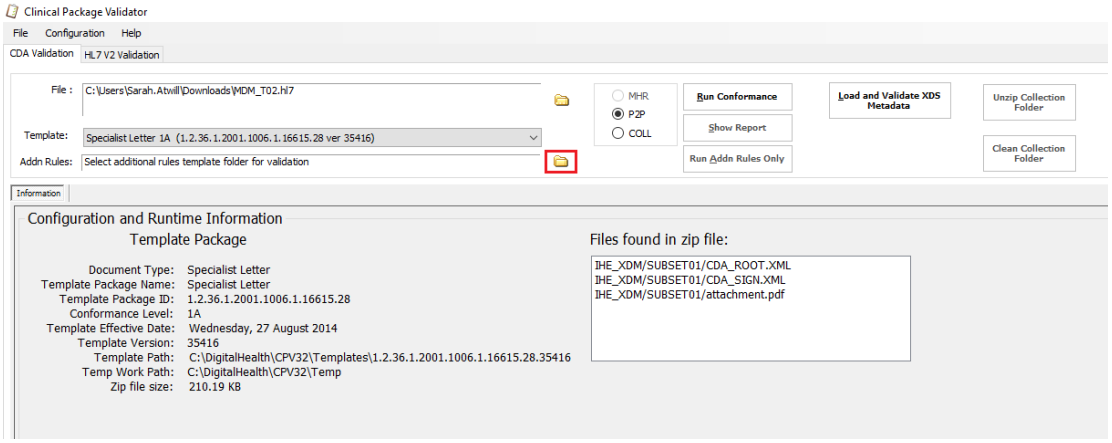


Figure 77 - Selecting the folder button

Navigate to the directory in which the additional template is stored (Figure 78).

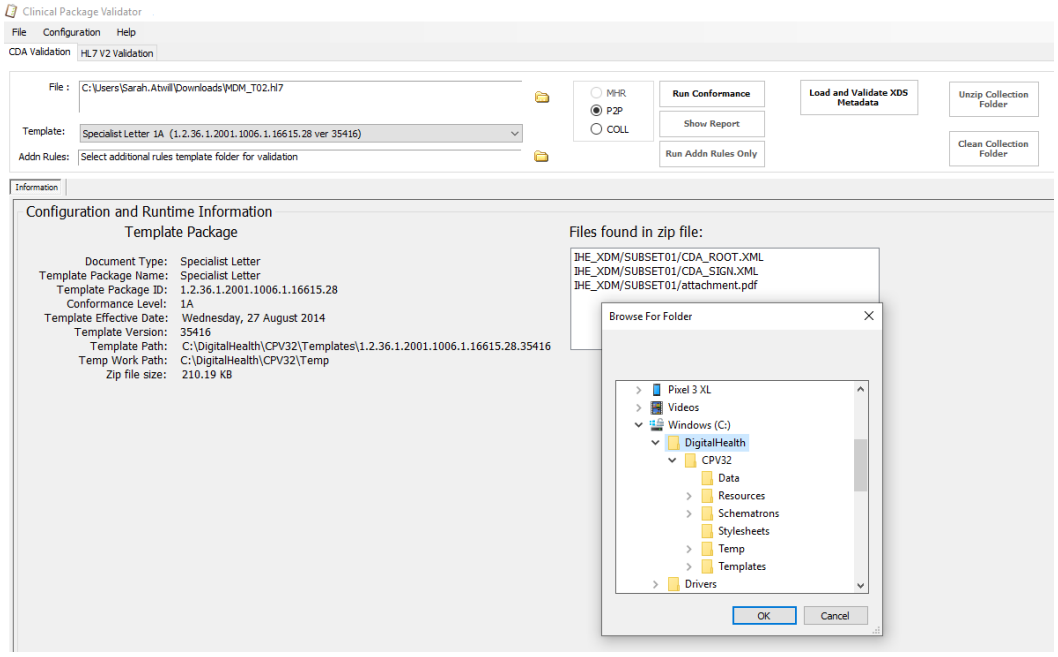


Figure 78 - Navigating to the directory containing the additional template

Select the additional template (Figure 79).

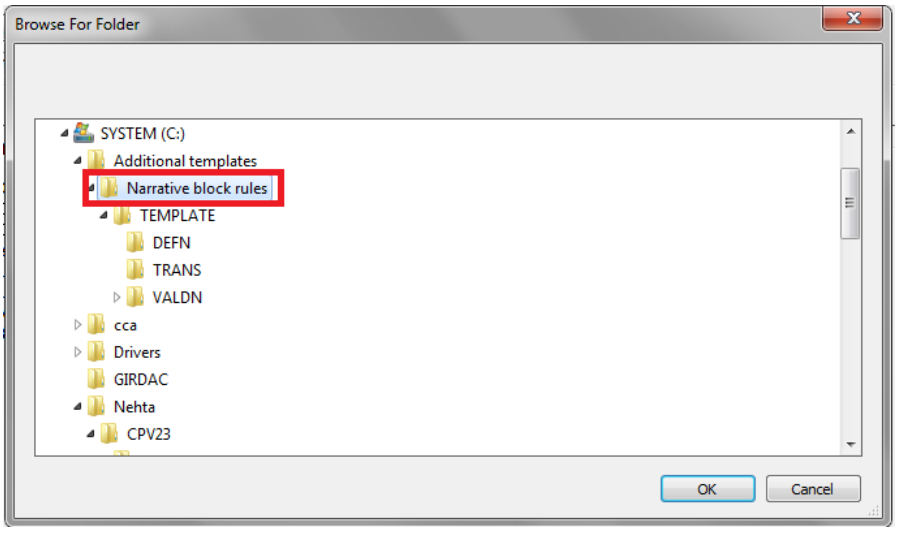


Figure 79 - Locating and selecting the additional template

In Figure 79, “C:\Additional templates\Narrative block rules” is the subdirectory in which the additional template was stored.

It is important to select the “Additional templates\Narrative block rules” subdirectory rather than the “Additional templates\Narrative block rules\TEMPLATE” subdirectory. The “Additional templates\Narrative block rules” subdirectory is referred to as a template package. It contains the additional template, an index and a README file (Figure 80).

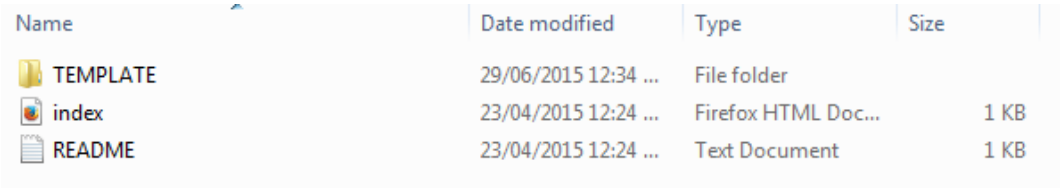


Figure 80 - Contents of a template package

An error message is displayed if the combined length of the template location and folder name is more than 80 characters (Figure 81).

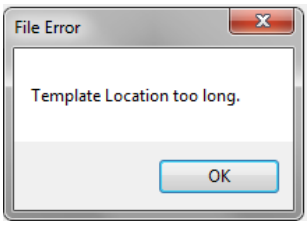


Figure 81 - Error message when the combined length of the template location and folder name is too long

Option two

The second option for selecting the location of an additional template is to drag the template folder and drop it into the Addn Rules field (Figure 82).

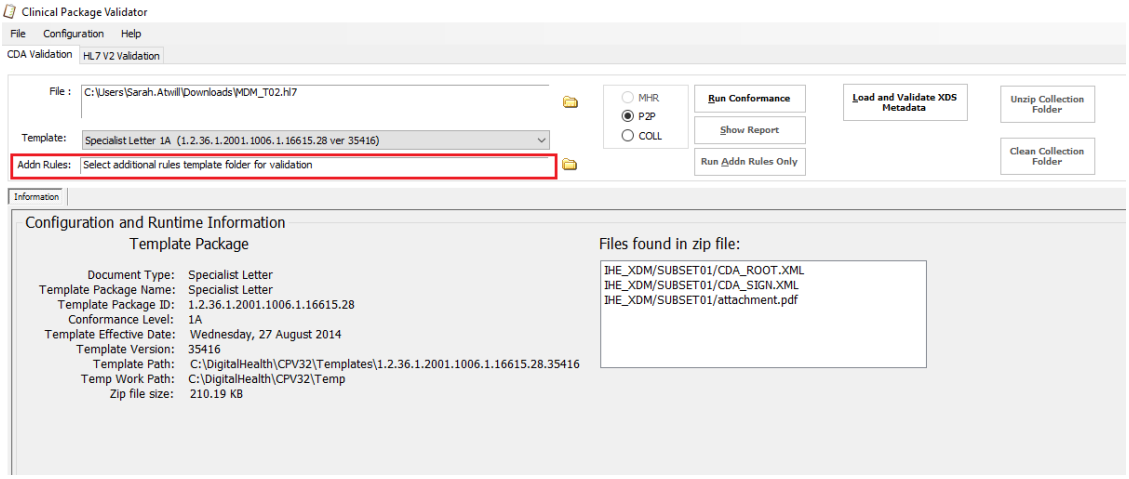


Figure 82 - Dragging and dropping an additional template folder into the Addn Rules field

4.2.1.4 Context parameter

The Validator requires a user to select the context for validating a clinical package. The selection of context determines which tests will be applied when a clinical package is validated (Section 2.1). The Validator supports the My Health Record (MHR) P2P and COLL contexts:

- The My Health Record context is the set of tests that apply to a clinical package sent to the My Health Record system.

- The P2P context is the set of tests that apply to a clinical package sent from one healthcare provider to another.
- The COLL context may be used by Additional Rules to assess a collection of documents, whereas the MHR and P2P contexts apply to a single document.

The default context for the Validator is My Health Record (MHR).

If a file with the '.hl7' filename extension is imported into the Validator, it is assumed to be an HL7 MDM wrapped clinical package and the Validator changes to the P2P context, as the My Health Record system does not support HL7 MDM. For all other types of files, the P2P context must be explicitly selected (Figure 83).

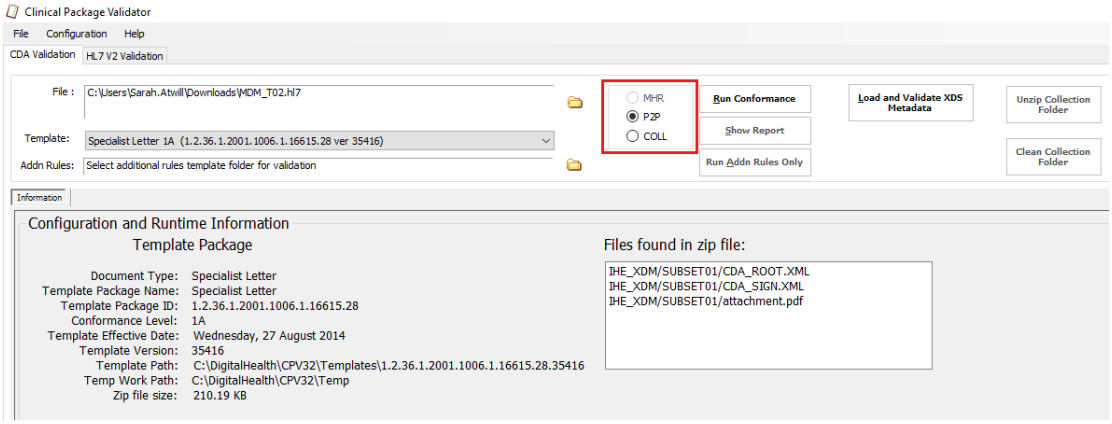


Figure 83 - Selecting the P2P context

The third context is "COLL", short for Collection, which is a special case that can only be used with Additional Rules, such as the IQ Rules Template Package, that are designed to assess collections. It assumes that you are wanting to compare a collection of documents which are located in the C:\DigitalHealth\ClinicalDocuments\Collection Folder.

When "COLL" is selected, a "dummy" file called "CDA-Empty.xml" is created in the C:\DigitalHealth\ClinicalDocuments folder which is required to enable the running of IQ Rules.

For further information, read the IQ Rules User Guide [IQRules].

4.2.2 Conformance levels

The conformance level against which a clinical document is to be validated is not explicitly selected. Instead, the user selects the My Health Record template package that applies to the type of clinical document and conformance level against which the clinical document is being assessed (Section 4.2.1.2).

Detailed descriptions of each conformance level can be found in *Clinical Documents - Common Conformance Profile* [ADHA2015c]. In general, the conformance levels are 1A, 1B, 2, 3A, and 3B. The conformance levels that actually apply depend on the type of the clinical document. The allowed conformance levels for each type of clinical document are listed in the relevant conformance profile in the end product published on the Agency website[6] (e.g. the event summary conformance profile is part of the Event Summary end product).

[6] http://www.digitalhealth.gov.au/implementation-resources/clinical-documents

4.2.2.1 Conformance level 1A

A conformance level 1A clinical document has an XML header, and an XML body containing only a caption and a reference to an attached file. The administrative details form part of the header. In this example, the referral is an Adobe PDF file and the link to the referral is all that is contained in the XML body.

In a conformance level 1A clinical document, the data that would normally appear in the body of the document is instead included in an attached file. The Validator can only validate data that appears in the header of the document.

The rendered view of a conformance level 1A Specialist Letter is shown in Figure 84. If the referral is an image (e.g. a JPEG file), the image may be displayed in the body of the rendered document, depending on the type of image.

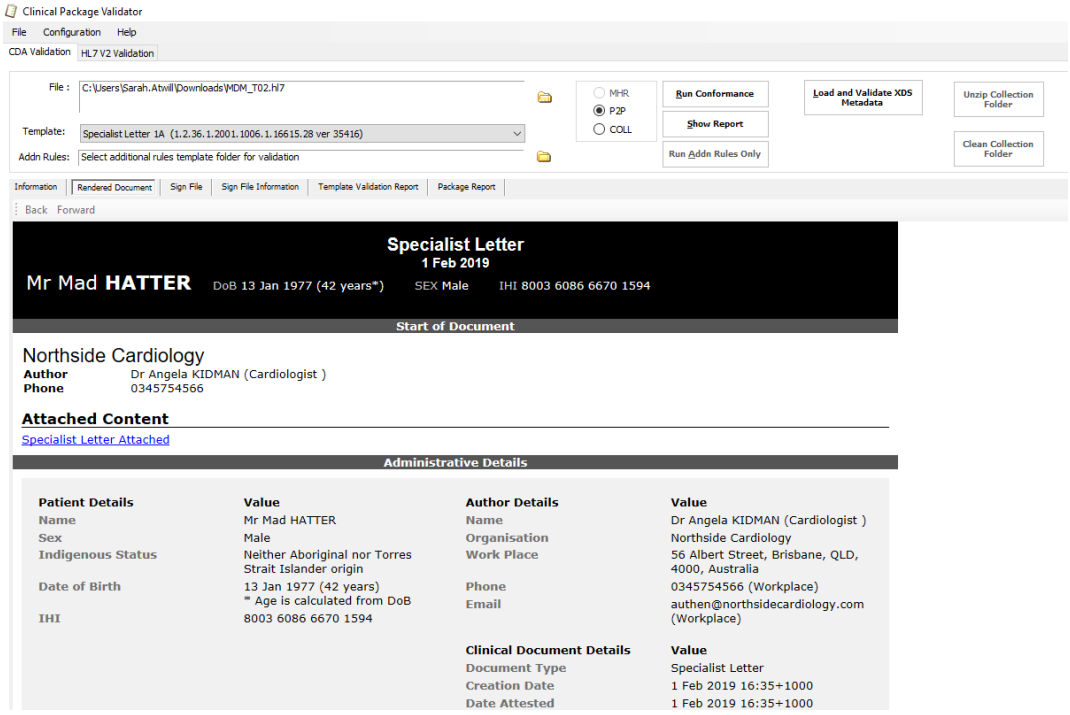


Figure 84 - Rendered view of a level 1A Specialist Letter

4.2.2.2 Conformance level 1B

A conformance level 1B clinical document has an XML header, and an XML body that includes at least one section that contains a label and a narrative block with clinical information. The data for the body is included in the narrative block(s) and may also appear as structured data.

As structured data is not required in the body of a level 1B clinical document, any structured data in the body will not be validated by the Validator using a conformance level 1B template. Any terminology codes from supported code systems will however be validated.

The rendered view of a conformance level 1B Specialist Letter is shown in Figure 85. There is only one section in the body of this rendered Specialist Letter, although level 1B clinical documents may have more than one section. The label of the section is Title and the text below the label is the rendered text from the narrative block.

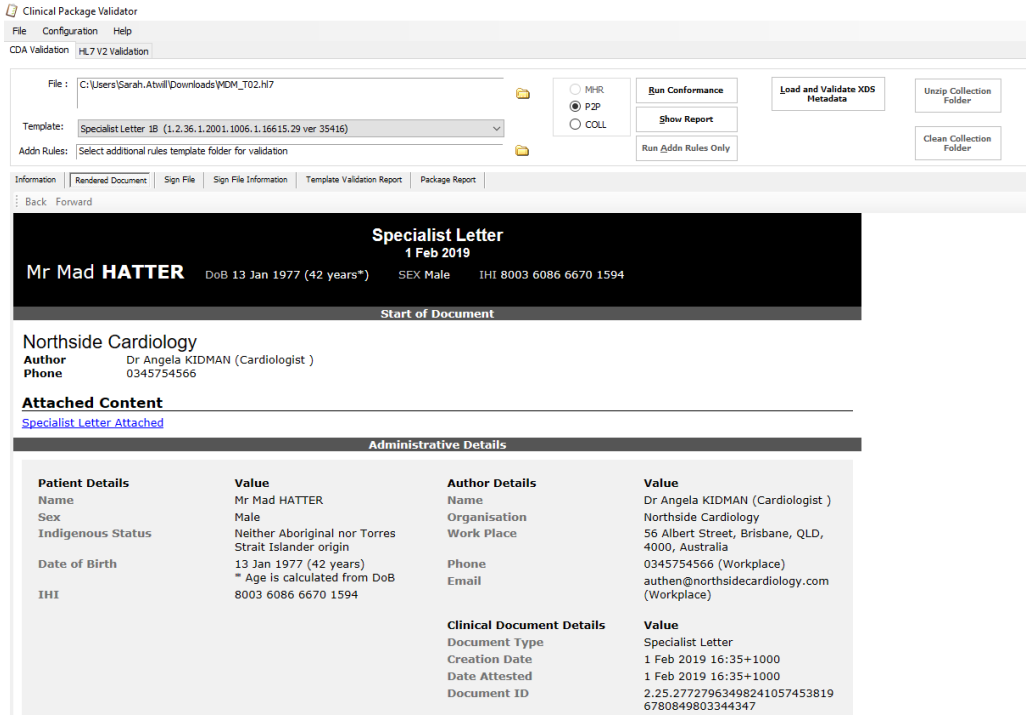


Figure 85 - Rendered view of a level 1B Specialist Letter

4.2.2.3 Conformance level 2

A conformance level 2 clinical document has an XML header and an XML body that contains all the mandatory sections specified in the relevant CDA implementation guide. Each mandatory section has a label and a narrative block with clinical information. The data for the body is included in the narrative blocks and may also appear as structured data.

As structured data is not required in the body of a level 2 clinical document, any structured data in the body will not be validated by the Validator using a conformance level 2 template. Any terminology codes from supported code systems will however be validated.

The rendered view of a conformance level 2 Specialist Letter is shown in Figure 86. Each section has a label and a narrative block.

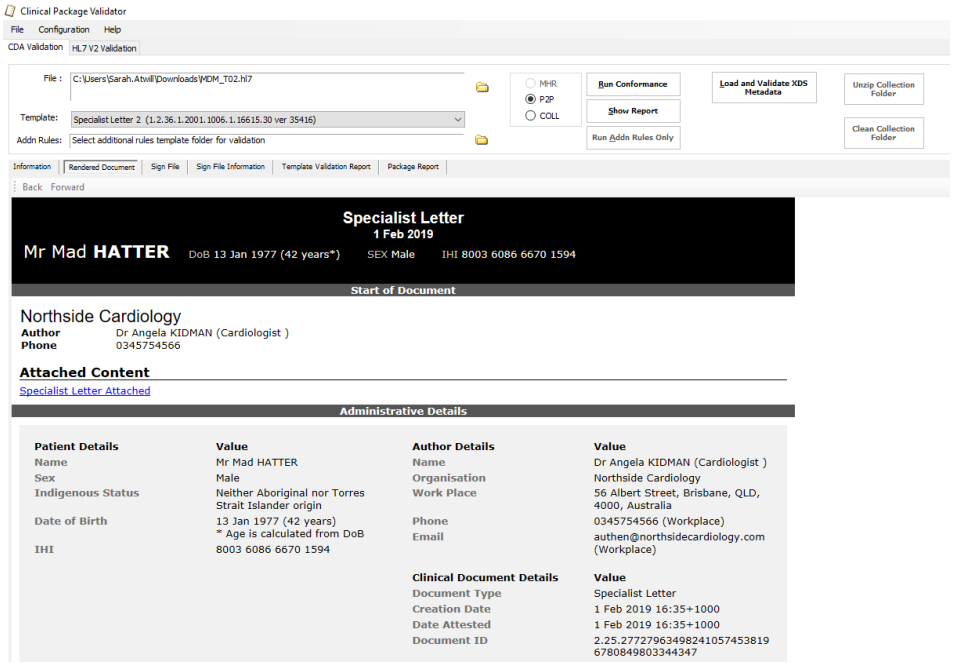


Figure 86 - Rendered view of a level 2 Specialist Letter

4.2.2.4 Conformance level 3A

A conformance level 3A clinical document has an XML header, and an XML body that contains all the mandatory sections specified in the relevant CDA implementation guide. Each mandatory section has a label and a narrative block with clinical information. In addition, each section has structured clinical information for mandatory data elements.

A rendered conformance level 3A clinical document is identical to a rendered conformance level 2 clinical document. Similarly, the XML files are identical except the XML file of the level 3A document contains additional structured data.

4.2.2.5 Conformance level 3B

A conformance level 3B clinical document has an XML header, and an XML body that contains all mandatory sections specified in the relevant CDA implementation guide. Each mandatory section has a label and a narrative block with clinical information. In addition, each section has structured clinical information for mandatory data elements.

A conformance level 3B clinical document must contain codes from specified code systems, which is optional for lower levels of conformance. Terminology codes such as AMT, SNOMED CT-AU, or PBS must be present wherever they are allowed in the level 3B clinical document.

A rendered conformance level 3B clinical document is identical to a rendered conformance level 2 and 3A clinical document. The XML file of a level 3B clinical document is identical to the XML file of a level 3A clinical document, except the XML file of the level 3B document must also contain codes from the specified code systems.

4.2.3 Configuration and runtime information

The **Information** tab displays information about the type of the clinical document being validated and information about the template (Figure 87) as well as the content of the zip file (if one chosen).

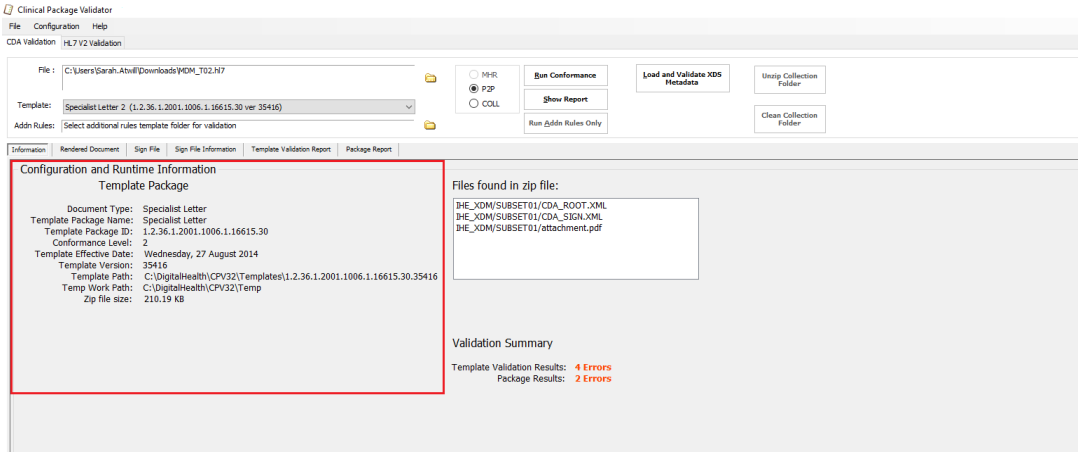


Figure 87 - Configuration and runtime information

The information displayed is described in Table 2.

Table 2 - Configuration and runtime information displayed in the Information tab

Label	Description
Document Type	The type of the clinical document to be validated.
Template Package Name	The name of the template package as recorded in the template package metadata.
Template Package ID	The identifier of the template package as recorded in the template package metadata.
Conformance Level	The conformance level as recorded in the template package metadata.
Template Effective Date	The date the template package was approved as recorded in the template package metadata.
Template Version	The version number of the template package as recorded in the template package metadata.
Template Path	The location of the template package.
Temp Work Path	The location used by the Validator to store temporary files.
Zip file size	The size of the zip file (if the file is a zip file).

If both the Document Type and the Template Package Name fields display names of differing types of documents, an incorrect template package has been selected.

Note: Some document names have changed over time, so it is possible that the Document Type and Template Package Name fields may display different names for the same type of document. This is not a problem, and validation can proceed normally.

If an additional template is selected, the **Information** tab displays information about the additional template (Figure 88).

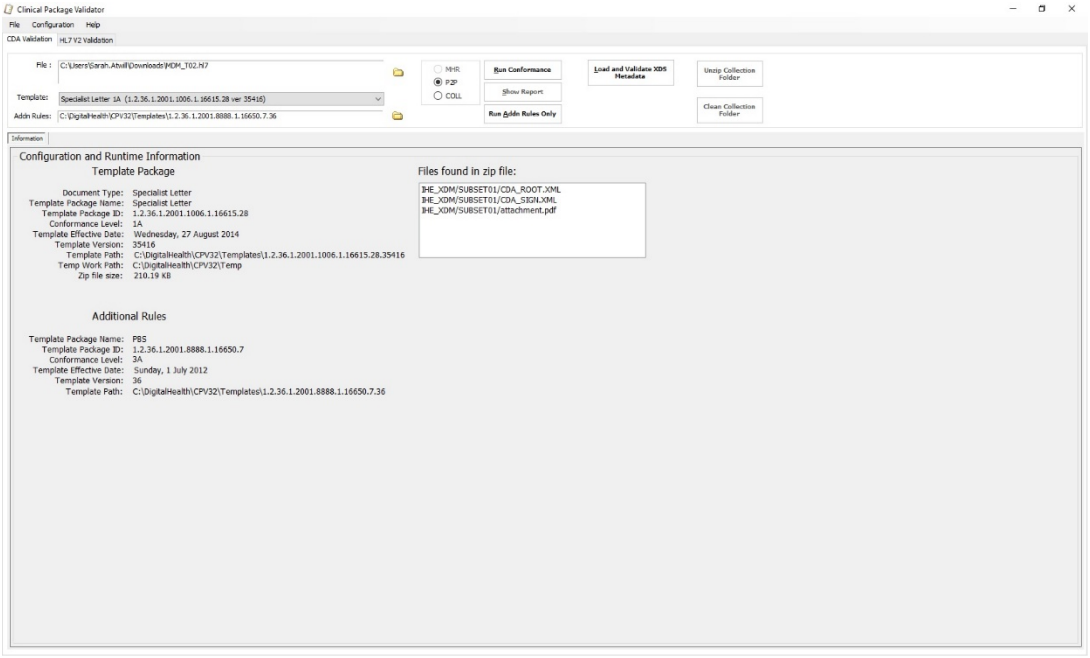


Figure 88 - Information about an additional template

The information displayed about the additional template is described in Table 3.

Table 3 - Information about the additional template

Label	Description
Template Package Name	The name of the additional template package as recorded in the template package metadata.
Template Package ID	The identifier of the additional template package as recorded in the template package metadata.
Conformance Level	The conformance level as recorded in the additional template package metadata.
Template Effective Date	The date the additional template package was approved as recorded in the template package metadata.
Template Version	The version number of the additional template package as recorded in the template package metadata.
Template Path	The location of the additional template package.

The “Files found in zip file” box only appear if a CDA Package has been selected. It lists the files found in the zip file.

4.2.4 Run Conformance command

The **Run Conformance** command allows a user to request the Validator to test whether a clinical document or clinical package (for which the Validator may be used) conforms to the relevant specifications (Figure 89), subject to the limitations set out in the Clinical Package Validator Product Data Sheet.

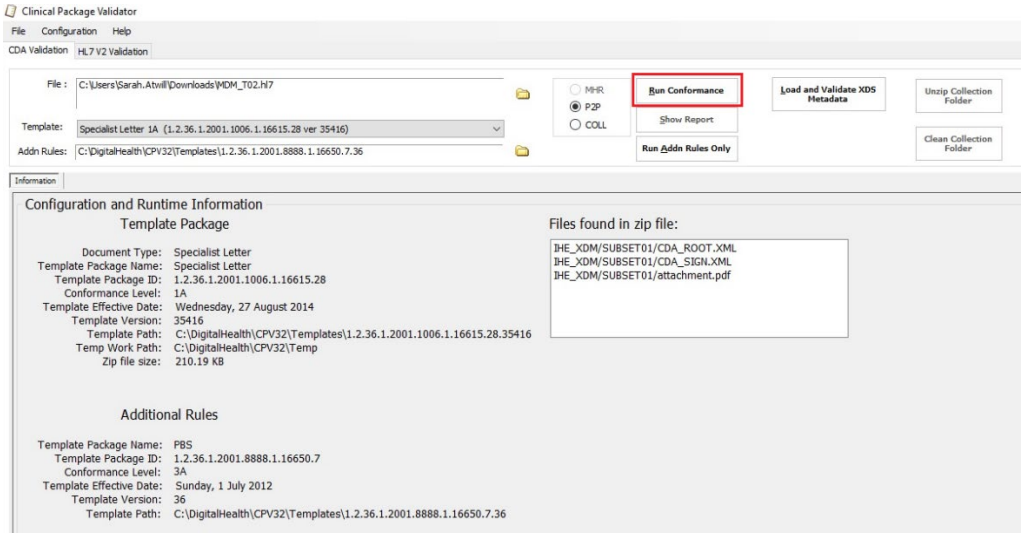


Figure 89 - Requesting the Validator to test the conformance of a clinical document or clinical package

The tests that are performed vary with the type of object being validated:

1 If the object is a clinical package (i.e. a .zip file), the package will be validated as well as the clinical document within the clinical package.

2 If the object is an HL7 MDM wrapped clinical package (i.e. .hl7 file), the clinical package is extracted and both the clinical package and the clinical document within it are validated.

3 If the object is a clinical document (i.e. an .xml), the clinical document is validated but package validation is not performed.

Template validation is applied to every type of supported clinical document. Validation results are displayed on the graphical user interface (Section 3.7).

4.2.5 Run Addn Rules Only command

The Run Addn rules Only command allows a user to request the Validator to just run the additional rules against the clinical document.

Validation results are displayed on the graphical user interface (Section 4.2.6).

4.2.6 Validation results

The following tabs are used to display the results of a clinical document or clinical package validation (for which the Validator may be used).

4.2.6.1 Information tab

In addition to displaying the information listed in Section 4.2.3, the **Information** tab also contains a summary of the validation results.

*Table 4 - Summary of validation results displayed in the **Information** tab*

Label	Description
Template Validation Results	The total number of errors reported by template validation.
Package Results	The total number of errors reported by package validation. This information is only included if a clinical package was validated.

Label	Description
Reference Set Results	The total number of errors and warnings reported by validation of codes against reference sets, where specified in CDA implementation guide.
Other Terminology Results	The total number of errors and warnings reported by validation of codes against supported code systems.
Additional Rules	The total number of errors reported by validation using the additional template. This information is only included if additional rules have been applied.

4.2.6.2 Rendered Document tab

The **Rendered Document** tab displays the rendered clinical document (for which the Validator may be used) that is being validated (Figure 90).

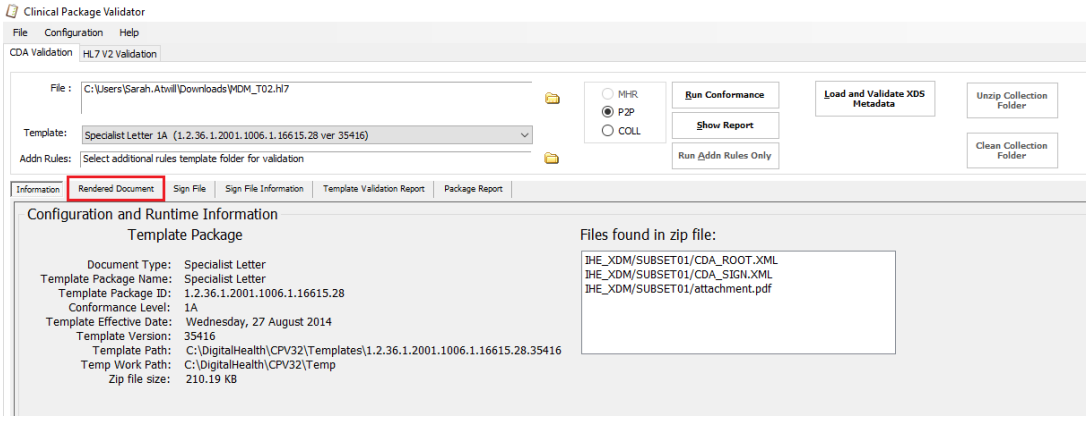


Figure 90 - Rendered Document tab

Figure 91 is an example of a rendered clinical document displayed on the **Rendered Document** tab. The **Back** and **Forward** buttons allow the user to go back to the rendered view of the clinical document, or forward from the rendered view of the clinical document to the attachment.

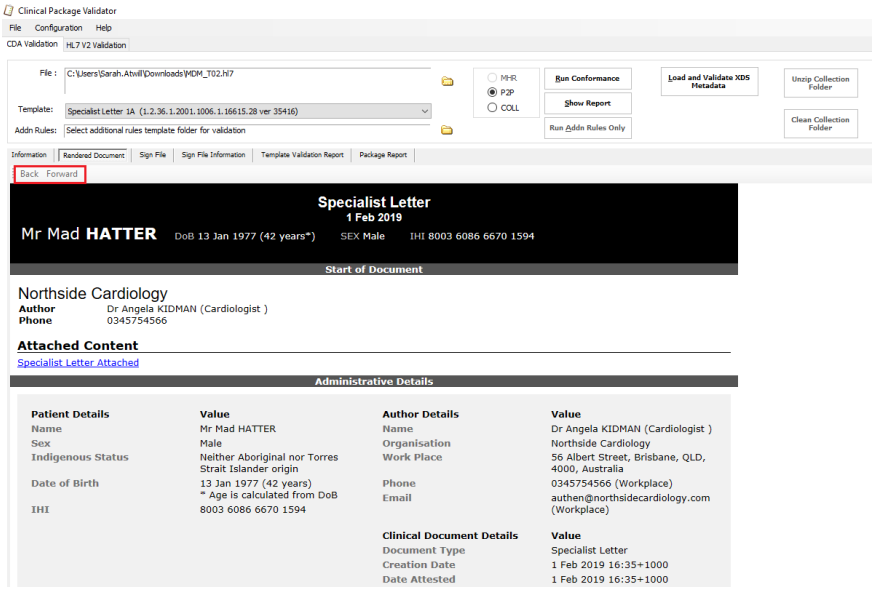


Figure 91 - Rendered clinical document displayed on the Rendered Document tab

4.2.6.3 Sign File tab

The **Sign File** tab displays the contents of the CDA_SIGN.XML file in a clinical package (Figure 92). The **Sign File** tab is only displayed when the file being validated is either a clinical package or an HL7 Medical Document Management (MDM) wrapped clinical package.

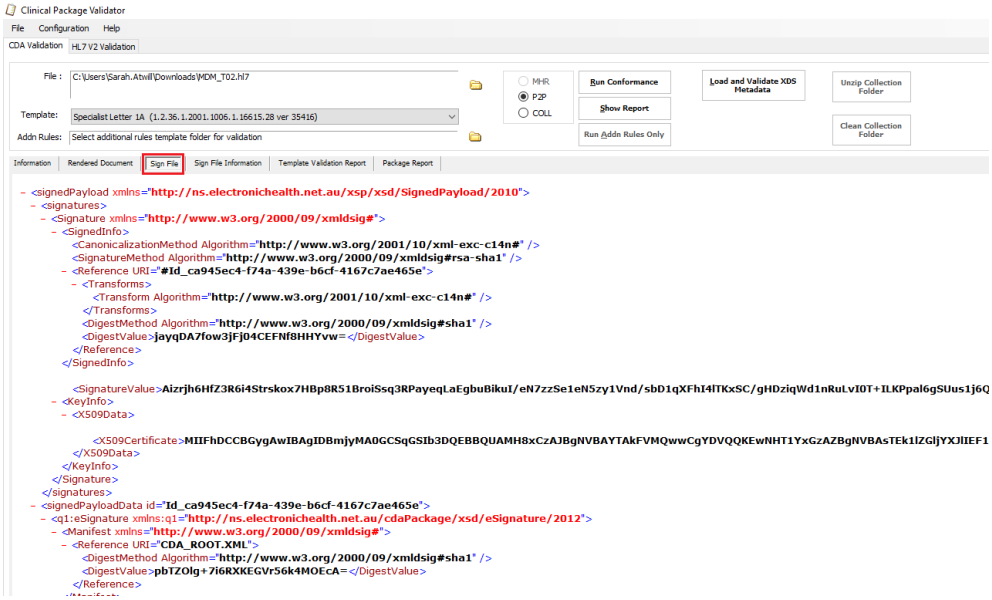


Figure 92 - Sign File tab

Figure 93 is an example of a CDA_SIGN.XML file that is displayed on the **Sign File** tab.

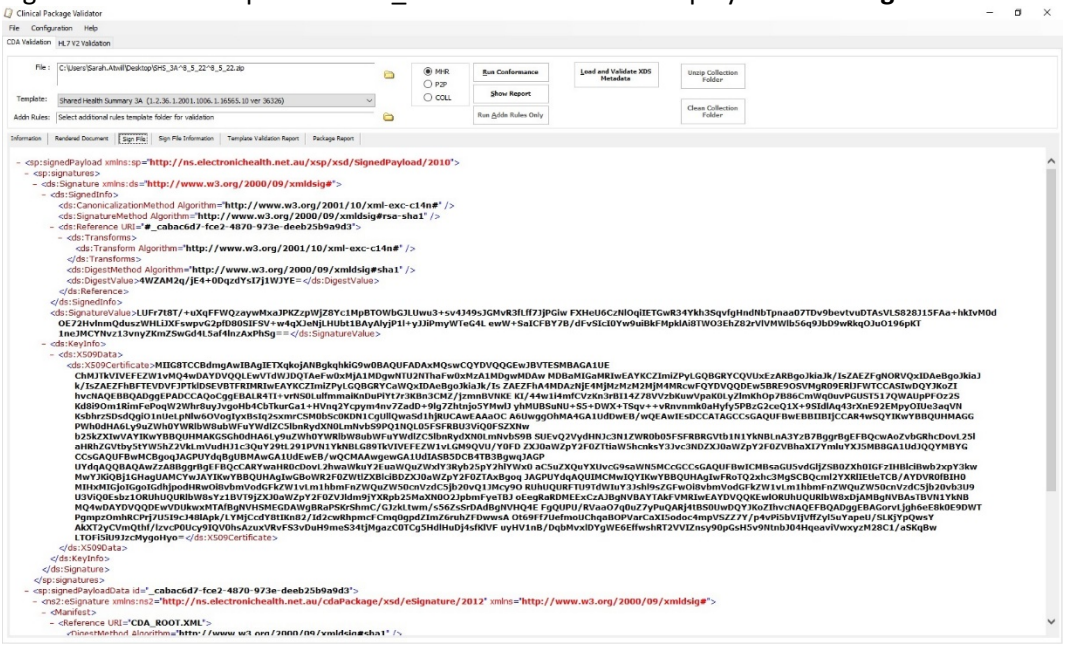


Figure 93 - Display of a CDA_SIGN.XML file

4.2.6.4 Sign File Information tab

The **Sign File Information** tab displays selected information from the CDA_SIGN.XML file in a clinical package (Figure 94). This complements information in the **Sign File** tab as this information is more readable. Information from the Public Key Infrastructure (PKI) certificate is also displayed.

The **Sign File Information** tab is only displayed when the file being validated is either a clinical package or an HL7 Medical Document Management (MDM) wrapped clinical package.

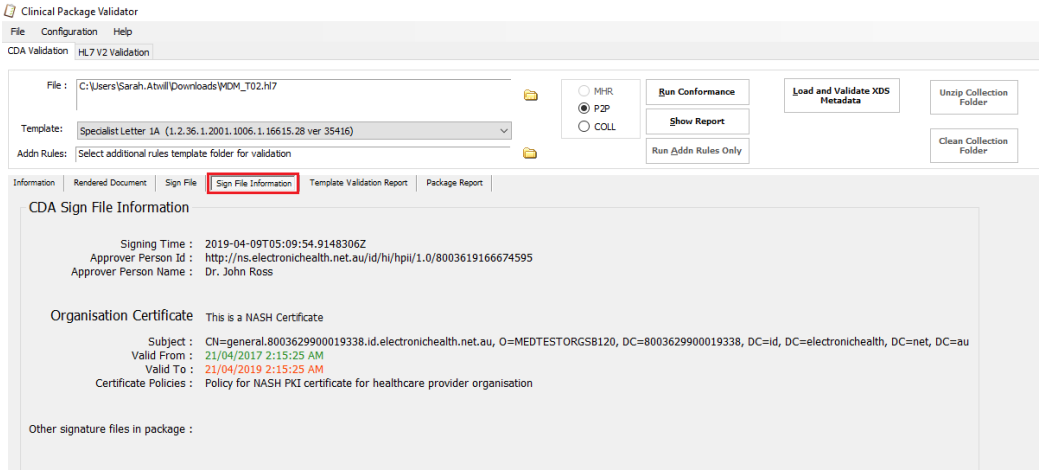


Figure 94 - Sign File Information tab

Figure 95 is an example of the information displayed on the **Sign File Information** tab

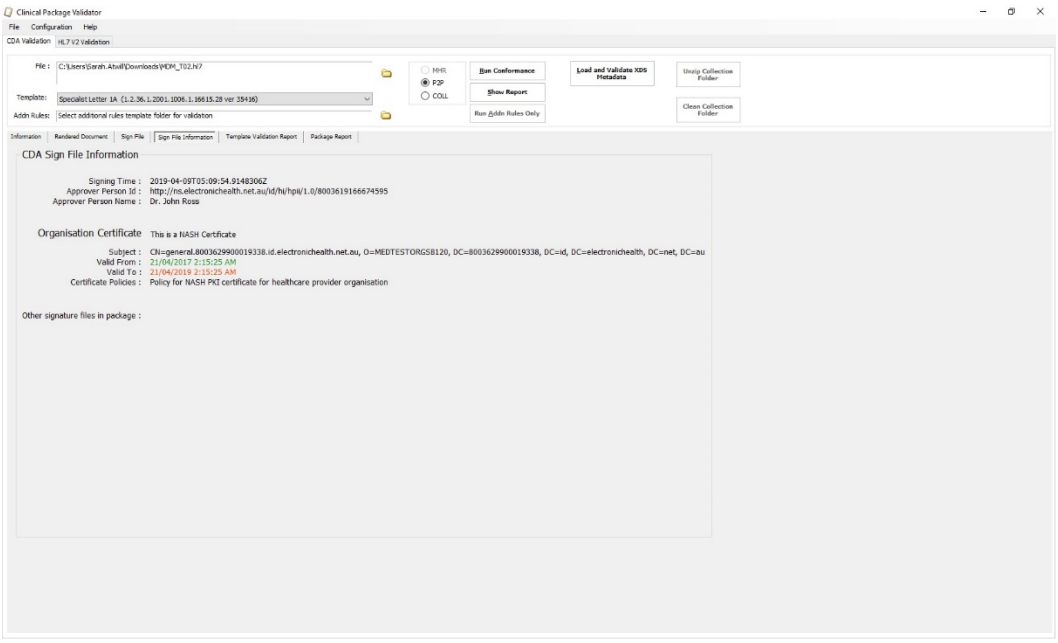


Figure 95 - Display of eSignature file information

The information displayed is listed in Table 5.

Table 5 - Summary of information displayed in the Sign File Information tab

Label	Description
Signing Time	The date and time that the clinical document was signed using the PKI certificate.
Approver Person Id	The identifier of the approver, in the form of a Uniform Resource Identifier (URI). The approver is the person that reviewed and approved the content of the clinical document [ADHA2015c]. Possible values are: 1. 'http://ns.electronichealth.net.au/id/null/person/1.0' if there was no approver. 2. 'http://ns.electronichealth.net.au/id/null/person/1.0' if there was no approver. 3. 'http://ns.electronichealth.net.au/id/hi/hpii/1.0/' followed by the approver's HPI-I. 4. 'http://ns.electronichealth.net.au/id/hi/ihi/1.0/' followed by the approver's IHI. 5. The domain namespace of an organisation followed by a local identifier. In this case the organisation is the one that allocated the identifier to the approver.
Approver Person Name	The title and name of the person that reviewed and approved the content of the clinical document. If the name is 'NA', there was no approver.
Organisation Certificate	States whether the certificate is a National Authentication Service for Health (NASH) Public Key Infrastructure (PKI) certificate.

Label	Description
Subject	A set of domain components ('DC'), an organisation name ('O'), and a common name ('CN'). The attributes most relevant to conformance testing are the organisation name, and the national healthcare identifier which is one of the domain components and is also found within the common name. The organisation name and identifier are the name of the healthcare provider organisation or supporting organisation that the certificate was assigned to, and the identifier of that organisation.
Valid From	The start of the validity period for the PKI certificate. The date is displayed in green if the current date (not the signing date) is at or after the start of the validity period, otherwise it is red.
Certificate Policies	States whether the certificate is for a healthcare provider individual, a healthcare provider organisation or a supporting organisation.
Other eSignature files in package	The name(s) of any other eSignature files found within the clinical package. These names are found by opening the clinical package whereas all of the information above this row in this table is obtained from the CDA_SIGN.XML eSignature file.

Figure 96 is another example of the information displayed on the **Sign File Information** tab when a certificate is not a NASH certificate, the certificate policy is unrecognised, and a second eSignature file was found in the clinical package.

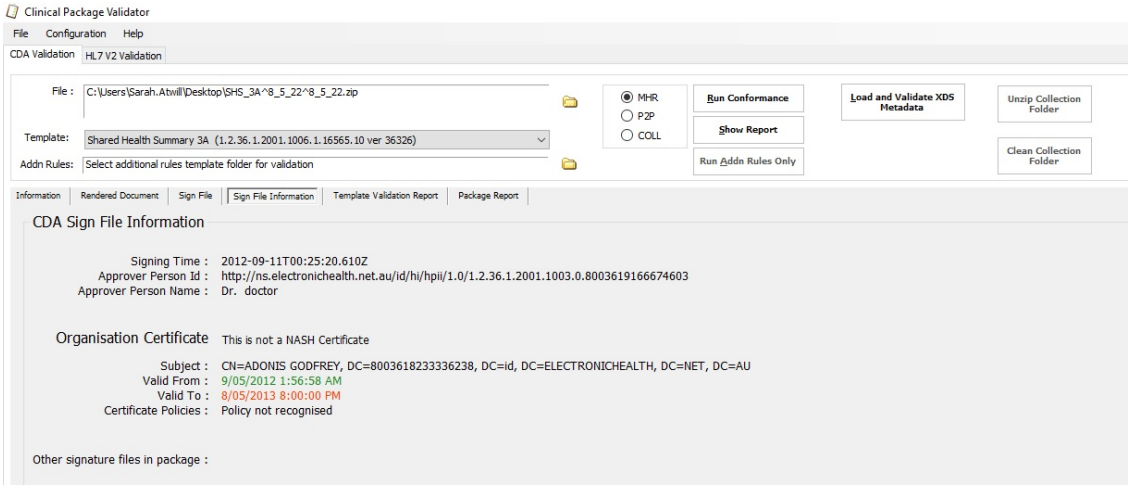


Figure 96 - An unrecognised type of certificate and a second eSignature

4.2.6.5 Template Validation Report tab

The **Template Validation Report** tab displays the results of the My Health Record template validation (for which the Validator may be used) and the clinical document XML file (Figure 97).

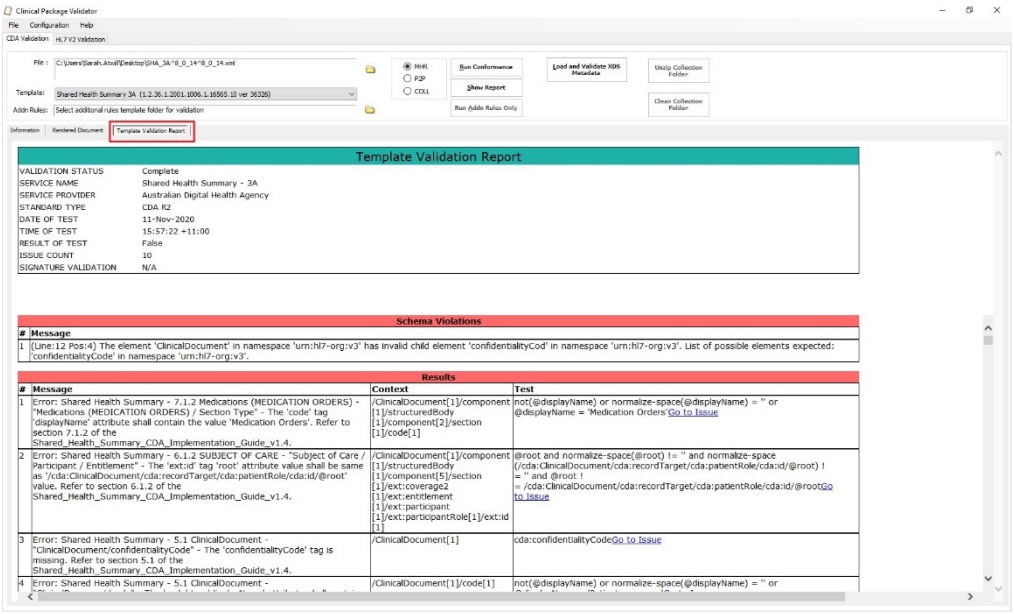


Figure 97 - Template Validation Report tab

Any template validation issue is displayed in this report along with information about the issue, and the context of the issue. A template validation report is generated each time the Validator is run.

Schema issues are displayed before Schematron issues, followed by a display of the clinical document XML file. Figure 98 shows a report with a schema issue highlighted.

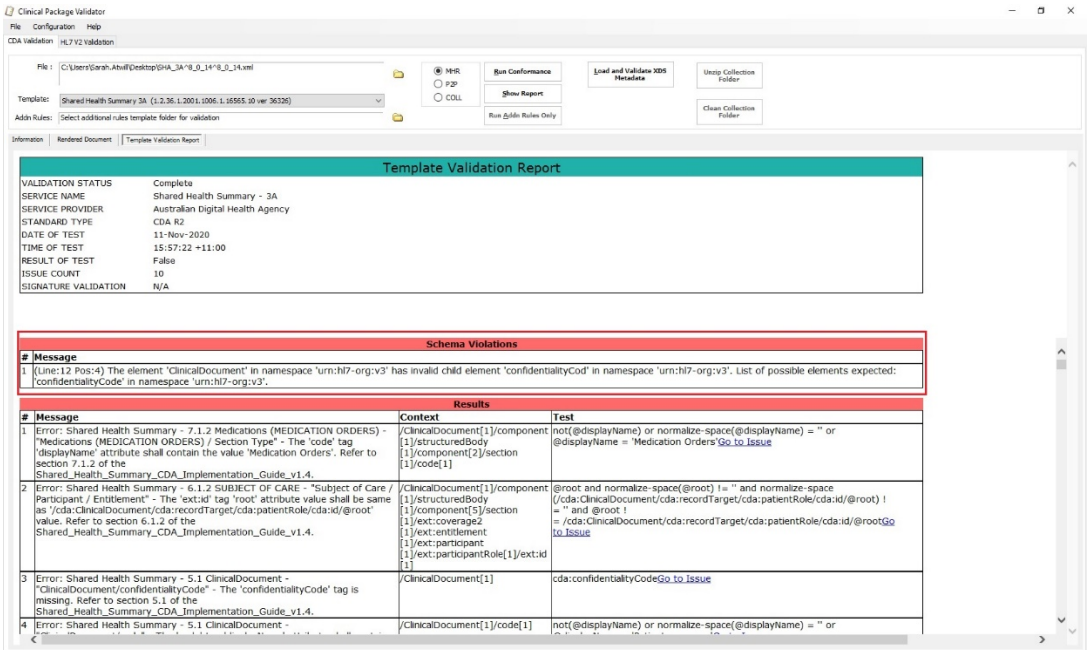


Figure 98 - Template Validation Report tab with schema issue highlighted

Figure 99 shows the report with Schematron issues highlighted.

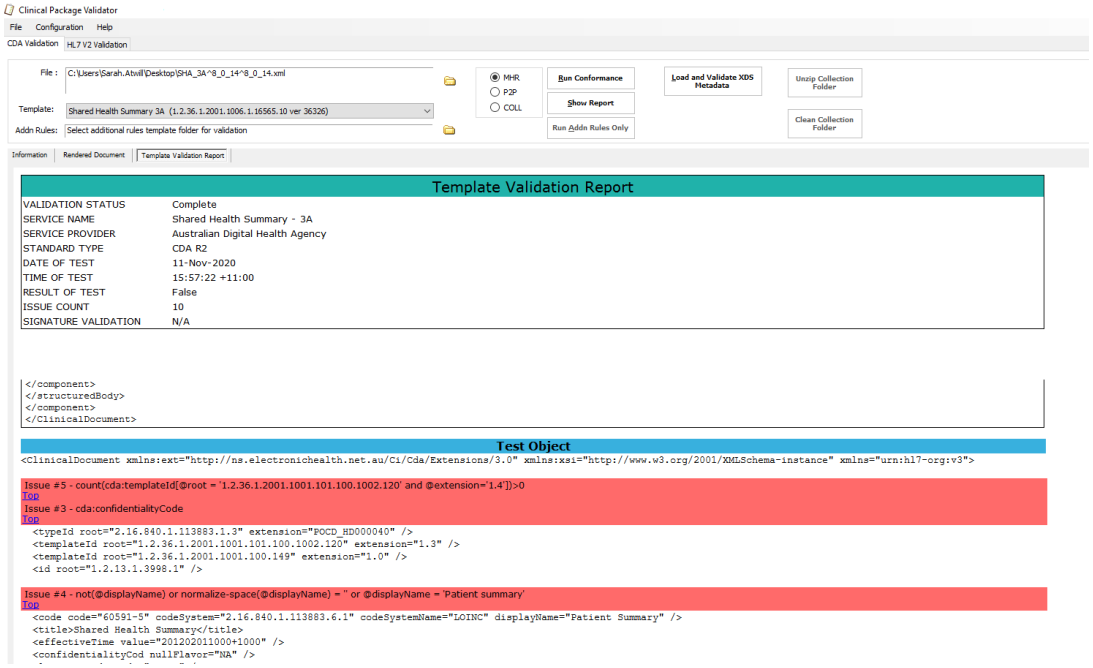


Figure 99 - Template Validation Report tab with Schematron errors highlighted

Figure 100 shows a report where no schema or Schematron issues were found during template validation.

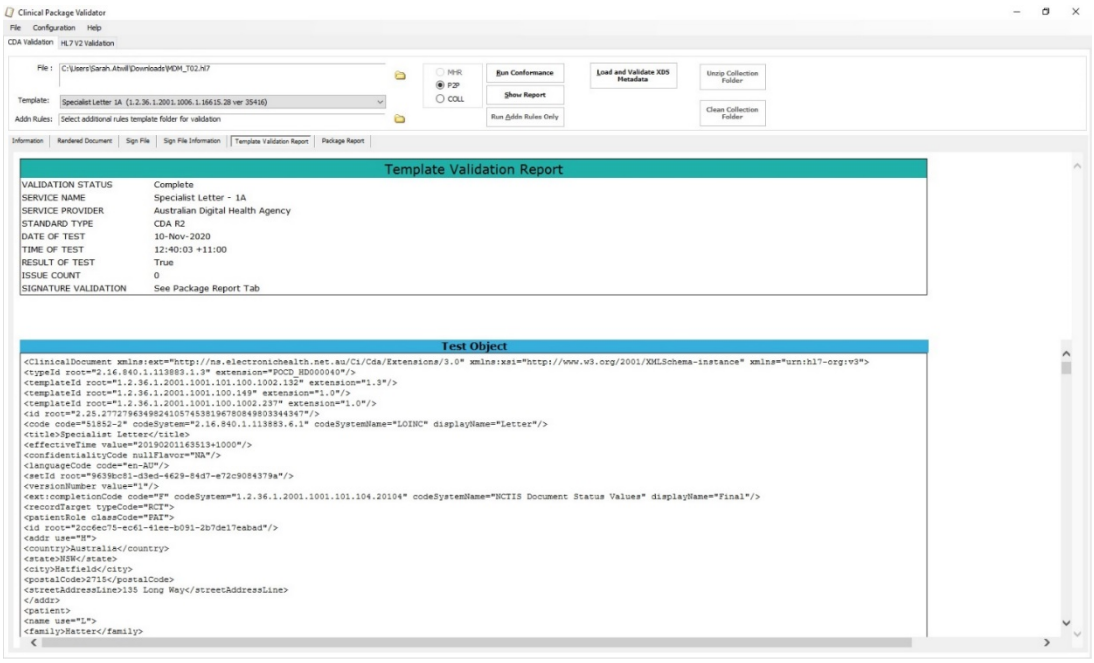


Figure 100 - Template Validation Report tab reporting no errors

The upper portion of the report includes a summary of the schema and Schematron errors and contains the following information.

Table 6 - Report labels

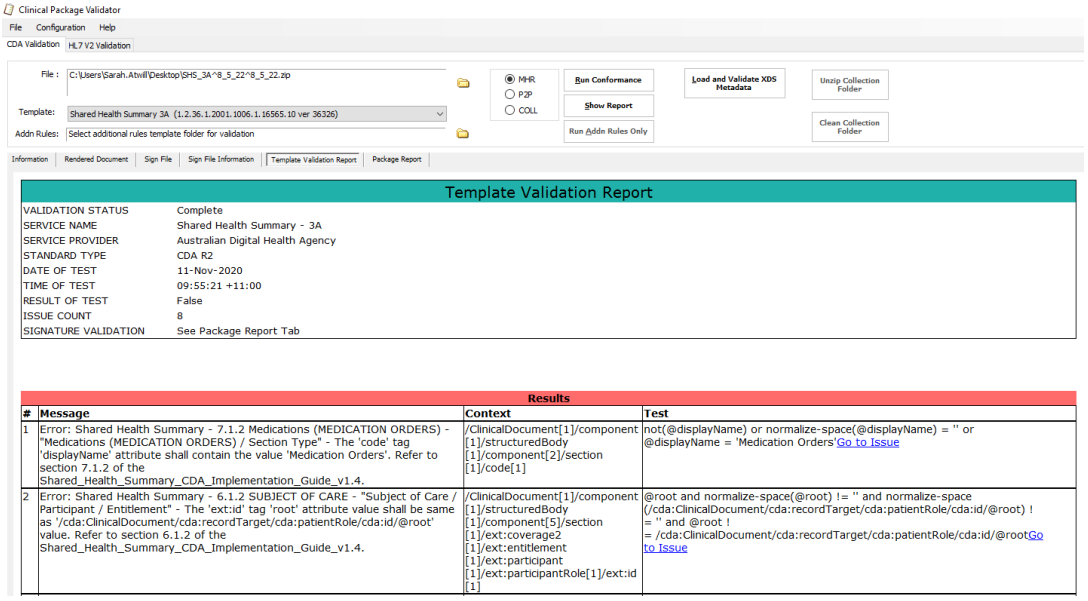
Label	Description
Validation status	**Complete** indicates the Validator has completed the validation process. If the Validator does not complete the validation process, the text **An Error occurred while trying to run the Validator. No Output was produced.** will be displayed.
Service name	The type of the clinical document and the target conformance level.
Service provider	Australian Digital Health Agency
Standard type	HL7 CDA R2 indicates the clinical document specifications are based on release 2 of the *HL7 Clinical Document Architecture* [HL72004].
Date of test	The date in the dd-mmm-yyyy format.
Time of test	The time when the test was run.
Result of test	The overall result of the validation. The result is True if template validation found no issues, and False if issues were found during template validation.
Issue count	The total number of schema and Schematron issues.
Signature validation	Indicates whether any errors were found in the eSignature. “N/A” is displayed if there was no eSignature. “See Package Report Tab” is displayed if there was an eSignature to validate.

If schema issues are found, the middle portion of the report contains the XML line number where the issue is located, the position of the issue within the line, and a description of the issue (Figure 97). If Schematron issues are found (Figure 98), the middle portion of the report contains the information in Table 7.

Table 7 - Schematron issue information

Label	Description
#	The issue number.
Message	The issue message.
Context	The location in the XML where the issue was found (XPath).
Test	The **Go To Issue** link at the end of the test information can be used to go to the location of the issue in the clinical document XML file.

Figure 101 is an example of information that is displayed in the **Template Validation Report** tab when a **Go To Issue** link is selected. The issue message highlighted in red indicates the XML element that was in issue.



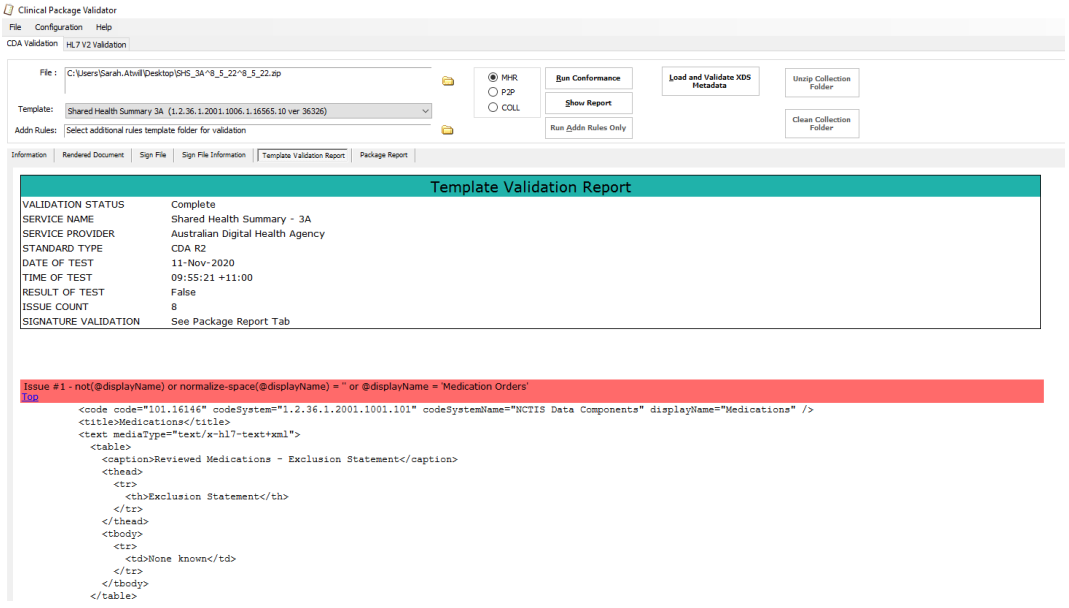


Figure 101 - Template Validation Report tab when a Go to Issue link is selected

Figure 102 shows the display of a clinical document XML file. The clinical document XML file is presented immediately after the display of any Schematron issues.

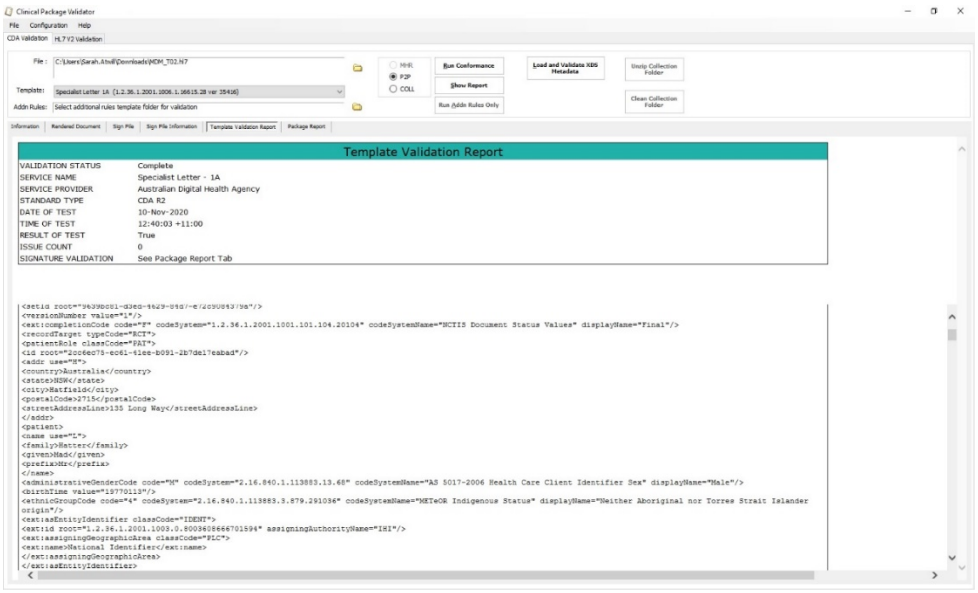


Figure 102 - Template Validation Report tab displaying a clinical document XML file

4.2.6.6 Additional Rules Report tab

The **Additional Rules Report** tab displays the outcome of applying additional rules to the clinical document (Figure 103). This tab is only displayed if an additional template package was imported into the Validator (Section 3.2.3).

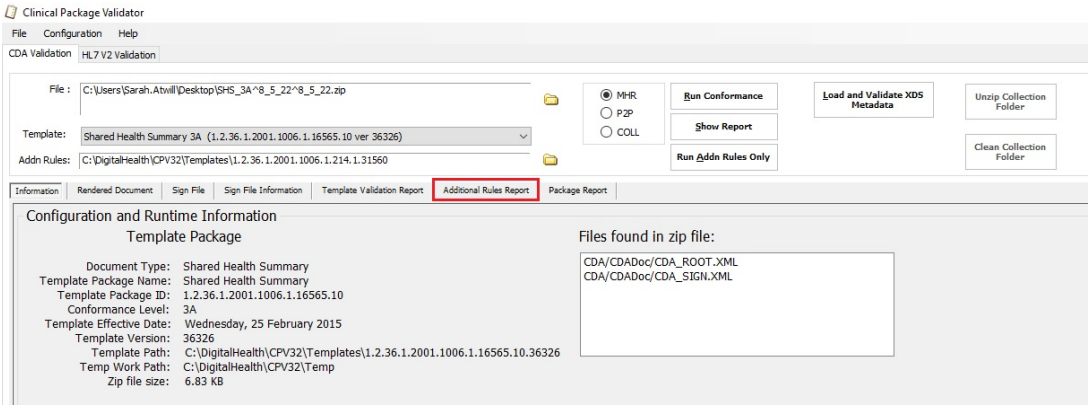


Figure 103 - Additional Rules Report tab

Information is displayed in the **Additional Rules Report** tab in the same way as information is displayed in the **Template Validation Report** tab (Section 4.2.6.5).

4.2.6.7 Package Report tab

The **Package Report** tab displays the results of package validation (for which the Validator may be used) (Figure 104). This tab is only displayed after a clinical package or an HL7 MDM wrapped clinical package has been validated (Section 4.2.1.1).

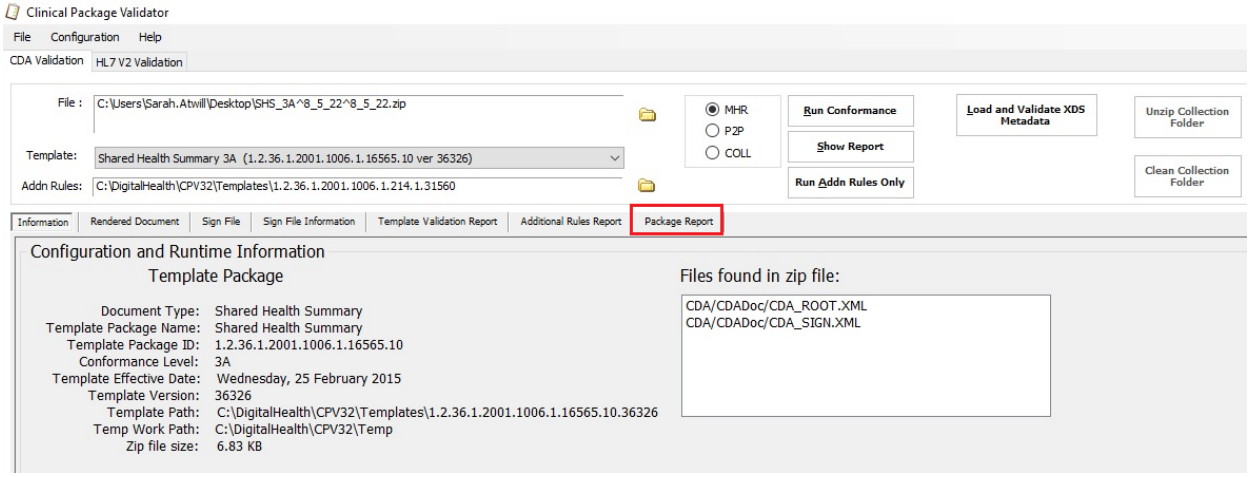


Figure 104 - Package Report tab

The **Package Report** tab displays the validation result for each test case that was applied for the selected context, and a symbol that summarises each validation result (Figure 105). Section 2.1 provides information about these test cases.

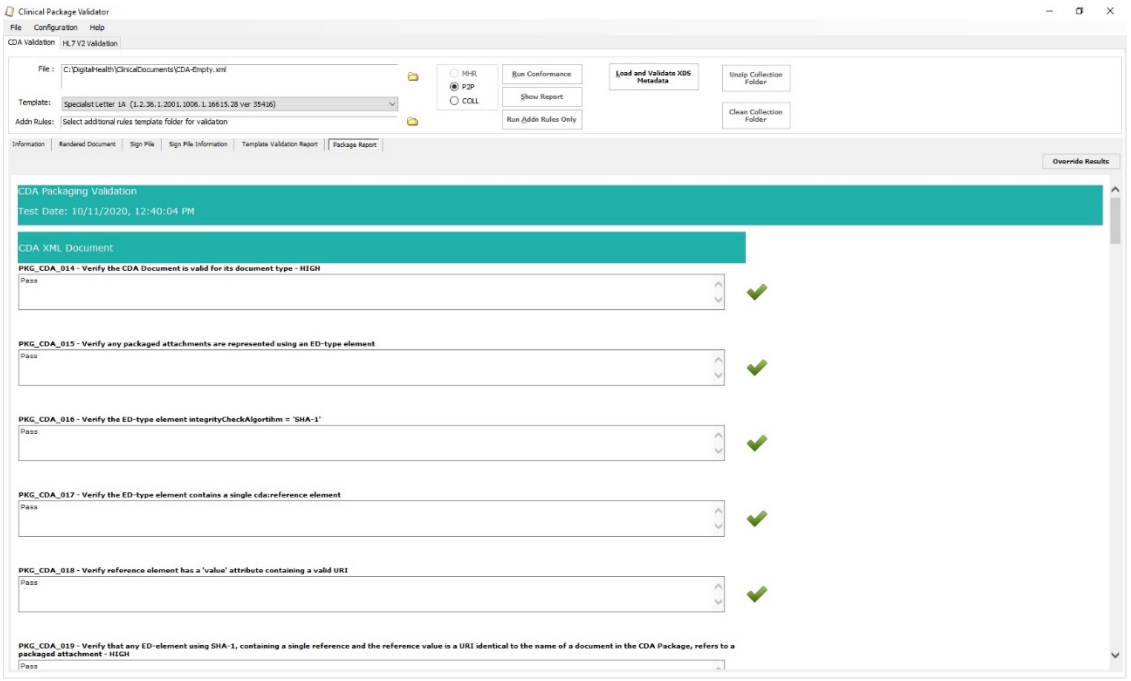


Figure 105 - Package Report tab showing results of clinical package validation

Package validation symbols and their meanings are listed in Table 8.

Table 8 - Package validation symbols

Symbol	Explanation	Usage
	Pass	The Validator determined a definite 'Pass' for the test case.
	Fail	The Validator determined a definite 'Fail' for the test case.
	Warning	The outcome of the test case can only be determined through manual inspection and then a decision of 'Pass' or 'Fail'.
	Not Run	The test case is conditional (e.g. upon an attachment being present) and the validation was not run as the condition was not met.

4.2.7 Override Results command

After the validation has been performed (for a purpose for which the Validator may be used), the **Override Results** command is displayed in the **Package Report** tab (Figure 106).

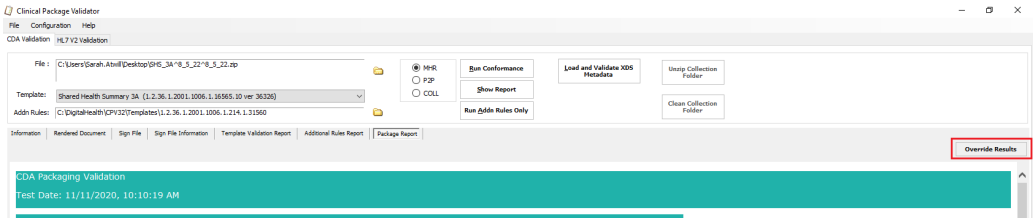


Figure 106 - Override Results command displayed in Package Report tab

The Validator allows the user to override package and code validation results. This is useful for tests that may result in a 'Warning' rather than a 'Pass' or a 'Fail'. A 'Warning' is displayed when the outcome of a test can only be determined by manual inspection.

The Validator allows a user to manually change a test result when a 'Warning' message is displayed. Although the purpose of this command is to allow a user to override a 'Warning' test result, the command also allows the user to override a 'Pass' or 'Fail' result. Overriding a 'Fail' result may result in an overall 'Pass' being reported in the test report although the My Health Record system may reject the clinical document when it is uploaded.

By selecting **Override Results**, a new window that allows the user to select the code validation result to be overridden is displayed (Figure 107).

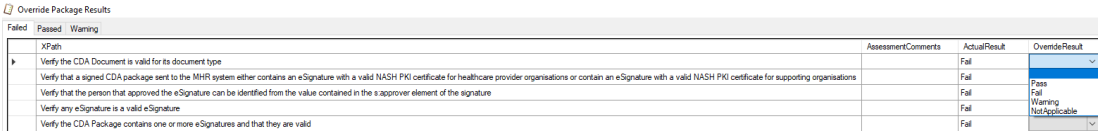


Figure 107 - Overriding a code validation result

The XPath column contains the value of the code, the code system object identifier, the value of the display name, and the location of the data element that contains the code.

The user can manually record a new result and the reason for the change (Figure 108).

Note: To avoid processing errors, double quotation marks (“”) should be used rather than single quotes (‘’) to record the reason for the change. For example, “SNOMED CT-AU” should be used rather than ‘SNOMED CT-AU’.

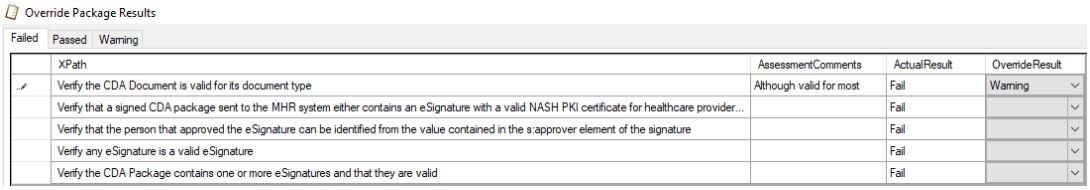


Figure 108 - Recording a new code validation result and the reason for the change

The override changes the overall test result displayed on the Validator’s **Information** tab (Section 4.2.6.1).

4.2.8 Show Report command

The **Show Report** command (Figure 109) allows a user to request the Validator to create a test report (for which the Validator may be used). It also creates an HTML file that is a rendered view of the clinical document that has been validated.

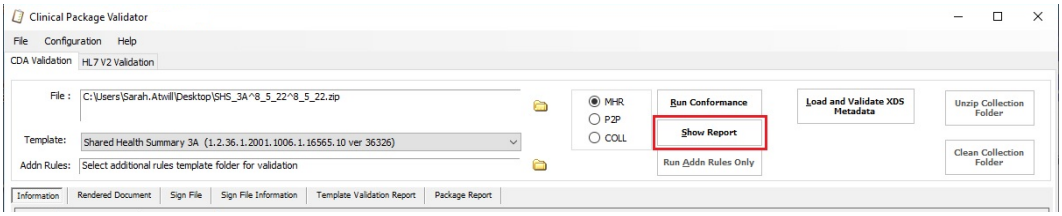


Figure 109 - Requesting the Validator to create a test report

The Validator also provides the user with the option of entering information about the tests performed including the name of the author and tester; the names of the software development and tester organisations; and information about the software being tested and the test environment (Figure 110). This information is remembered and does not need to be re-entered every time the Validator is used.

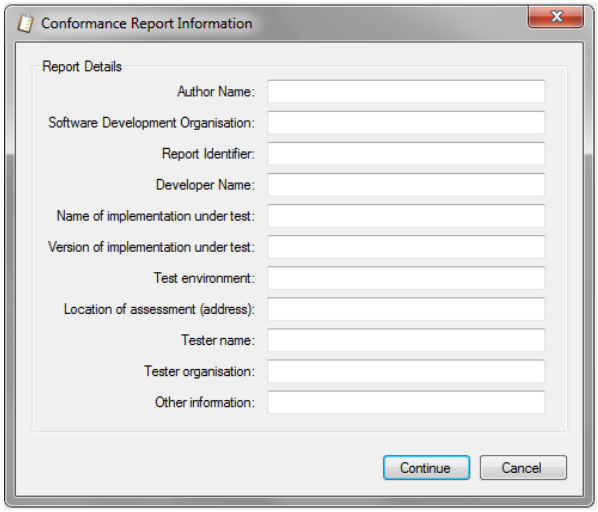


Figure 110 - Recording information about the tests performed

The test report is in Adobe PDF format and contains:

- assessment details, such as the conformance level, context and template ID;
- a list of terminology codes used;
- a summary of the validation test results;
- a detailed error report for each type of validation;
- a reference to the HTML file containing the rendered view of the clinical document that has been validated; and

The PDF file is saved and is located in the same directory from which the target file has been selected.

The summary of the validation test results shows the overall result and the total number of issues and warnings reported by the Validator (Figure 111).

Note: If there are warnings but no issues, the overall test result recorded is a 'Pass', as shown in the 'Overall result' for the 'Other Terminology Validation' in Figure 111.

	Overall result	*Error count*	*Warning count*	*Comments*
Template Validation	✖	2	0	
Package Validation	✖	12	0	
Terminology Validation	✖	1	10	
Other Terminology Validation	✔	0	4	
Additional Rules Validation	➖	0	0	

Figure 111 - Test report showing a summary of the validation results

If a validation result is overridden before the test report is generated, the report records the original result, the new result, and the reason why the original result was overridden (Figure 112). The Validator uses the new result when it produces the summary of validation results in the test report.

#	CodeSet/Code/Name	Comment	Result	Override
1	Code Value = 73817000 CodeSystem = 2.16.840.1.113883.6.96 Display Name= Enteritis due to radiation Calculated XPath:/cda:ClinicalDocument[1]/cda:component[1]/cda:structuredBody[1]/cda:component[1]/cda:section[1]/cda:component[1]/cda:section[1]/cda:entry[1]/cda:observation[1]/cda:value[1]	The actuial codeSystemName value "SNOMED" is close enough to the expected value "SNOMED CT-AU"	Warning	Pass

Figure 112 - Record of an overridden test result in a test report

4.2.9 Cumulative report of test results

The Validator creates a cumulative report of test results in the file report.tsv which is located in the same directory as where the target file has been selected from. A new row is added to this file each time a clinical document or clinical package (for which the Validator may be used) is validated. The information recorded in this file is listed in Table 9.

Table 9 - Information in the report.tsv file

Label	Description
Input File Name	The filename of the clinical document or clinical package that was validated.
Test Date	The date and time the file was assessed by the Validator.
Template Name	The name of the My Health Record template package as recorded in the template package metadata.
Template ID	The identifier of the My Health Record template package as recorded in the template package metadata.
Template Version	The version number of the My Health Record template package as recorded in the template package metadata.
Template Effective Date	The date the My Health Record template package was approved as recorded in the template package metadata.
Conformance Level	The conformance level as recorded in the template package metadata.
Test Context	The value is either 'MHR' or 'P2P'.
Package Errors	The number of clinical package errors.
Package Warnings	The number of clinical package warnings.
Template Errors	The number of errors reported by applying the My Health Record template package.
Template Warnings	The number of warnings reported by applying the My Health Record template package. **Note:** template packages in the My Health Record system do not report warnings.
Additional Rules Issues	The number of issues reported by applying the additional template package.
Additional Rules Warnings	The number of warnings reported by applying the additional template package.
Summary	An overall Pass or Fail.

The Validator also creates an analysis report of test results in the file analysis.tsv which is located in the same directory as where the target file has been selected from. A new row is added to this file for each result reported for a clinical document being validated so there may be 0 or more rows returned depending on the issues found. The information recorded in this file is listed in Table 10.

Table 10 – Encoded information

Label	Description
Document Creation Date	The date the document was created (effectiveTime).

Label	Description
Document Creation Time	The time the document was created (effectiveTime).
Error ID	<empty>
Priority	The priority level of issue raised.
Error Type	Whether the issue is from a template package or conformance set of rules (Conformance or Template).
Error Classification	Classification type of issue raised (ERROR, HINT, RECOMMENDATION, FINDING, WARNING).
Document Type	CDA document type.
Conformance Level	The level of conformance for which the document is being tested.
Template Package ID	Template package ID being used to test the document.
Document Id	The id of the document (id root).
Site Impacted in Sample	The Organisation that authored the document.
Error	The error reported.
Conformance Specification	<empty>
Guidance	<empty>
XML Evidence	<empty>
Proposed Vendor Action	<empty>
Timeframe for Completion	<empty>
Expected Version Containing Fix	<empty>

Many of the fields may be empty as this analysis report is used by the conformance team to record and analyse issues.

4.2.10 Load and Validate XDS Metadata

The **Load and Validate XDS Metadata** command allows a user to request the Validator to test a SOAP upload document xml file and run a series of tests that the My Health Record system also performs when uploading a document (Figure 113).

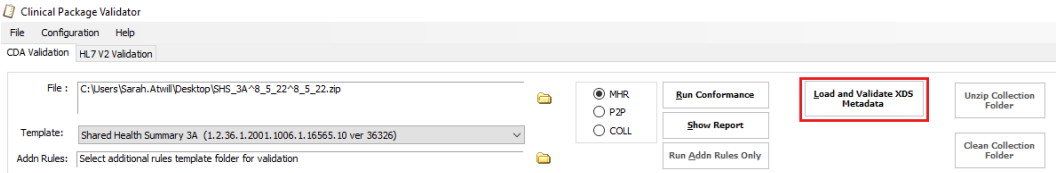


Figure 113 - Requesting the Validator to load and validate a soap upload document request

This function would generally be used when a user is experiencing problems uploading a document to the My Health Record system and getting the generic error message "PCEHR_ERROR_3002 - Document metadata failed validation".

In order to use this command, the user must have captured the SOAP request message that is sent to the My Health Record system and saved as an xml file.

The SOAP message can come in two formats:

- where the document is contained within the xml file as a base64 string

```
            </RegistryObjectList>
        </SubmitObjectsRequest>
        <Document id="DOCUMENT_SYMBOLICID_01">UEsDBBQAAAAAAGdq5koAAAAAAAAAAAAAA
    </ProvideAndRegisterDocumentSetRequest>
```

- or as a reference to the document when the sent as in MTOM format. For this format, the Validator will also prompt the user to select the CDA Package (zip) that was sent in order to complete the tests it runs.

```
        </SubmitObjectsRequest>
        <Document id="DOCUMENT_SYMBOLICID_01">
            <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:10203d2b-5132-4878-a23e-c162340b0988" />
        </Document>
    </ProvideAndRegisterDocumentSetRequest>
```

The screenshot below (Figure 114) shows the result of having selected a SOAP message to validate. The example shows the IHI's have not matched.

The screen details show:

- the certificate that signed the SOAP message;
- the XDS Metadata that was included with the CDA document;
- the custom SOAP Header data; and
- a Report summary of the tests carried out.

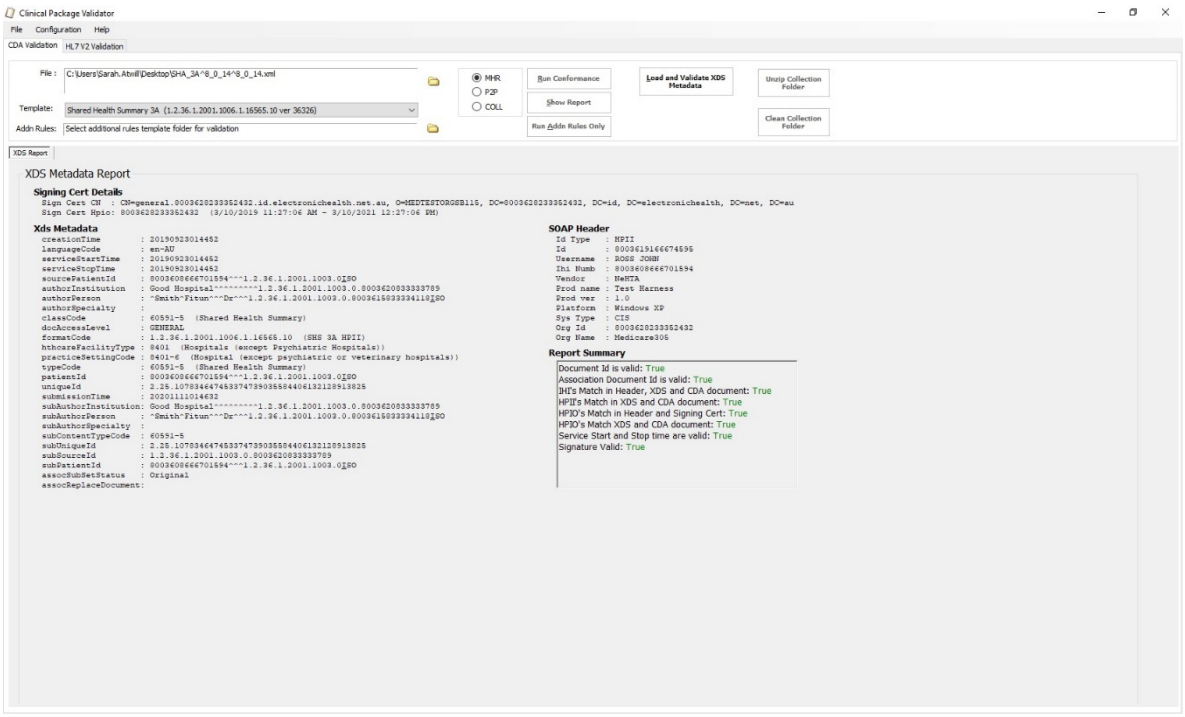


Figure 114 - Requesting the Validator to load and validate a soap upload document request

The tests that are performed are:

1 check that the file provided is a validate XML document;

2 the Soap message was signed;

3 the XDS Metadata was included;

4 the CDA Package was valid (zip file format); and

5 report the following test results:

   a the IHI's Match in SOAP Header, XDS metadata and CDA document

   b the HPI-I's Match in the XDS metadata and CDA document

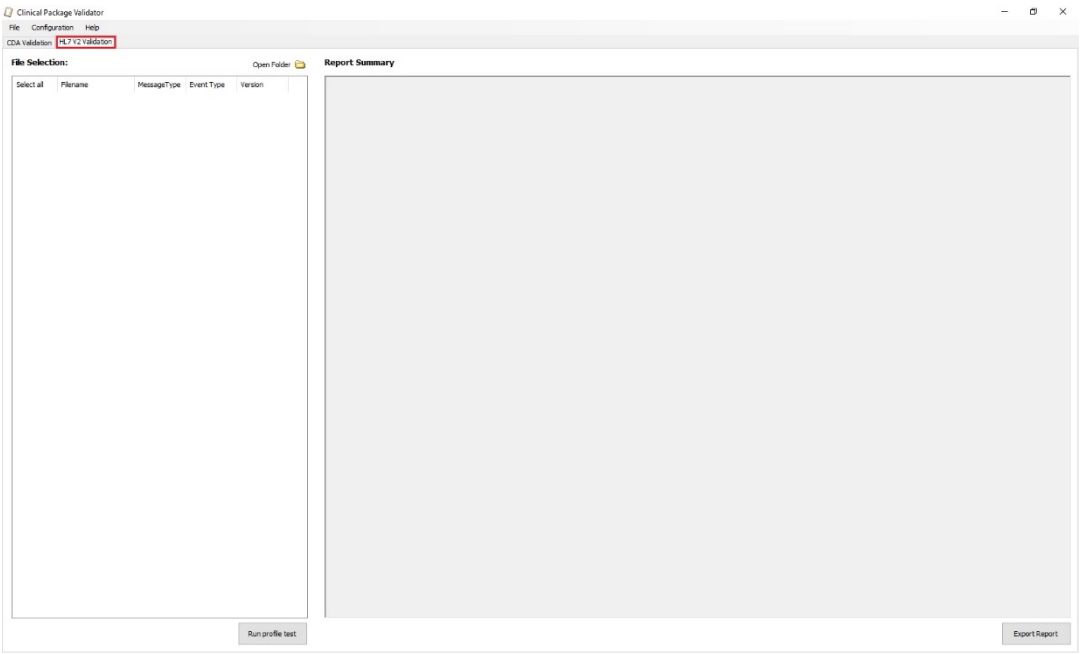
   c the HPI-O's Match in Header and Signing Cert

   d the HPI-O's Match XDS metadata and CDA document

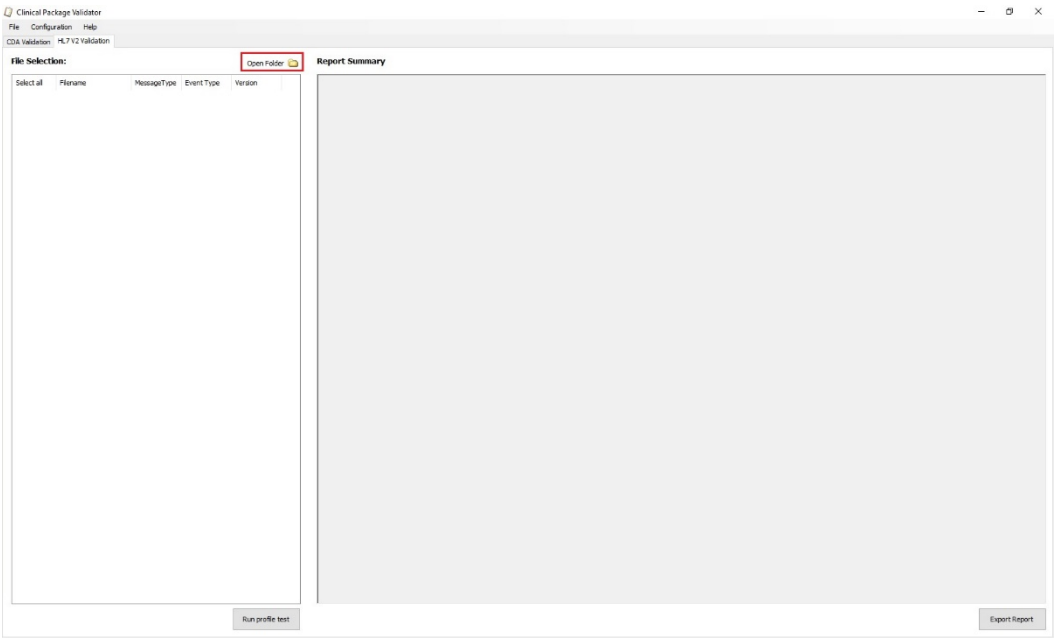
   e the Service Start and Stop time are valid.

4.3 HL7 v2 validation

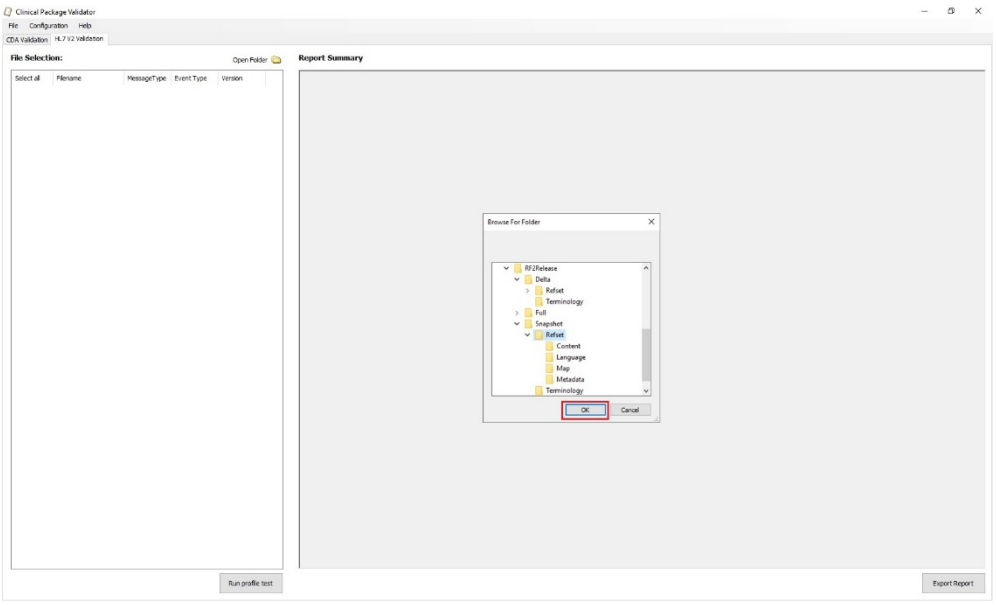
1 To validate a number of HL7 v2 messages go to the HL7 v2 validation tab



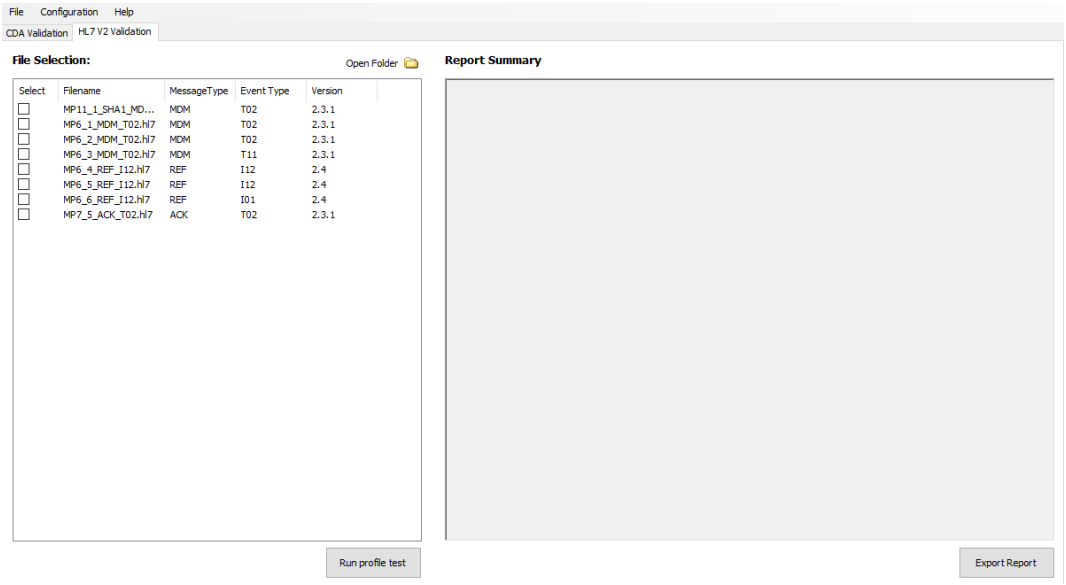
2 Select to open the folder where the HL7 messages are stored



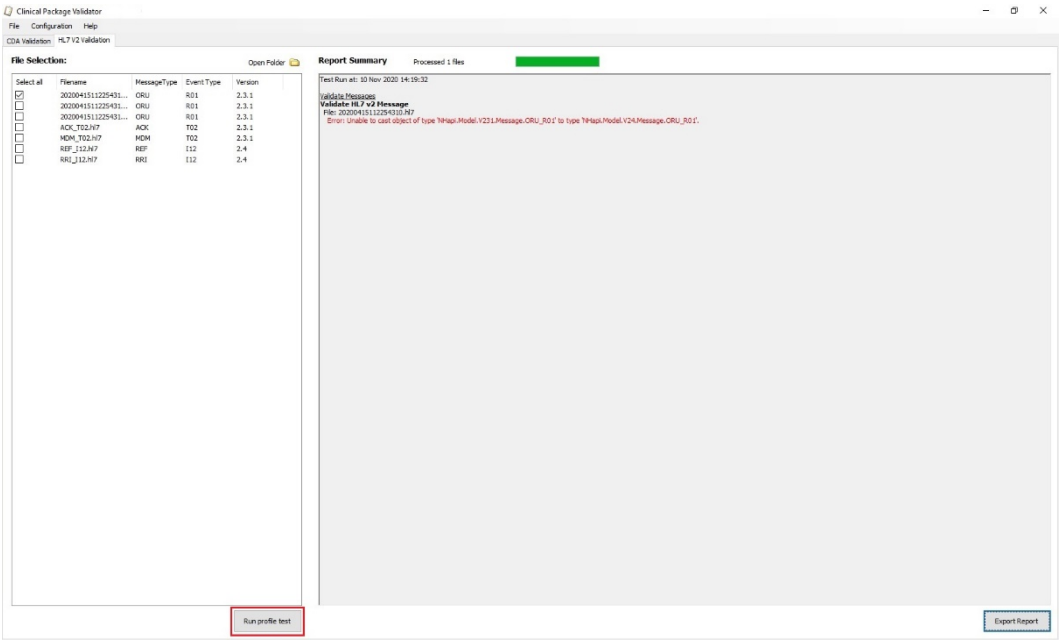
3 Choose the appropriate folder



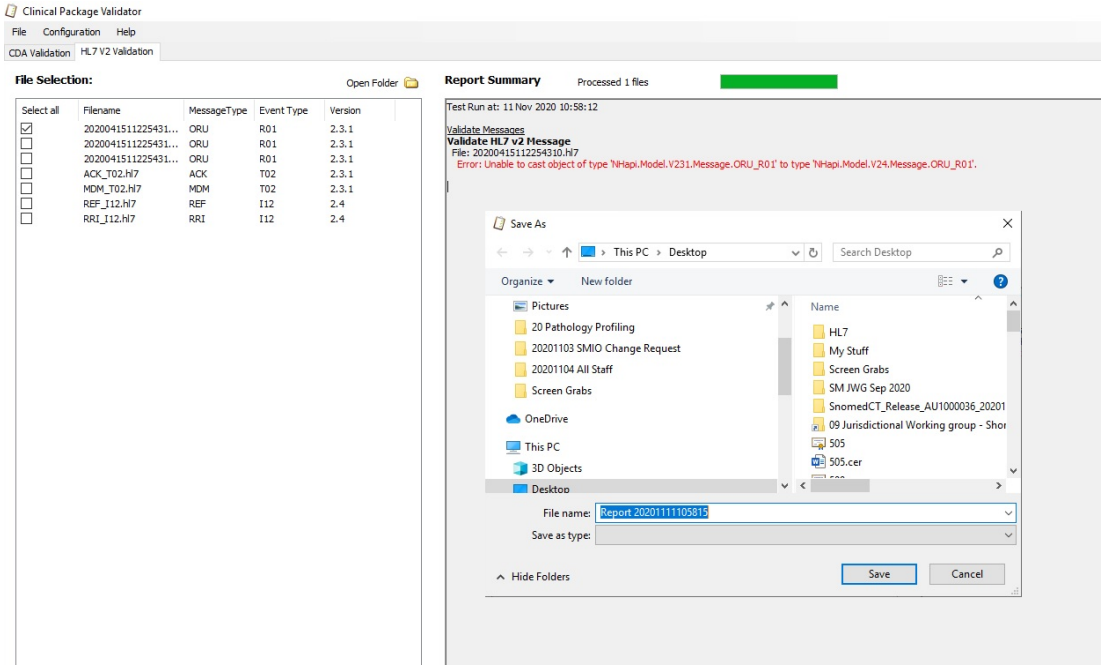
4 The HL7 v2 files available within the folder will appear



5 Select the HL7 v2 messages to test and select the “Run Profile Test” button



6 To save the report choose Export Report and select a location to save the file



5 Examples of validation

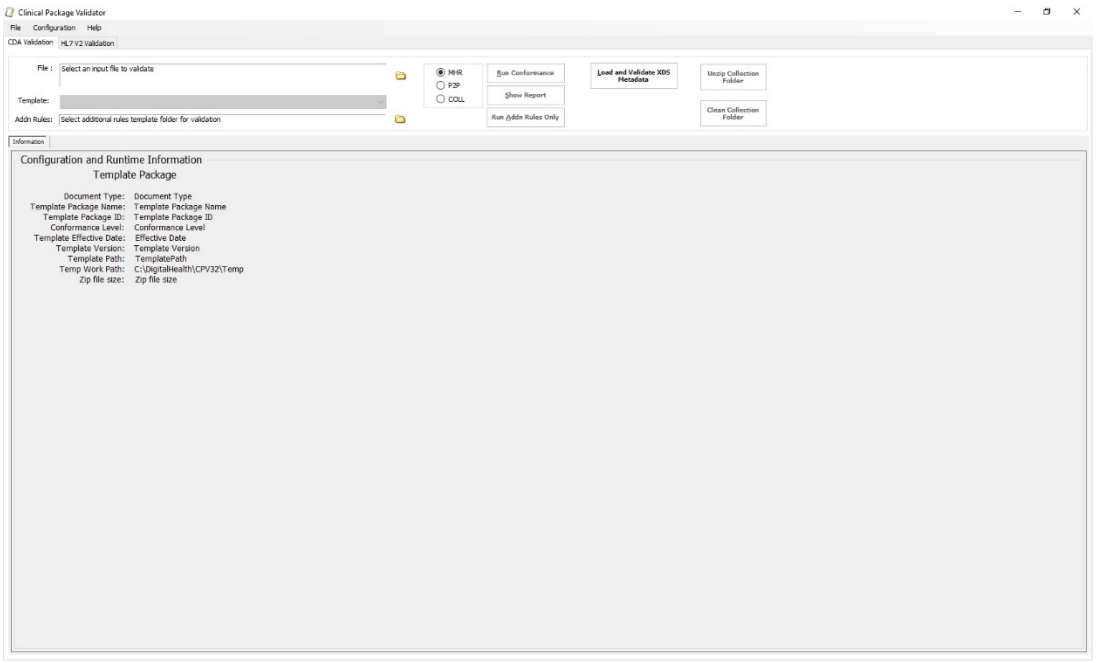
This section provides examples to demonstrate how the Validator can be used to validate clinical documents and clinical packages (for which the Validator may be used). The Validator is a tool to assist users only and while it assists in testing conformance, it does not, and should not be relied upon to test all conformance specifications. Please refer further to the Clinical Package Validator Product Data Sheet and confirm your intended use with the Agency Help Desk on 1300 901 001.

5.1 CDA validation

5.1.1 Validate a 3A Shared Health Summary, My Health Record context

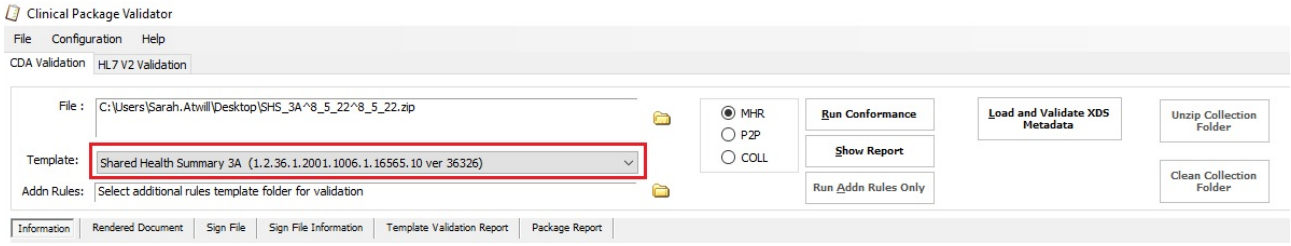
The following example shows how the Validator may be used to assess a Shared Health Summary document within a clinical package for level 3A conformance in the My Health Record context.

1 If the Validator has been installed and configured but is not already launched, launch the Validator as follows. If using Windows 8 or above, click the Windows icon to open the Metro view and either search for “Clinical Package Validator” or visually locate the application menu. The default screen will appear.

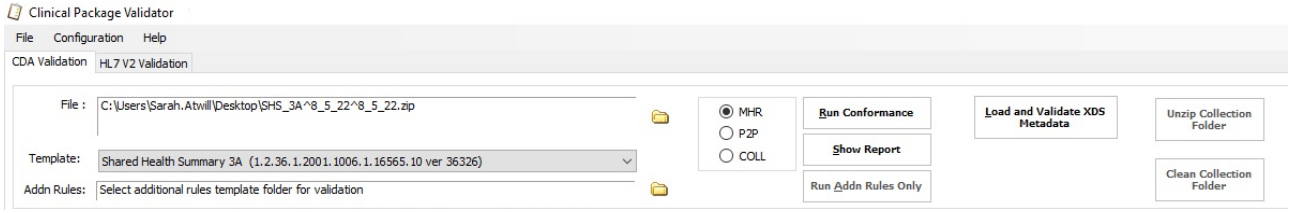


2 Click the Open File button ( ) next to the File parameter and locate the clinical package to be validated, or drag the clinical package onto the **Information** tab or the **File** location field (Section 4.2.1.1).

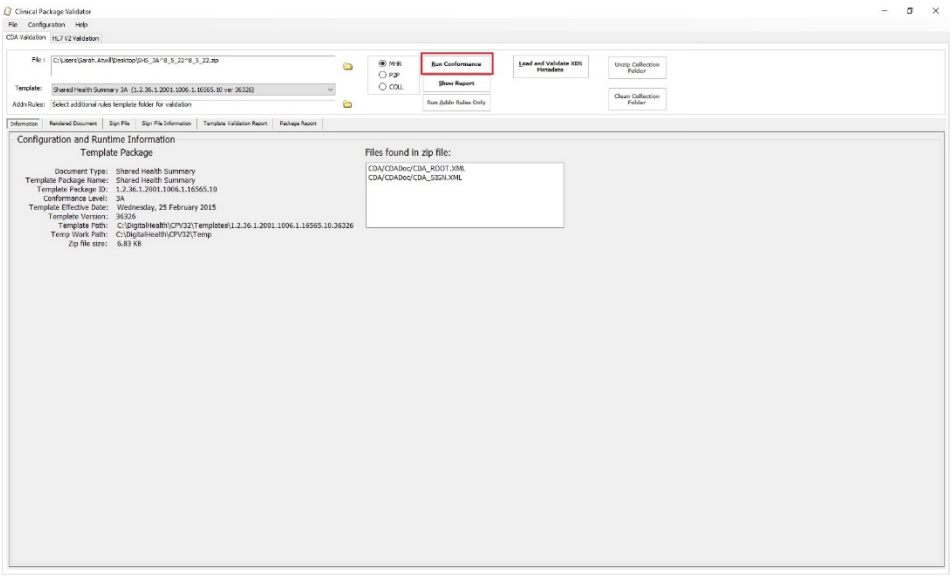
3 If more than one template is imported in the Validator, select the relevant template for a level 3A Shared Health Summary. The default context for validating a clinical package is My Health Record (Section 01).



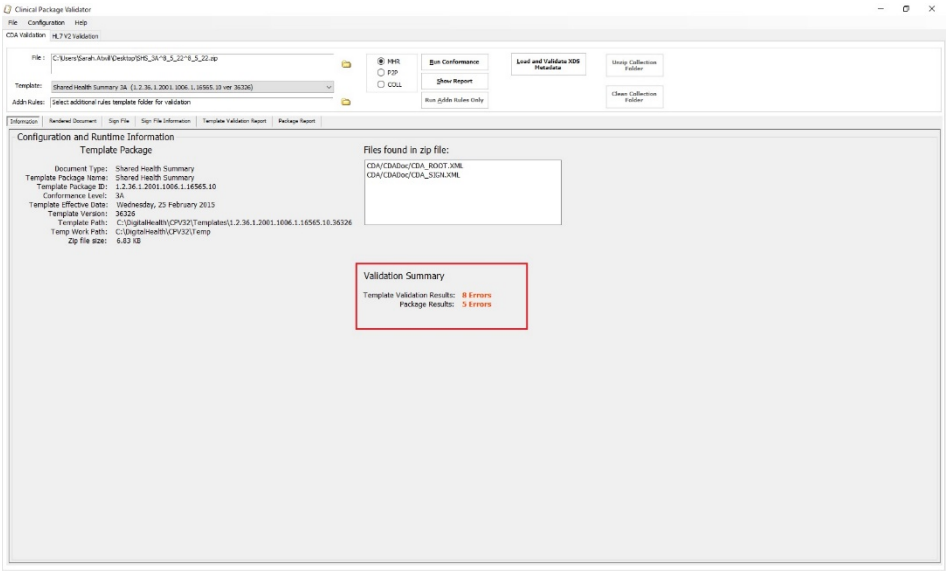
After the parameters for testing a Shared Health Summary for level 3A conformance in the My Health Record context have been selected, the **Information** tab provides detail about the clinical document and template package.



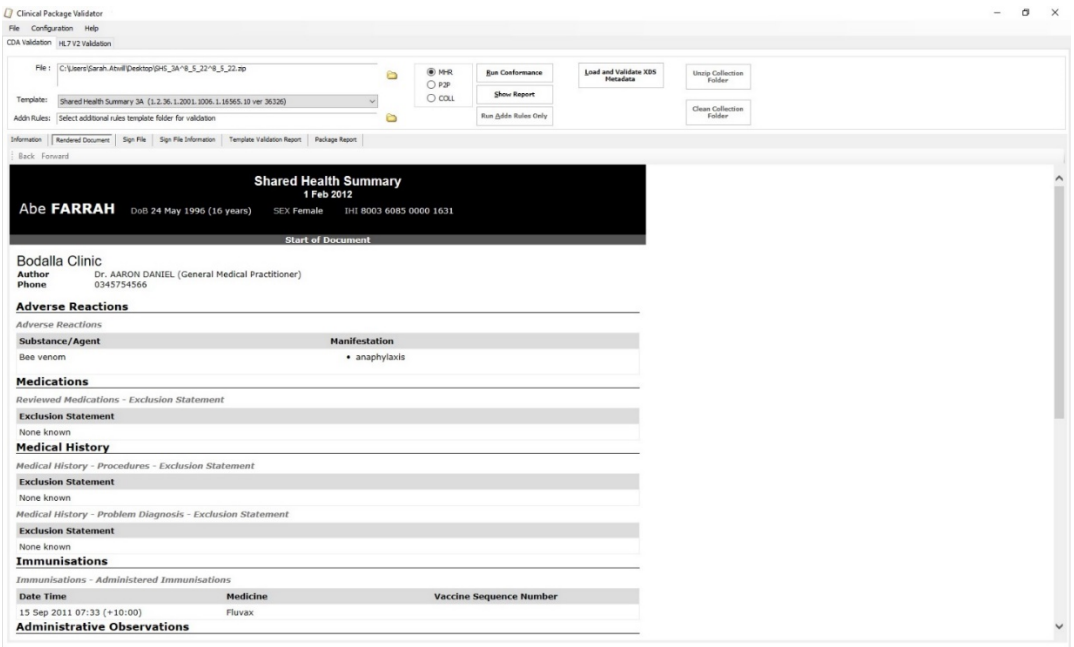
4 Click the **Run Conformance** button to perform the validation.



5 The **Information** tab shows a summary of the validation results. In this example there are 8 template errors and 5 CDA Package errors



6 The **Rendered Document** tab displays a rendered view of the validated Shared Health Summary.



5.1.1.1 View information about the eSignature file

The **Sign File Information** tab displays information about the primary eSignature file of the clinical package (Figure 115).

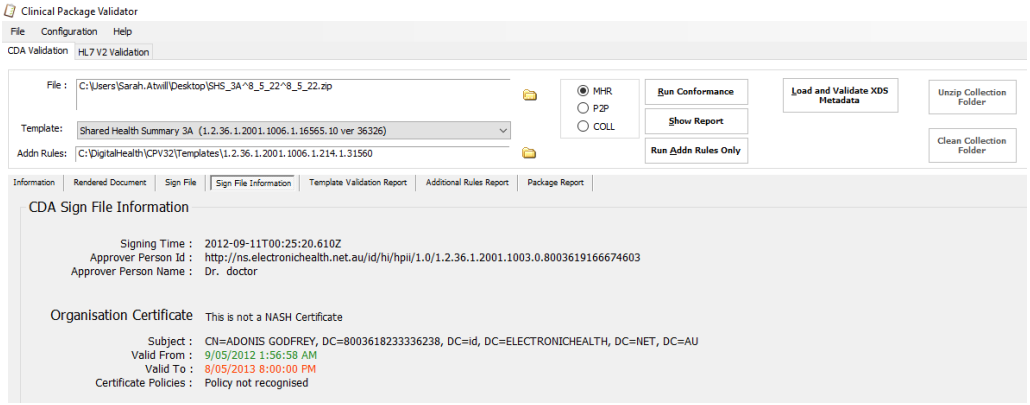


Figure 115 - Sign File Information tab showing a summary of information about the primary eSignature file

This includes information about an approver's name and HPI-I, whether the PKI certificate had expired at the time of running the validation, and whether the PKI certificate is a NASH certificate.

The approver's name and HPI-I reported by the Validator in this tab can be used to check whether the healthcare software system is recording the same person as the approver of the clinical document.

In the above example, the Validator reported that the PKI certificate had expired when the validation was performed. This is not a problem if the certificate had not expired when the clinical document was signed.

In the above example, the Validator reported that the organisation certificate was not a NASH certificate. It is a requirement that all certificates used for signing clinical packages be NASH PKI certificates for healthcare provider organisations or NASH PKI certificates for supporting organisations. This error is also reported against the relevant test case in the **Package Report** tab.

The information displayed in the **Sign File Information** tab is usually the only information about the eSignature that is needed for validating a clinical document.

Additional information about the eSignature may be obtained by examining the CDA_SIGN.XML file displayed in the **Sign File** tab. In the following example, the PKI certificate is displayed to the right of the <X509Certificate> XML tag (Figure 116).

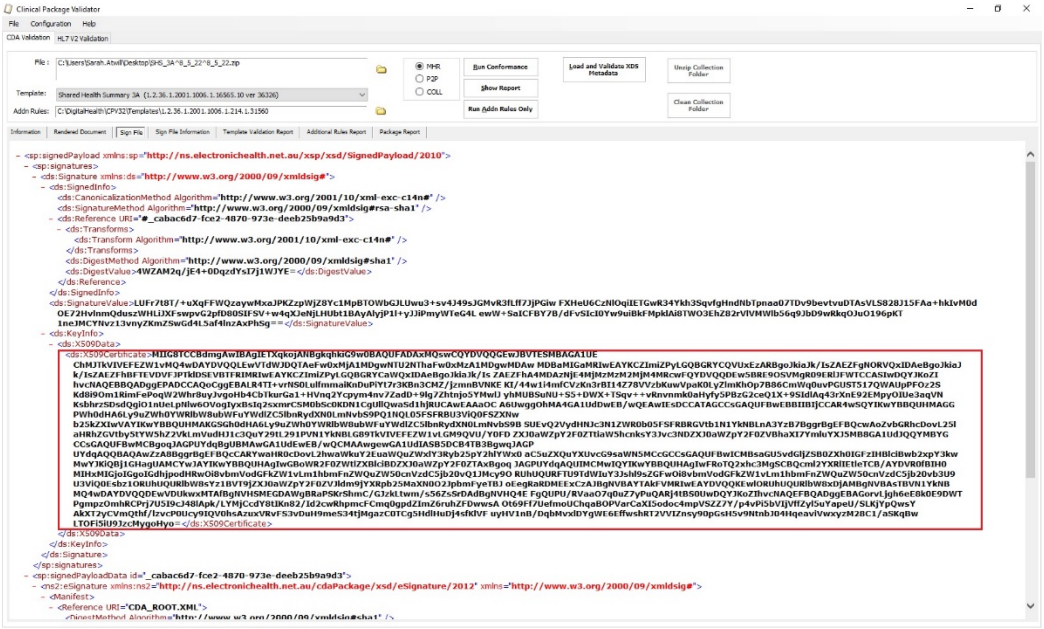


Figure 116 - Sign File tab displaying the PKI certificate in the CDA_SIGN.XML file

To examine the PKI certificate, use a text editor to copy and save it to a document with the '.cer' filename extension. Double-click to open the saved document and view the certificate (Figure 117 and Figure 118).

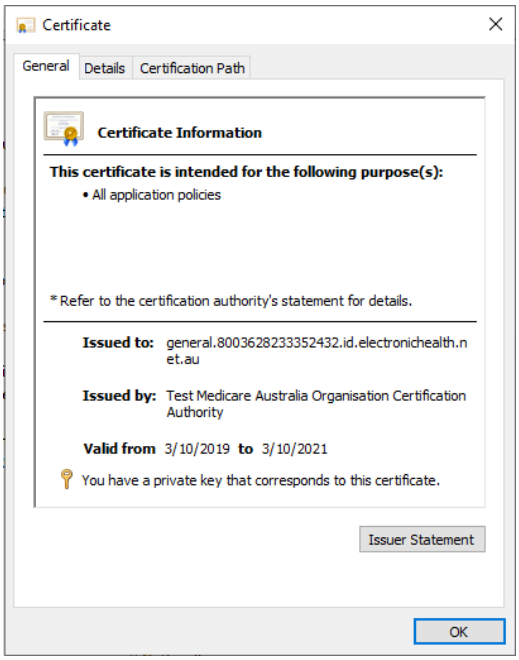


Figure 117 - A view of the PKI certificate

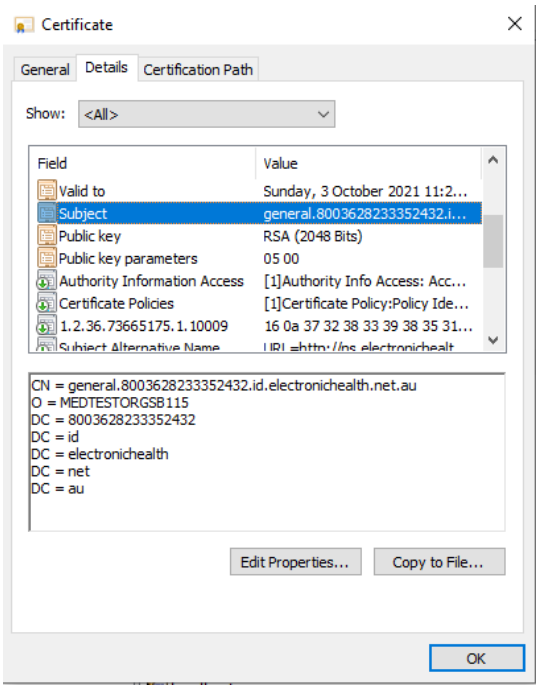


Figure 118 - PKI certificate details

5.1.1.2 Display template validation results

To view the detailed template validation results, click the **Template Validation Report** tab (Figure 119) In this example, the error count was zero as no errors were found during template validation. The clinical document still needs to be manually inspected for conformance to requirements because not all conformance tests can be specified in a template (Section 0.2).

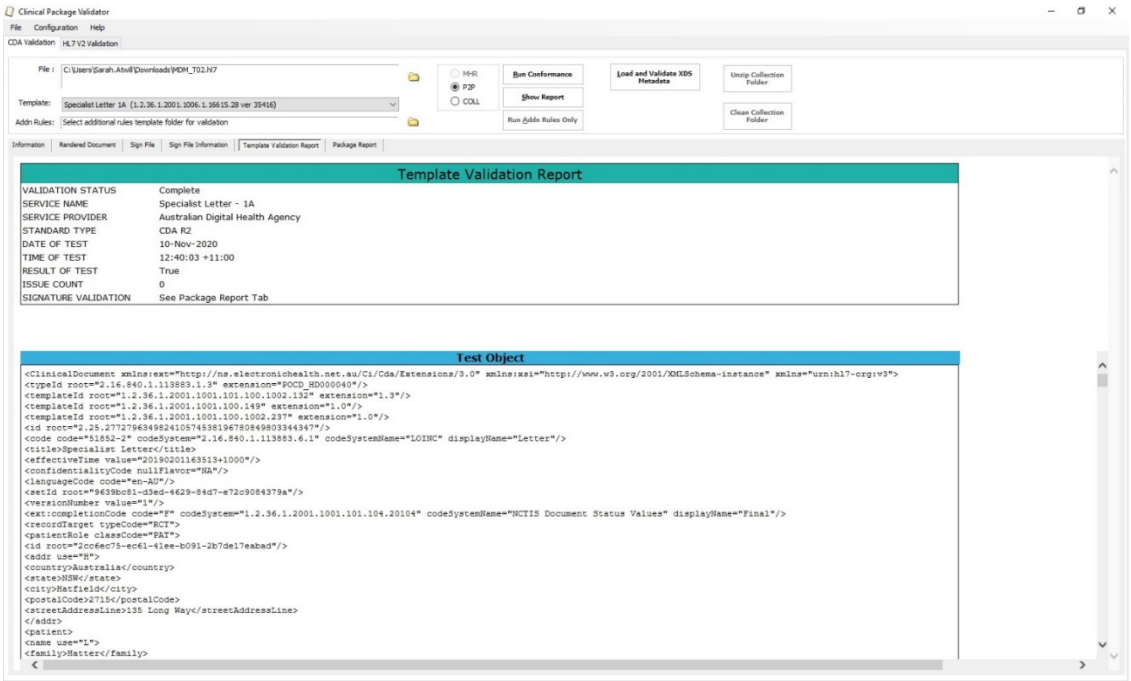


Figure 119 - Template Validation Report tab showing template validation results

5.1.1.3 Display package validation results

To view the detailed package validation results, click the **Package Report** tab (Figure 120). The Validator reports some errors with the clinical package in this example, including the absence of a valid NASH certificate.

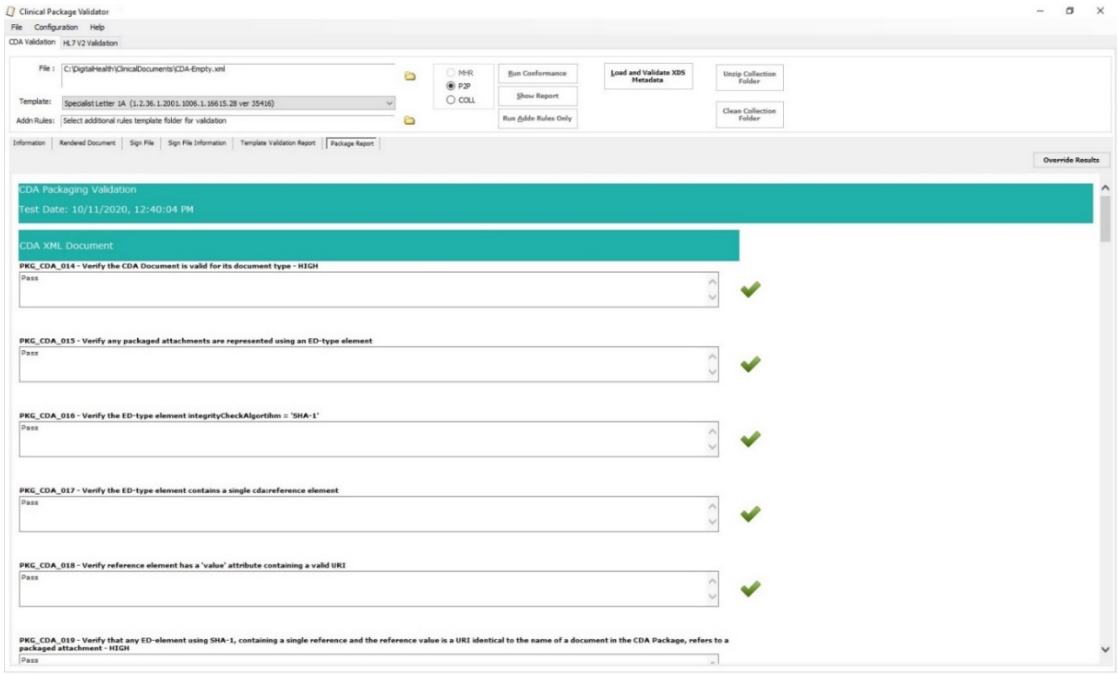


Figure 120 - Package Report tab showing package validation results

5.1.1.4 Generate a test report

To generate a test report in Adobe PDF format, click the **Show Report** button (Figure 121). Enter details of information to be included in the report and then click the **Continue** button (Figure 122).

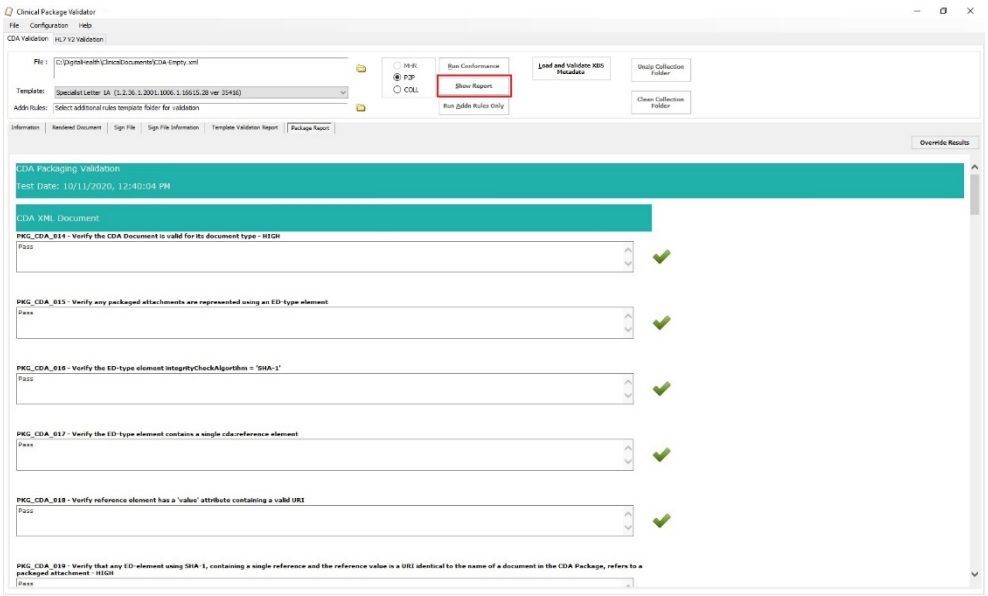


Figure 121 - Generating a test report

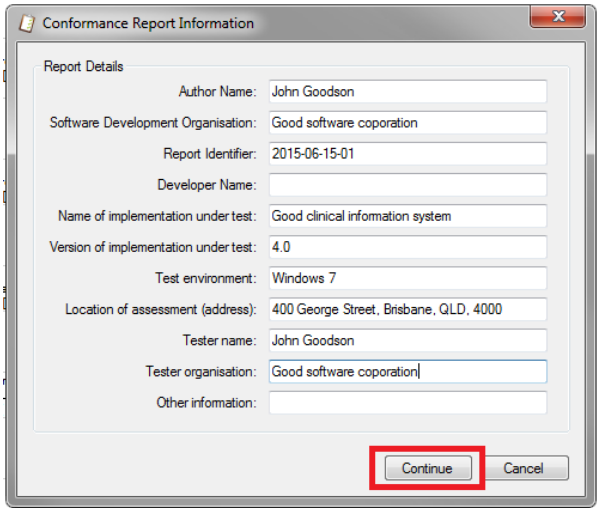


Figure 122 - Adding test report information

The test report is now complete (Figure 123).

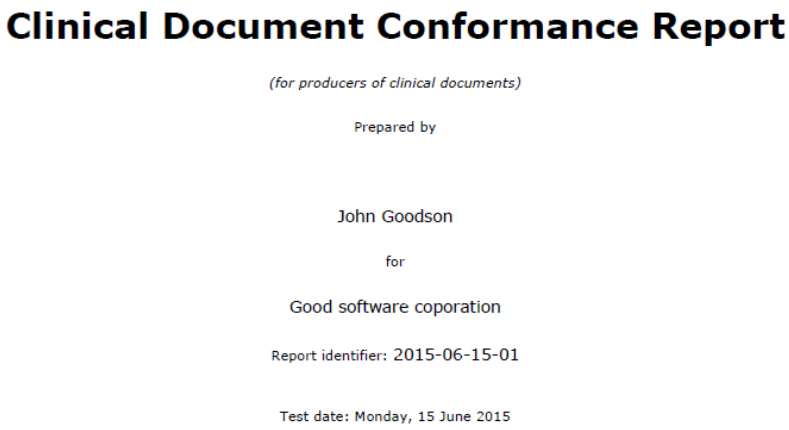


Figure 123 - Completed test report

5.2 HL7 v2 validation

5.2.1 Validate a REF^I12 message

1 To validate a REF^I12 message, select the Open folder icon and navigate to the file/s you want to validate (Figure 124) and then click the Ok button

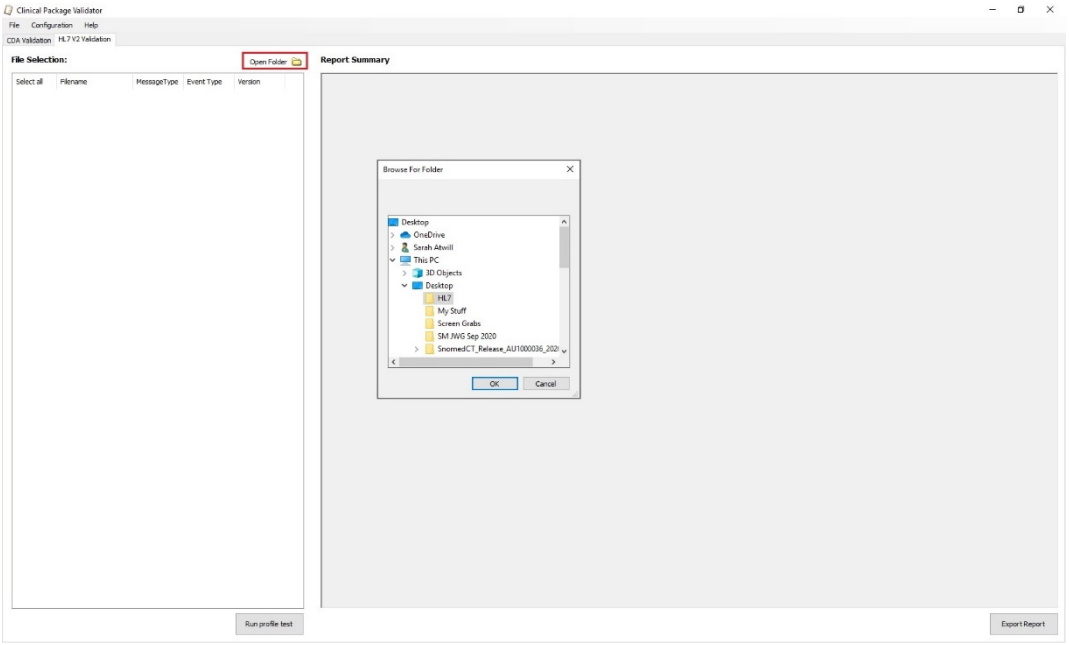


Figure 124 – Open files

2 Select the files you wish to run a profile report on (Figure 125) and click the Run profile test button

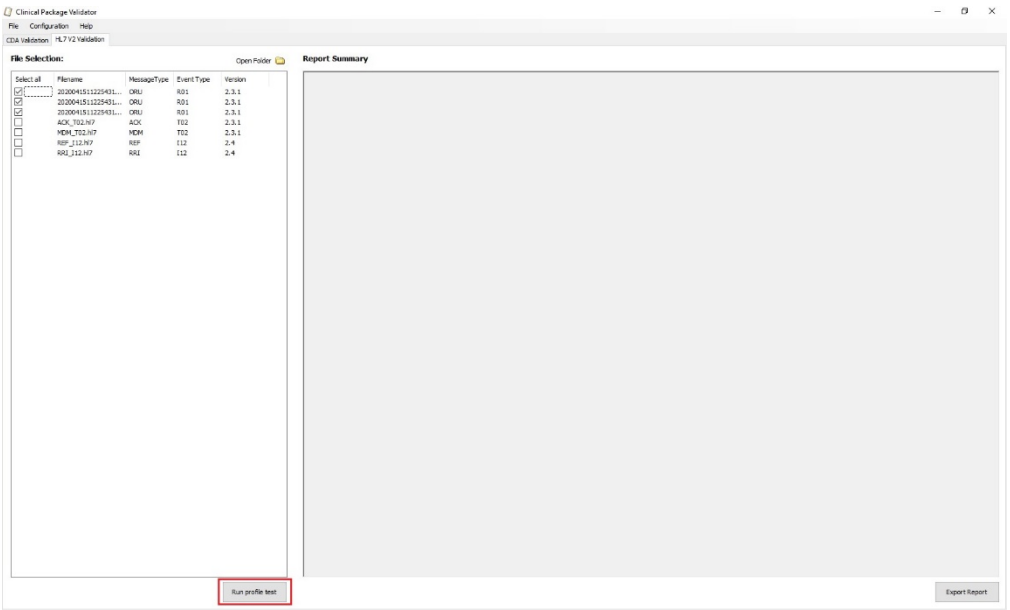


Figure 125 – Run profile test

3 The Report Summary will load (Figure 126)

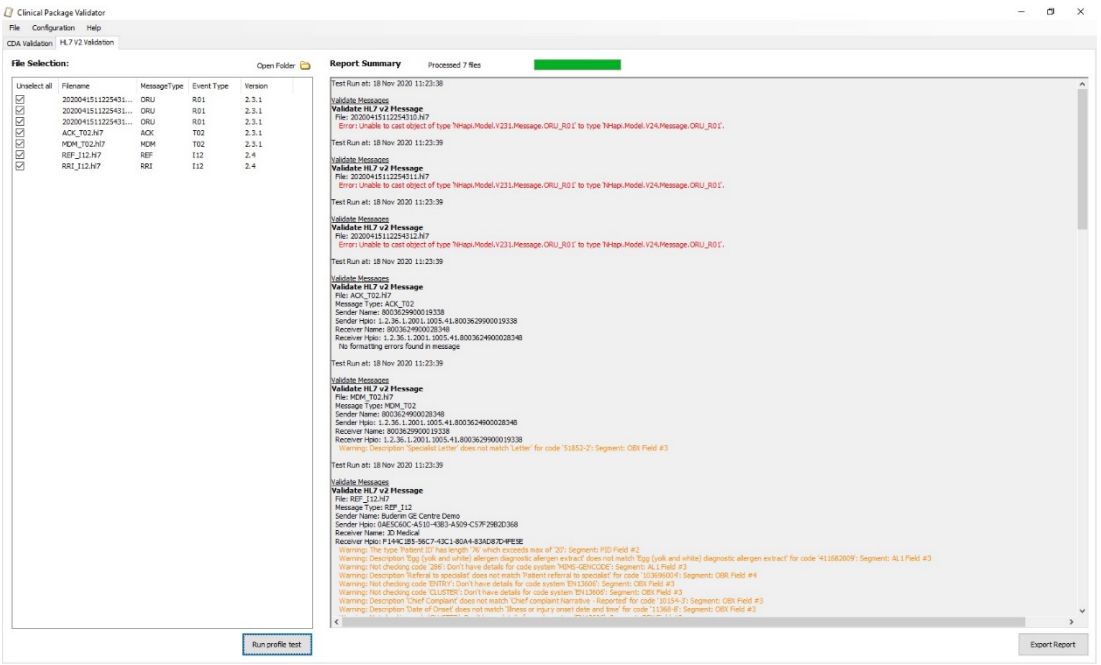


Figure 126 – Report Summary

4 The Report Summary displays detailed information for each file, including Errors/Warnings (Figure 127)

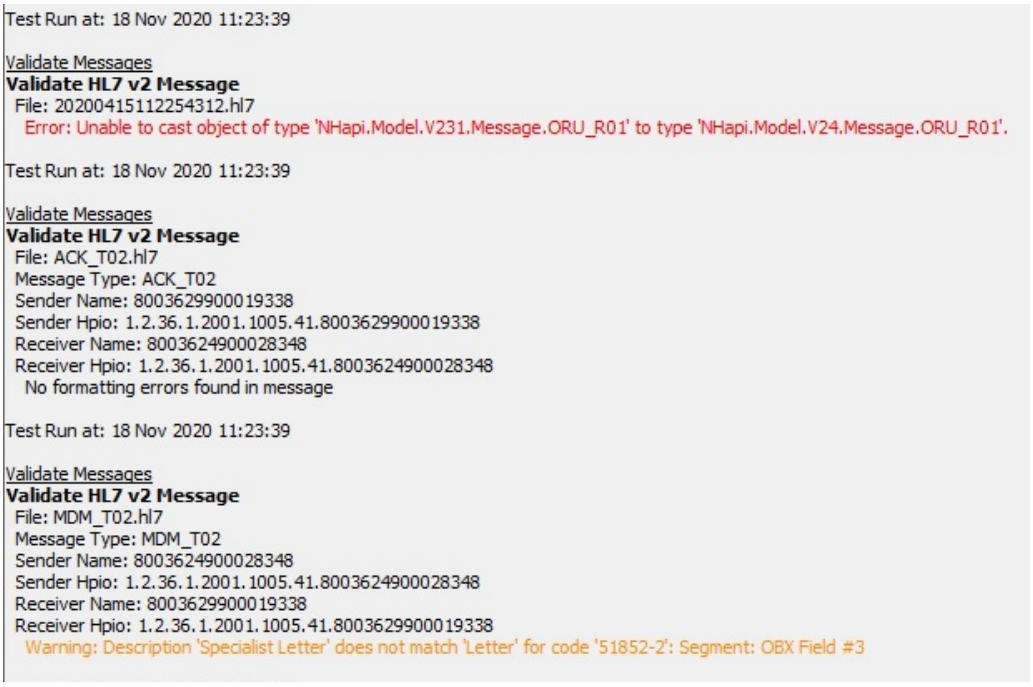


Figure 127 – Report Summary

6 Batch validation

6.1 CDA validation

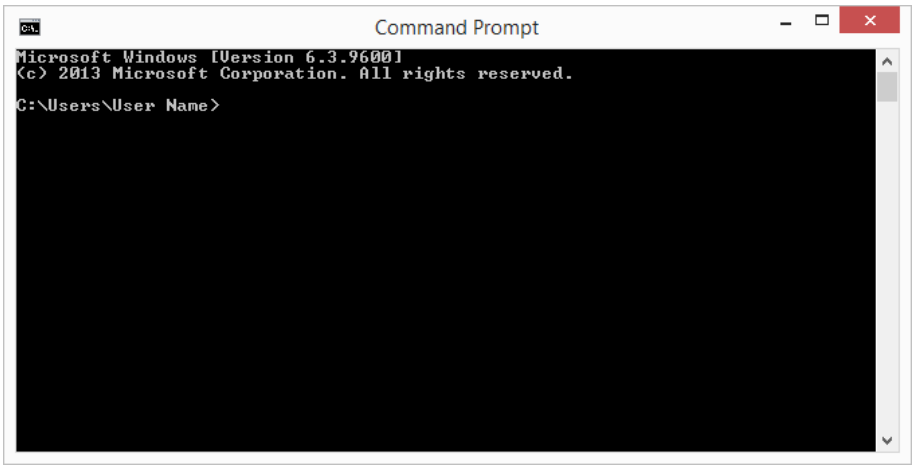
As an alternative to using the graphical user interface of the Validator, the command line interface can be used to perform template, package, and code validation (for which the Validator may be used), and to produce test reports in XML format. The command line interface is used to validate batches of clinical documents and clinical packages. To confirm when the Validator can be used please refer to the *Clinical Package Validator Product Data Sheet* and confirm the intended use with the Agency Help Desk on 1300 901 001.

The Validator's command line interface is intended for use by software developers and testers using scripts or a batch file. For example, a software developer can create a script to command the Validator to validate a file, or to analyse test reports to determine whether any errors were found or if an expected error was reported. The script is invoked through the Validator command line interface.

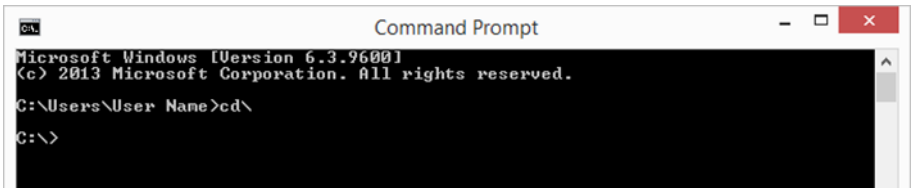
6.1.1 Using the command line interface

To validate a file using the Validator command line interface.

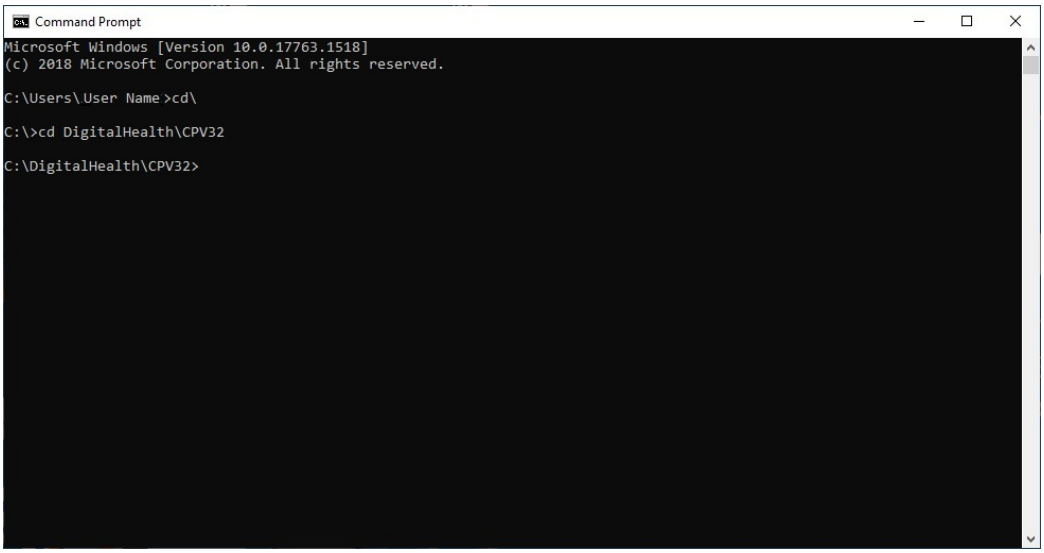
1 Open the Windows Command Prompt by selecting Start -> All programs -> Accessories, and then clicking Command Prompt.
Note: The username that is displayed will vary from user to user.



2 Navigate to the root directory by typing the command cd\ and then press Enter.



3 Navigate to the root directory by typing the command cd\ and then press Enter.



4 Launch the Validator through the command line interface by typing the command:

`CPvalidator.exe FileDirectory FileToValidate TemplateDirectory Target IncludePDFReport AdditionalRulesTemplateFolder`
where:

a `FileDirectory` is the absolute path to the directory that contains file to be validated;

b `FileToValidate` is the file/pattern of files to be validated;

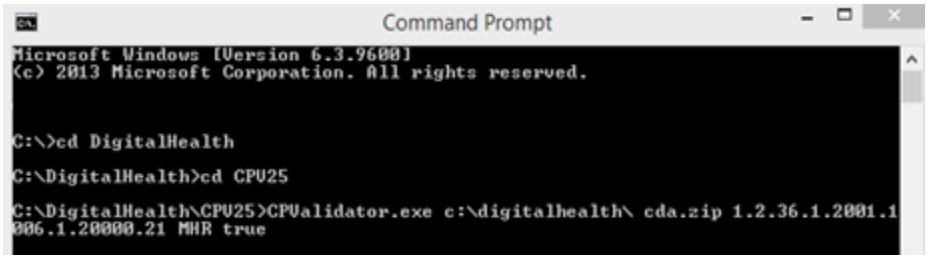
c `TemplateDirectory` is the OID of the template package which must have been downloaded already through the GUI;

d Target is the context for package validation, i.e. My Health Record or P2P;

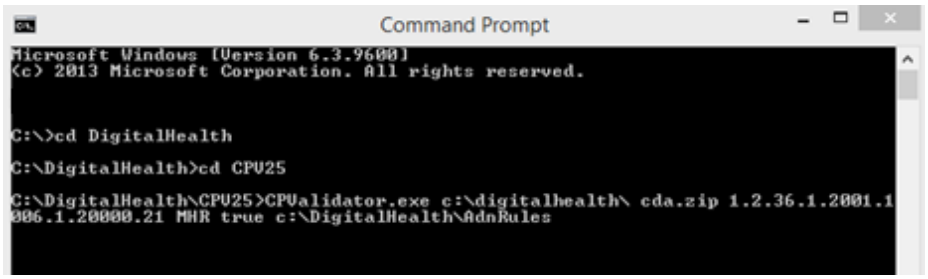
e `IncludePDFReport` is either True or False depending on if you require the PDF summary report (which is saved and located in the same directory as where the target file has been selected from); and

f `AdditionalRulesTemplateFolder` is the absolute path to the root directory that contains the template package with additional validation rules.

If no additional rules are to be applied during package validation, the `AdditionalRulesTemplatePath` can be excluded from the above command, as below.



If, however, additional rules are also to be applied during package validation, the `AdditionalRulesTemplatePath` should be included in the above command, as below, and press Enter.



When using the command line interface, there are a number of useful tips:

a Make sure to execute the command from the C:\DigitalHealth\CPV232 directory as the executable needs a number of files in order to run.

b Put quotation marks around each parameter, as for example some directory names have spaces in their names.

c For directory paths, use a double backslash as the application interprets single backslashes as escape characters

d For processing multiple files at the same time, you can only process files of the same document type, given that you have to specify the template type.

Below are some examples of how to use the command line interface.

Drop into a command prompt and type:

```
CD C:\DigitalHealth\CPV32
```

Example 1: single file with no PDF report

```
cpvalidator "c:\\temp\\" "cda.xml" "1.2.36.1.2001.1006.1.170.5" "MHR" "false"
```

Example 2: multiple files with a PDF report

```
cpvalidator "c:\\temp\\" "pres*.xml" "1.2.36.1.2001.1006.1.170.5" "MHR" "true"
```

Example 3: single file using an additional rules template

```
cpvalidator "c:\\temp\\" "cda.xml" "1.2.36.1.2001.1006.1.170.5" "MHR" "false"
"c:\\temp\\AdnRulesTemplate"
```

Example 4: single zip file testing for Point to Point (P2P) rules and a report

```
cpvalidator "c:\\temp\\" "cda.zip" "1.2.36.1.2001.1006.1.170.5" "P2P" "true"
```

Example 5: single file using an additional rules template and no template package

```
cpvalidator "c:\\temp\\" "cda.xml" "" "MHR" "false" "c:\\temp\\AdnRulesTemplate"
```

6.1.1.1 Analysing validation results

Validation results are stored in XML files in the "C:\DigitalHealth\CPV32\Temp" directory (Figure 128). If a script is written to validate a batch of clinical packages or clinical documents, it could contain instructions to copy these files to another directory after each validation because these files will be removed when another validation is performed. This is not mandatory as if you request the report PDF, it should contain all the details you need about each file, in addition to the report.csv and analysis.csv file.

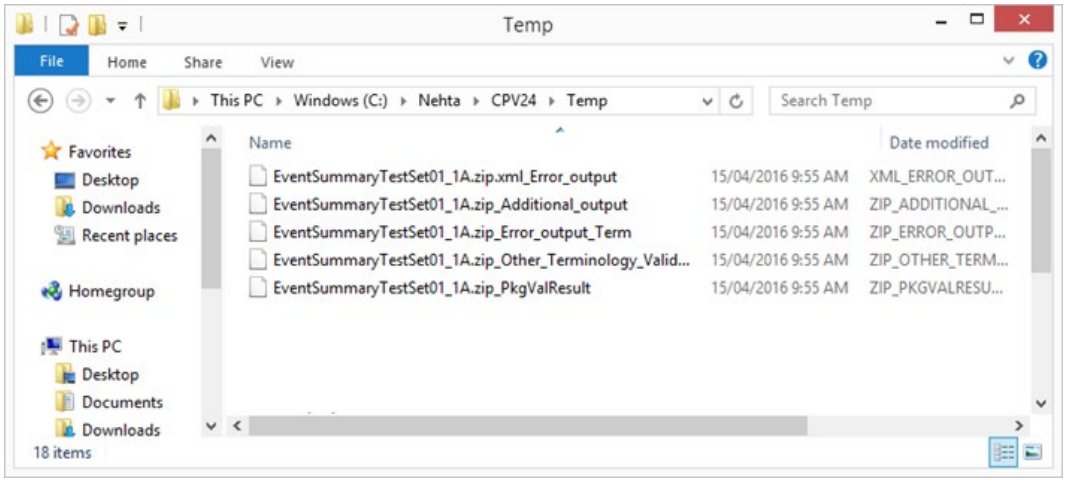


Figure 128 - Storage of validation results including additional template validation results

The XML files created by the Validator to store validation results are listed in Table 11, where <FileName> is the name of the file (i.e. clinical package or clinical document) that was validated.

Table 11 - Files created by the Validator to store validation results

File name	Description
<FileName>.xml_Error_output.xml	An XML report of template validation errors.
<FileName>.xml_PkgValResult.xml	An XML report of package validation errors. This report is only produced if the file that was validated was either a clinical package or an HL7 MDM wrapped clinical package.
<FileName>_Additional_output.xml	An XML report of additional template validation errors. **Note:** This xml file only appears if the AdditionalRulesTemplatePath is used.

The file "report.tsv" (Section 4.2.9) contains a cumulative summary of test results, file "analysis.tsv" (Section 4.2.9) contains a list of the issues, and the XML files listed in Table 11 contain detailed test results. The cumulative summary of test results may be used to complement analysis of the detailed test results.

A user of the Validator may write a script to interrogate the test report XML files. Unlike the graphical user interface where an image is displayed for a test result (Figure 129), the test results in the XML files are stored as references to an image (Figure 130) i.e. the graphical user interface reports a 'Fail' as ✖ whereas the test result XML files report a 'Fail' as <IMG src="No.png" />.

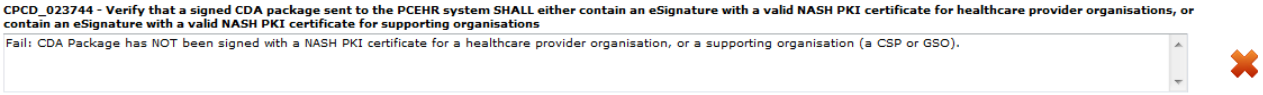


Figure 129 - Graphical user interface with image reporting a 'Fail' for clinical packaging test case CPCD_023744

```
    <TestDataItem>
      <Reference>CPCD_023744</Reference>
      <Message>Fail: CDA Package has NOT been signed with a
NASH PKI certificate for a healthcare provider organisation,
or a supporting organisation (a CSP or GSO).</Message>
      <Description>Verify that a signed CDA package sent to
the PCEHR system SHALL either contain an eSignature with a
valid NASH PKI certificate for healthcare provider
organisations, or contain an eSignature with a valid NASH PKI
certificate for supporting organisations </Description>
      <Result>
        <text>Fail: CDA Package has NOT been signed with a
NASH PKI certificate for a healthcare provider organisation,
or a supporting organisation (a CSP or GSO).</text>
        <image>
          <IMG src="No.png" />
        </image>
      </Result>
    </TestDataItem>
```

Figure 130 - XML file with image reference reporting a 'Fail' for clinical packaging test case CPCD_023744

Scripts that interrogate the test result XML files should search for XML tags with references to an image. They should not search the error message for 'Pass', 'Fail' or 'Warning' text.

Table 12 describes the image reference associated with each test result.

Table 12 - Test results and image references

Test result	Image reference
Pass	<IMG src="Accept.png" />
Fail	<IMG src="No.png" />
Warning	<IMG src="Warning.png" />
Not Run	<IMG src="NA.png" />

6.2 HL7 v2 validation

1 When validating files, there is an option to select files individually or to select all files (Figure 131). Click on the Select all field to select all files.

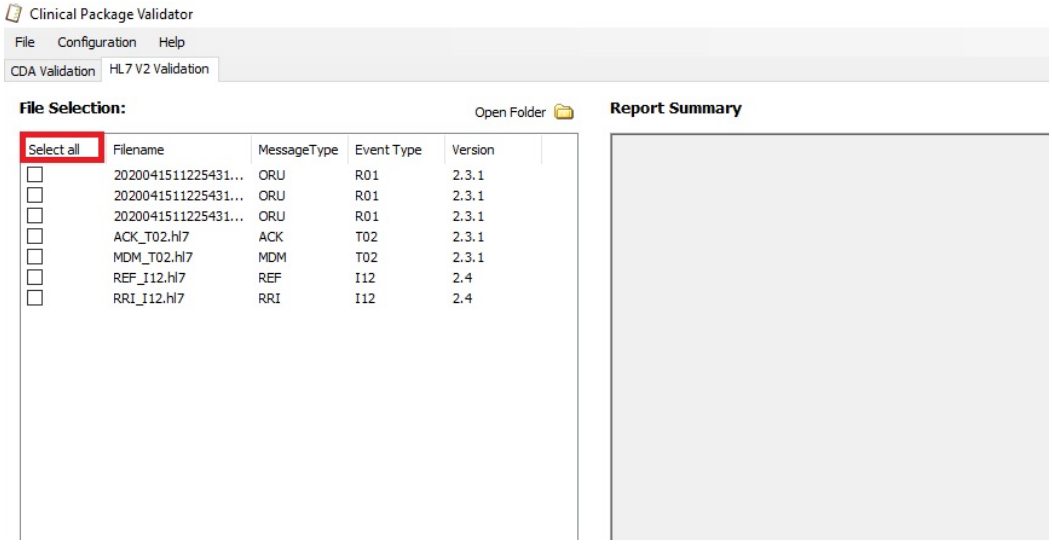


Figure 131 – Select all files

2 There is also the option to Unselect all files (Figure 132)

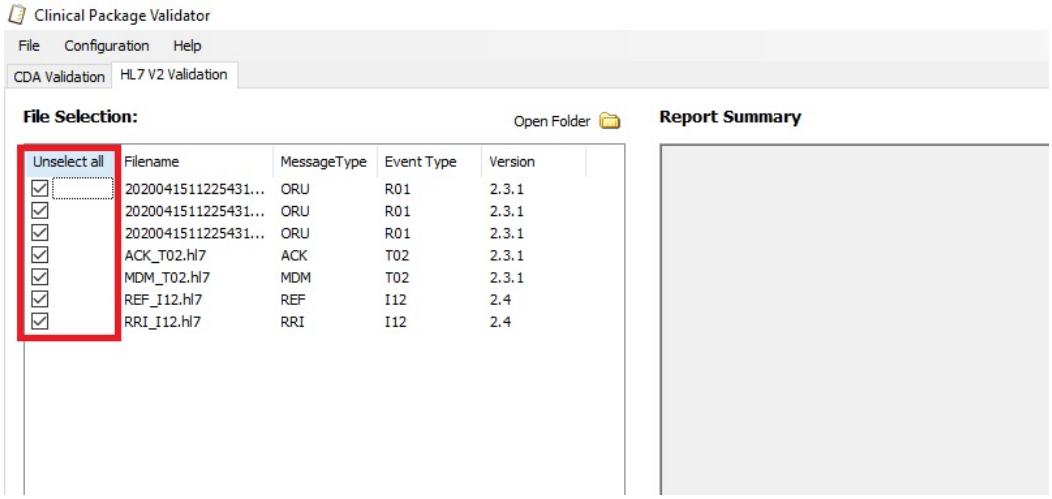


Figure 132 – Unselect all files

Acronyms

Acronym	Description
AMT	Australian Medicines Terminology
ANZSCO	Australian and New Zealand Standard Classification of Occupations
ANZSIC	Australian and New Zealand Standard Industrial Classification
CDA	Clinical Document Architecture
DVA	Department of Veterans' Affairs
HL7	Health Level Seven
HPI-I	Healthcare Provider Identifier - Individual
HPI-O	Healthcare Provider Identifier - Organisation
HTML	Hypertext Mark up Language
IHI	individual healthcare identifier
IHTSDO	International Health Terminology Standards Development Organisation
JPEG	Joint Photographic Experts Group (image format)
MDM	Medical Document Management
MTOM	Message Transmission Optimization Mechanism
NASH	National Authentication Service for Health
OID	object identifier
P2P	provider to provider
PBS	Pharmaceutical Benefits Scheme
PKI	Public Key Infrastructure
SNOMED CT-AU	Systematized Nomenclature of Medicine Clinical Terms - Australia
SOAP	Simple Object Access Protocol
URI	Uniform Resource Identifier
XML	Extensible Markup Language
ZIP	archive file format

Glossary

Term	Meaning
body	The body of a clinical document contains the clinical information.
header	The header of a clinical document contains information about the patient, healthcare provider and administrative details.
narrative block	A narrative block is an XML fragment enclosed within <text> elements. A narrative block contains unstructured narrative text that is to be rendered into human-readable form. The narrative block may contain XML tags that rendering systems use to format the narrative.
SNOMED CT	Systematized Nomenclature of Medicine - Clinical Terms is the internationally pre-eminent clinical terminology that has been identified as the preferred national terminology for Australia and has been endorsed by all Australian governments.
SNOMED CT-AU	SNOMED CT Australian Release (SNOMED CT-AU) is the Australian extension to SNOMED CT, providing local variations and customisations of terms relevant to the Australian healthcare community. It includes the international resources along with all Australian developed terminology.
structured data	Structured data is an XML fragment intended for computer processing. Structured data is not rendered for human readers. Some structured data is referred to as ‘coded data’ as it is associated with a code system.

References

[HL72004] *HL7 Clinical Document Architecture, Release 2.0*; Health Level Seven International, 2004; http://www.hl7.org/

[IHTSDO2014] *SNOMED CT Technical Implementation Guide*; International Health Terminology Standards Development Organization, 2014; http://www.snomed.org/

[ADHA2011a] *CDA Package*, version 1.0; ADHA, 2011; https://www.digitalhealth.gov.au/implementation-resources/clinical-documents/EP-2320-2016/NEHTA-1229-2011

[ADHA2012a] *CDA Rendering Specification*, version 1.0; ADHA, 2012; https://www.digitalhealth.gov.au/implementation-resources/clinical-documents/EP-2320-2016/NEHTA-1199-2012

[ADHA2012b] *P2P Document Delivery Technical Service Specification*, version 1.1; ADHA, 2012; https://www.digitalhealth.gov.au/implementation-resources/clinical-documents/EP-1254-2012/NEHTA-1227-2012

[ADHA2014a] *Australian CDA Schema Extension*, version 3.0; ADHA, 2014; https://www.digitalhealth.gov.au/implementation-resources/ehealth-reference-platform/EP-2344-2016/NEHTA-2150-2014

[ADHA2014b] *PCEHR Document Exchange Service Logical Service Specification*, version 1.3.1; ADHA, 2014; https://www.digitalhealth.gov.au/implementation-resources/national-infrastructure/EP-2109-2015/NEHTA-1970-2014

[ADHA2014c] *PCEHR Document Exchange Service Technical Service Specification*, version 1.5.1; ADHA, 2014; https://www.digitalhealth.gov.au/implementation-resources/national-infrastructure/EP-2109-2015/NEHTA-1971-2014

[ADHA2015b] *Conformance Test Specification for CDA Packaging*, version 1.5; ADHA, 2015; https://www.digitalhealth.gov.au/implementation-resources/clinical-documents/EP-2320-2016/NEHTA-2065-2015

[ADHA2015c] *Clinical Documents - Common Conformance Profile*, version 1.6; ADHA, 2015; https://www.digitalhealth.gov.au/implementation-resources/clinical-documents/EP-2320-2016/NEHTA-1850-2015

[ADHA2015d] *eHealth Prescription Record - PCEHR Conformance Profile*, version 1.3; ADHA, 2015; https://www.digitalhealth.gov.au/implementation-resources/clinical-documents/EP-1919-2015/NEHTA-1912-2015

[HL7OOWG] https://confluence.hl7australia.com/display/OO/Australian+Diagnostics+and+Referral+Messaging+-+Localisation+of+HL7+Version+2.4

[HL7MDM] Ref 2 Medical Document Management (MDM^T02) specification – CDA packaging - HL7 Version 2.3.1 https://developer.digitalhealth.gov.au/specifications/ehealth-foundations/ep-3365-2020/dh-3368-2020

[IQRules] https://developer.digitalhealth.gov.au/products/clinical-package-validator-and-information-quality-rules-iq-rules