Australian Government
Australian Digital Health Agency

Clinical Package Validator
Product Data Sheet

16 June 2023 v3.3
Approved for external use
Document ID: DH-3777:2023

Key features

- automated validation of clinical documents, packages and messages
- Schematron-based validation using dynamically loaded My Health Record template packages
- Schematron-based validation using the Agency's Information Quality Rules (IQ Rules)
- schema-based validation using HL7 v2 and CDA[1] conformance points
- clinical package validation
- clinical terminology validation
- validations in interactive and batch mode

Usage (internal, external)

- developers and implementers of clinical information systems
- developers and implementers of secure messaging systems
- the System Operator of the My Health Record system
- the National Infrastructure Operator (NIO)

Format

- ZIP file
- documentation

[1] HL7 and CDA are trademarks of Health Level Seven International and are registered with the United States Patent and Trademark Office.

1 Overview

The Clinical Package Validator[2] (the Validator) is a tool to automate some of the tests needed to assess conformance of clinical documents, packages and messages with eHealth specifications.

The Validator does not test conformance against all requirements. This *Product Data Sheet* details the sets of tests that are supported, those that are partially supported, and contains a general description of the types of tests that are not supported (see section 6.1.2).

Results from the Validator must not be relied upon to determine software conformity when declaring conformity to the System Operator of the My Health Record system. A developer will need to supplement results from the Validator with their own testing before declaring conformity.

2 Summary of functionality

CDA documents are built to a certain level of conformance, ranging from a basic CDA header accompanied by an attachment, through to a fully populated CDA body with SNOMED CT-AU, Australian Medicines Terminology (AMT) or PBS coding.[3]

XDS metadata is included in SOAP messages for uploading CDA documents to the My Health Record system. The SOAP message may be imported into the Validator which then displays the content of the XDS metadata and validates some of the XDS metadata elements.

For point-to-point information exchange of CDA documents, they need to be encapsulated as clinical packages and wrapped in an HL7 v2.x MDM message.

HL7 v2.x messages are also built to a certain level of conformance, ranging from basic acknowledgements to messages containing various format attachments (i.e. CDA clinical packages, pdf, etc.).

The Validator provides automated, however non-comprehensive, test capabilities for all these cases.

Once the Validator tests have been performed, an on-screen report is generated and displayed. The report can be saved as a record of the automated testing performed by the Validator. A software developer is free to utilise these documents as part of their overall documentation of their work to determine software conformance.

As per the *Non-Production Disclaimer* provided as part of the Validator *Software Package*, the Validator should not be relied on to determine software conformance. The Validator tests need to be accompanied by other test artefacts to demonstrate how a developer satisfied themselves that the software was conformant before declaring conformance to the My Health Record System Operator.

2.1 CDA validation

The Validator tests a CDA document through:

- Validation against the XML schema for CDA documents
- Validation against Schematron rules contained in My Health Record (MHR) template packages
- Validation against Schematron rules contained in the Agency's IQ Rules
- Validation of usage of SNOMED CT-AU, AMT or PBS codes (partial testing only)
- Validation against other terminologies (partial testing only).

[2] The Clinical Package Validator was previously published under the name CDA Validator.

[3] "SNOMED" and "SNOMED CT" are registered trademarks of the International Health Terminology Standards Development Organisation (IHTSDO).

The scope of CDA tests performed by the Validator against Schematron rules is dependent on the rule sets loaded into the Validator at runtime. The Agency publishes two types of Schematron rule sets:

- MHR template packages (based on Schematron v1.5)
- IQ Rules (based on ISO Schematron).

2.1.1 My Health Record template packages

My Health Record template packages are specific to a particular document type at a particular conformance level. Each MHR template package contains Schematron rules to provide automated test coverage for a subset of the conformance requirements for a particular document type at a particular conformance level. These conformance requirements are published across different specification documents for the document type.

They include:

- CDA Implementation Guide
- Structured Content Specification
- MHR Conformance Profile
- other specification documents referenced by any of the above specifications.

MHR template packages are compatible with the My Health Record system and are executed by it for each document upload.

The scope of MHR template packages is limited to the capabilities of the Schematron v1.5 standard.[4]

2.1.2 Information Quality Rules

The Agency also publishes the Information Quality Rules (IQ Rules[5]), which can be applied for all document types and allow for the much more fine-grained validation of clinical documents.

IQ Rules have been developed for use with the Validator and are not compatible with the My Health Record system.

IQ Rules are based on the more recent ISO Schematron standard.[6]

2.1.3 Scope limitation

Neither MHR template packages nor IQ Rules can cover all conformance requirements for any of the clinical document types. Consequently, the Validator cannot cover all conformance requirements. Its use, as it pertains to CDA validation, is strictly limited, as described in section 6.

For the scope of terminology validation against SNOMED CT-AU, AMT, PBS other terminologies, please refer to section 6.2.

2.1.4 Testing of clinical packages

If provided with a clinical package or a clinical package embedded in an HL7 MDM wrapper, the Validator applies a subset of the test cases in *Conformance Test Specification for CDA Packaging v1.5*[7] for both the My Health Record (MHR) and point-to-point (P2P) contexts. See section 6 for details.

Please note The structure of the HL7 MDM message (which contain the clinical package in Base64 format), is tested by the HL7 v2 validation functionality.

[4] ISO IEC 19757-3:2006 [Schematron15]
[5] https://developer.digitalhealth.gov.au/specifications/ehealth-reference-platform/ep-2895-2019
[6] ISO IEC 19757-3:2016 [SchematronISO]
[7] https://developer.digitalhealth.gov.au/specifications/clinical-documents/ep-2807-2019/nehta-2065-2015

2.2 HL7 v2 validation

The Validator tests a HL7 v2 message through:

- Validation against the XML schema profile for HL7 v2 message structure
- Validation against the XML codes table for HL7 and user defined tables
- Validation of usage of SNOMED CT-AU and LOINC

The scope of HL7 v2 tests performed by the Validator is dependent on the profile schemas and related code tables loaded into the Validator at runtime. The Agency publishes two types of rule sets for each clinical message:

- Schema profile
- Code table

2.2.1 Schema profiles

Schema profiles are specific to a particular message and event type (trigger event). Each schema profile contains structure definition and rules to provide automated test coverage for a subset of the conformance requirements for a particular trigger event.

2.2.2 Code tables

The Agency also publishes code tables which can be applied for all trigger events relating to that HL7 version. Code tables allow for small set code validation based HL7 and user defined sets.

2.2.3 Scope limitation

Neither schema profiles nor code tables can cover all conformance requirements for any of the trigger events. Consequently, the Validator cannot cover all conformance requirements. Its use, as it pertains to HL7 v2 validation, is strictly limited, as described in section 7.

For the scope of terminology validation against SNOMED CT-AU and LOINC, please refer to section 7.2.

3 Product components

Table 1 - Clinical Package Validator product components

Component	Comment
Product Data Sheet	This document
Software Package	Contains: • Installer (an .exe file containing the Validator software) • `ISSetupPrerequisites` (a folder of installation scripts for prerequisite software) • Non-production disclaimer.
Installation Guide	A PDF file containing instructions on how to successfully install the Validator.
User and Configuration Guide	A PDF file containing instructions on how to configure and use the Validator to assist testing clinical documents, packages and messages.
Release Note	Notes specific to this release of the Validator.

4 System requirements

The Validator is a Microsoft Windows-based tool and requires .NET 4 framework and SQL CE 3.5 to be installed on the machine, which are included as part of the installation (see the Validator's *Installation and Configuration Guide*). The user is responsible for ensuring relevant licences are in place.

5 Scope of tests

The Validator does not and cannot provide tests for all conformance requirements applicable to clinical documents, packages and messages. Developers are responsible to perform additional tests for all remaining conformance requirements before declaring conformance of their product.

6 CDA validation scope

CDA documents and clinical package test cases are documented in the *Conformance Test Specification for CDA Packaging v1.5*. The Validator's in-scope tests include many of these test cases.

Table 2 specifies the degree to which these test cases are automated by the Validator. Test cases not listed in the table are not in-scope for this version of the Validator. The tests performed depend on whether the MHR or P2P context is selected (see the *User and Configuration Guide* product component for details about contexts).

Developers will need to run their own tests to determine conformance with the full set of requirements for clinical packages.

The following notations are used in Table 2: A (automated), M (manual), N/A (not applicable), and NR (not reported). See Table 3 for further details of these notations.

Table 2 - Scope of tests for CDA validation

Test Case	MHR	P2P	Description
Test set: Context My Health Record			
DEXS-L_154	A	N/A	Verify the package contains only CDA_ROOT.XML, CDA_SIGN.XML and packaged attachments.
DEXS-L_155	A	N/A	Verify the package does not contain INDEX.HTM, README.TXT or repository metadata.
DEXS-L_156	A	N/A	Verify that any attachments are not CDA packages or CDA documents.
DEXS-T_121	A	N/A	Verify that the CDA package contains one, and only one, signature file.
DEXS-T_125	A	N/A	Verify that the packaged attachment files are located in the same folder as the CDA_ROOT.XML document.
CPCD_023741	A	N/A	Verify the CDA package only references attachments that are of the supported MIME type.
CPCD_024629	A3	N/A	Verify the CDA package only references packaged attachments where the filename extensions are those listed for the supported MIME types.
CPCD_024630	A3	N/A	Verify all packaged attachments have filename extensions which matches their MIME type.
CPCD_023743	A	N/A	Verify the CDA package is not larger than 10MB.
CPCD_023744	A1	N/A	Verify the eSignature for the CDA package has been signed with a NASH PKI certificate for a healthcare provider organisation, or a supporting organisation (a CSP or GSO).
Test set: Context P2P			
P2P_T13	N/A	A	Verify that the CDA package zip does not contain repository metadata.
CPCD_023748	N/A	A1	Verify that the eSignature for the signed CDA package has been signed with a NASH PKI certificate for a healthcare provider organisation.
Test set: Base CDA Package			
PKG_CDA_002	A	A	Verify the base CDA package contains one and only one root entry.

Test Case	MHR	P2P	Description
PKG_CDA_004	A	A4	Verify that, for every reference to a packaged attachment, there is a corresponding document in the CDA package.
PKG_CDA_005	N/A	A	Verify that, for every reference to a CDA package packaged attachment, there is a corresponding item in the CDA package.
PKG_CDA_006	NR	M1	Verify the list of eSignatures in a CDA package.
PKG_CDA_007	A	A5	Verify all eSignatures are valid eSignatures.
Test set: Signed CDA Package			
PKG_CDA_013	A	A	Verify the CDA package contains one or more eSignatures.
Test set: XDM-ZIP Representation			
XDM_ZIP_105	N/A	N/A	Verify the zip file is a valid XDM-Zip CDA package.
XDM_ZIP_106	A	A	Verify there is one submission set.
XDM_ZIP_108	A	A	Verify the CDA document has the filename CDA_ROOT.XML.
XDM_ZIP_109	A	A	Verify that an eSignature has the filename CDA_SIGN.XML.
Test set: eSignature			
PKG_CDA_024	A	A	Verify eSignature XML document conforms to a Signed Container defined by ATS 5821-2010[8] and the root element is a signedPayload element.
PKG_CDA_025	A	A	Verify the Signed Payload contains only one ds:Signature element.
XSP_SCP_000	A	A	Verify that the eSignature Signed Payload is valid against the Signed Payload and Signed Payload Data XML schemas.
XSP_SCP_001	A	A	Verify the id attribute is unique within the signed XML document.
XSP_SCP_002	A	A	Verify the sp:signedPayloadData element is the only element signed by all the signatures.
XSP_SDP_000	A6	A6	Verify that the Signature element of the signedPayload conforms to the XML Signature Syntax and Processing Recommendation from W3C and is valid against the XML Signature Schema.
XSP_SDP_001	A	A	Verify that a detached signature is used by ensuring the signature element references an XML document or element in the same document.
XSP_SDP_002	A	A	Verify that Exclusive XML Canonicalisation was used on the signed contents in the ds:SignedInfo element.
XSP_SDP_003	A	A	Verify the Algorithm attribute = 'http://www.w3.org/2001/10/xml-exc-c14n#' for the ds:CanonicalisationMethod element.
XSP_SDP_004	A	A	Verify the Algorithm attribute = 'http://www.w3.org/2000/09/xmldsig#rsa-sha1' for the ds:SignatureMethod element.

[8] *ATS 5821-2010 E-health XML secured payload profiles*, Standards Australia, available from: http://infostore.saiglobal.com/store/details.aspx?ProductID=1391034

Test Case	MHR	P2P	Description
XSP_SDP_004_1	A	A	Verify the ds:SignatureMethod algorithm was used to calculate the signature value.
XSP_SDP_005	A	A	Verify there are one or more ds:Reference elements in the ds:SignedInfo element.
XSP_SDP_006	A	A	Verify that each ds:Reference element in ds:SignedInfo contains a URI attribute.
XSP_SDP_007	A	A	Verify the URI attribute of ds:Reference has a '#' character followed by a fragment identifier.
XSP_SDP_008	A	A	Verify the fragment identifier after the '#' character matches the id attribute in sp:signedPayloadData.
XSP_SDP_009	A	A	Verify a ds:Transforms element is present in ds:Reference element.
XSP_SDP_010	A	A	Verify there is only one ds:Transform element in the ds:Transforms element.
XSP_SDP_011	A	A	Verify the Exclusive XML Canonicalisation algorithm was used on the content being signed.
XSP_SDP_012	A	A	Verify the Algorithm attribute = 'http://www.w3.org/2001/10/xml-exc-c14n#' for the ds:Transform element.
XSP_SDP_013	A	A	Verify the value of ds:DigestValue element matches value calculated using SHA-1 algorithm on Exclusive XML Canonicalisation of signed payload.
XSP_SDP_014	A	A	Verify the Algorithm attribute = 'http://www.w3.org/2000/09/xmldsig#sha1' for the ds:DigestMethod element.
XSP_SDP_015	A	A	Verify the ds:KeyInfo element is present in ds:Signature.
XSP_SDP_016	A	A	Verify the ds:X509Data element is present in ds:KeyInfo element.
XSP_SDP_017	A	A	Verify the ds:X509Certificate element is present in ds:X509Data element.
XSP_SDP_018	A	A	Verify X509Certificate element contains the encoded value of the signing certificate.
XSP_SDP_019	A	A	Verify that ds:Signature element does not contain a ds:Object element.
PKG_CDA_026	A	A	Verify the sp:signedPayloadData element contains only one s:eSignature element.
PKG_CDA_027	A	A	Verify there is one ds:Reference element in the Manifest, and that it is set to the SHA-1 digest of the root XML document.
PKG_CDA_029	A2	A2	Verify that the person that approved the eSignature can be identified from the value contained in the s:approver element of the signature.
PKG_CDA_030	A	A	Verify the s:signingTime element contains a valid time.
PKG_CDA_031	A	A	Verify the s:signingTime element includes an explicit timezone.

Test Case	MHR	P2P	Description
Test set: CDA XML Document			
PKG_CDA_014	A	A	Verify the CDA document is valid for its document type.
PKG_CDA_015	A	A	Verify any packaged attachments are represented using an ED-type element.
PKG_CDA_016	A	A	Verify the ED-type element integrityCheckAlgortihm = 'SHA-1'.
PKG_CDA_017	A	A	Verify the ED-type element contains a single cda:reference element.
PKG_CDA_018	A	A	Verify reference element has a 'value' attribute containing a valid URI.
PKG_CDA_019	A	A	Verify that any ED-element using SHA-1, containing a single reference and the reference value is a URI identical to the name of a document in the CDA package, refers to a packaged attachment.
PKG_CDA_020	A	A	Verify the integrityCheck attribute of any ED-element matches the SHA-1 digest of the referenced atomic packaged attachment.
PKG_CDA_021	A3	A3	Verify the mediaType attribute of any ED-type elements is of an agreed Internet type.
PKG_CDA_022	N/A	A	Verify the integrityCheck attribute of any ED-type elements matches the SHA-1 digest of any one eSignature inside the CDA package packaged attachment.
PKG_CDA_023	N/A	A	Verify the mediaType attribute of any ED-type elements is 'application/x.electronichealth.cda.package' for a CDA package packaged attachment.
CDAR_AS_050	A	A	Verify that, for each linkHtml element in the CDA document, if the linkHtml element contains a relative reference, then verify the document being referenced exists in the CDA package.
CDAR_AS_053-06	M2	M2	Verify that no attachments contain executable code (e.g. JavaScript code in HTML documents).
CDAR_AS_053-07	A	A	Verify that no referenced attachments require resources to be downloaded from external network locations, unless the document is of the type that allows references to objects outside of the CDA package.
CDAR_AS_053-10	A	A4	Verify that all referenced attachments are located within the CDA package and are in accordance with the *CDA Packaging* specification, unless the document is of the type that allows references to objects outside of the CDA package.
PKG_024732	A	A	Verify that, where the document is of the type that allows references to objects outside of the CDA package and the reference is to an external location or website, the reference is a non-zero length string containing a Uniform Resource Identifier (URI).
PKG_024988	A	A	Verify that, where the document is of the type that allows references to an object outside the CDA package and the reference is to an external atomic attachment or website, the reference is represented by an ED-element.

Test Case	MHR	P2P	Description
PKG_025077	A	A	Verify that, where the document is of the type that allows references to objects outside of the CDA package and the reference is to an atomic attachment, it has an approved media type and filename extension.
PKG_024990	M2	M2	Verify that, where the document is of the type that allows references to attachments outside the CDA package and the reference is to an external atomic attachment and an integrityCheck is included, the integrityCheck attribute has a value that is the SHA-1 digest of the byte stream and the value of the integrityCheckAlgorithm attribute (if included) is SHA-1.
Test set: Clinical Package			
PKG_PKG_009	NR	M1	Verify that each identifier associated with a member of the CDA package is not an empty value and conforms to the URI specification.
PKG_PKG_010	NR	M1	Verify that each identifier associated with a member of the CDA package is unique.

Table 3 – Key for Table 2

A	The test case is automated without exceptions.
A_1	The test case is automated with the exception of the test step requiring the validation of the NASH certificate by accessing the NASH service to determine if the certificate is on the revocation list.
A_2	The test case is automated with the exception of the test step requiring the validation of the healthcare identifier by accessing the Healthcare Identifiers Service.
A_3	The test case is automated; however the Validator does not report the correct result if the clinical document attachment is compressed (requirements for compressed attachments are listed in the HL7 Version 3 Standard: Data Types – Abstract Specification, Release 1).
A_4	The test case is automated; however the Validator does not report the correct result if a clinical document attachment is in a different folder to the clinical document (this is allowed in the P2P context and disallowed in the My Health Record context).
A_5	The test case is automated with the exception that the only eSignature that is verified is the one with filename CDA_SIGN.XML (multiple eSignatures are allowed in the P2P context and disallowed in the My Health Record context).
A_6	The test case is automated but only covers verification of the most common errors. The full test case should be performed using the guidance provided in version 1.5 of the *Conformance Test Specification for CDA Packaging*.
M_1	The Validator does not report any outcome for the test case. The test result is determined only after a manual inspection of the package index if one is present in the clinical package (a package index is allowed in the P2P context and disallowed in the My Health Record context).
M_2	The Validator does not report any outcome for the test case. The test case may be performed using the guidance provided in version 1.5 of the *Conformance Test Specification for CDA Packaging*.
NR	The test result is not reported by the Validator as it cannot fail in the selected context.
N/A	The test result is not reported by the Validator as the test case is not applicable for the selected context.

6.1 Scope of Schematron validation

The actual tests for clinical documents depend entirely on the MHR template packages provided to the Validator by the developer.

The Agency publishes MHR template packages for the following types of clinical documents:

- Advance Care Directive Custodian Record
- Australian Childhood Immunisation Register
- Australian Immunisation Register
- Australian Organ Donor Register
- Birth Details
- Child Parent Questionnaire
- Consumer Entered Achievements
- Consumer Entered Measurements
- Discharge Summary
- eHealth Diagnostic Imaging Report
- eHealth Dispense Record
- eHealth Pathology Report
- eHealth Prescription and Dispense View
- eHealth Prescription Record
- eReferral
- Event Summary
- Health Check Assessment
- Health Check Schedule View
- Medicare DVA Benefits Report
- Medicare Overview
- Observation View
- Personal Health Notes (previously known as Consumer Entered Notes)
- Personal Health Summary (previously known as Consumer Entered Health Summary)
- Pharmaceutical Benefits Report
- Shared Health Summary
- Specialist Letter.

MHR template packages for these document types are available from the Agency's website[9] or can be downloaded from the Software Vendor Test (SVT) environment of the My Health Record system.

[9] https://www.digitalhealth.gov.au/implementation-resources/clinical-documents

The Schematron rules provided by any of these MHR template packages are not comprehensive in their coverage of conformance requirements for clinical documents. Developers are responsible for performing their own testing to supplement the Validator's automated tests before declaring conformance for their product.

The Validator is preloaded with the comprehensive set of template packages for the following document types:

Document type name	Conf. level	Template Package ID	Template Package Version	MHR Conformance Profile Version number	MHR Conformance Profile Publication date
Shared Health Summary	3A	1.2.36.1.2001.1006.1.16565.10	36326	Version 1.6	17-Apr-15
Shared Health Summary	3B	1.2.36.1.2001.1006.1.16565.11	36326	Version 1.6	17-Apr-15
Event Summary	3A	1.2.36.1.2001.1006.1.16473.14	36678	Version 1.4	17-Apr-15
Event Summary	3B	1.2.36.1.2001.1006.1.16473.15	36678	Version 1.4	17-Apr-15
Discharge Summary	1A	1.2.36.1.2001.1006.1.20000.23	40382	Version 1.5	09-Oct-13
Discharge Summary	1B	1.2.36.1.2001.1006.1.20000.24	40382	Version 1.5	09-Oct-13
Discharge Summary	2	1.2.36.1.2001.1006.1.20000.25	40382	Version 1.5	09-Oct-13
Discharge Summary	3A	1.2.36.1.2001.1006.1.20000.26	40382	Version 1.5	09-Oct-13
Discharge Summary	3B	1.2.36.1.2001.1006.1.20000.27	40382	Version 1.5	09-Oct-13
e-Referral	1A	1.2.36.1.2001.1006.1.21000.18	32624	Version 1.4	09-Oct-13
e-Referral	1B	1.2.36.1.2001.1006.1.21000.19	32624	Version 1.4	09-Oct-13
e-Referral	2	1.2.36.1.2001.1006.1.21000.20	32624	Version 1.4	09-Oct-13
e-Referral	3A	1.2.36.1.2001.1006.1.21000.21	32624	Version 1.4	09-Oct-13
e-Referral	3B	1.2.36.1.2001.1006.1.21000.22	32624	Version 1.4	09-Oct-13
Specialist Letter	1A	1.2.36.1.2001.1006.1.16615.28	35416	Version 1.5	30-Sep-14
Specialist Letter	1B	1.2.36.1.2001.1006.1.16615.29	35416	Version 1.5	30-Sep-14
Specialist Letter	2	1.2.36.1.2001.1006.1.16615.30	35416	Version 1.5	30-Sep-14
Specialist Letter	3A	1.2.36.1.2001.1006.1.16615.31	35416	Version 1.5	30-Sep-14
Specialist Letter	3B	1.2.36.1.2001.1006.1.16615.32	35416	Version 1.5	30-Sep-14

Document type name	Conf. level	Template Package ID	Template Package Version	MHR Conformance Profile Version number	MHR Conformance Profile Publication date
eHealth Dispense Record	3A	1.2.36.1.2001.1006.1.171.5	36140	Version 1.3	20-Feb-15
eHealth Prescription Record	3A	1.2.36.1.2001.1006.1.170.5	36140	Version 1.3	20-Feb-15
Diagnostic Imaging Report	3A	1.2.36.1.2001.1006.1.222.4	40364	Version 1.1	Mar 2016
Pathology Report	3A	1.2.36.1.2001.1006.1.220.4	40368	Version 1.1	Mar 2016
Pharmacist Shared Medicines List	1A	1.2.36.1.2001.1006.1.237.1	40647	Version 1.1	Jan 2019
Goals of Care	3A	1.2.36.1.2001.1006.1.100001.2	40657	Version 1.0	Mar 2020

The following sections give general descriptions of the types of tests that are either fully or partially *excluded* from the Validator and its MHR template packages.

6.1.1 Tests that must be done manually

The following types of tests are not (and cannot be) automated by the Validator:

- Tests for the equivalence between clinical information in a clinical document in atomic format versus the clinical information in the narrative blocks in a clinical document.
- Tests for consistency between the definition of a component of a clinical document (i.e. a section, data group or data element) and the information in that component.
- Clinical document tests that can only be performed by looking up an external software system (e.g. the healthcare identifiers service, the HL7 OID registry).
- Validation of the values of codes for most code systems (e.g. SNOMED CT-AU, AMT, PBS, clinical specialty codes, Australian vaccine codes). The exception is the validation of codes where the set of values is listed in the CDA implementation guides.

6.1.2 Tests not included in this release

The following tests are not included in the Validator, but may be included in a future release:

- My Health Record usability recommendations related to the content of a clinical document.
- Requirements for terminology codes (e.g. a code must be a specific length for it to be an AMT code).

- Requirements for clinical document narrative blocks, stated in the CDA Rendering Specification[10] and the HL7 CDA R2 specification[11] (e.g. the requirement for the allowed set of mark-up content; the requirements for the inclusion of style codes).
- Data elements in a clinical document that are individually conformant but, when combined, do not make sense and are likely to result from an error in the clinical information system (e.g. having a fully structured address without the state; having a healthcare provider's contact details listed as their home details).

6.1.3 Tests partially supported in this release

The following tests are only partially supported by the Validator in this release:

- Requirements in specifications not published by the Agency but which must be implemented in order to conform to the CDA implementation guides, e.g., Health Care Client Identification (SA 5017); HL7 Data Types v1 specification; HL7 CDA R2 specification.
- Requirements for identifiers in a clinical document (e.g. the same identifier should be used for every occurrence of an object in a clinical document, and different objects must not have the same identifier).
- Requirements for entity identifiers and entitlements (e.g. a Medicare card number cannot be used for an entity identifier for a person; a local entity identifier must contain the 'extension' attribute).
- Requirements for data types used in the Agency's structured content specifications (e.g. the requirements for CodeableText, CodedText, Duration, Quantity).
- Requirements in the conformance profile for a specific type of document. The coverage depends on the type of clinical document. For example, Agency-published MHR template packages do not enforce any requirements in the conformance profile for eHealth Prescription Record.

Given conformance with these requirements is only partially tested by the Validator, a developer needs to perform their own testing for these requirements.

6.2 Scope of clinical terminology validation

With some exceptions, the values of codes for code systems are *not tested* by the Validator. The exceptions are the validation of codes from the code systems listed in Table 4 where the set of values is listed in the CDA implementation guides.

Table 4 - Scope of tests for clinical terminology validation

Code system	Version	Date of publication	Source
Australian Medicines Terminology (AMT)	V2.54 V2.55 V2.56 V3 20170228	28 Mar 2014 28 Apr 2014 30 May 2014 28 Feb 2017	https://www.healthterminologies.gov.au/ncts/#/learn?content=documentlibrary
Australian and New Zealand Standard Industrial Classification (ANZSIC)	2006	2006	http://www.abs.gov.au/ausstats/abs@.nsf/mf/1292.0

[10] https://developer.digitalhealth.gov.au/specifications/clinical-documents/ep-2807-2019/nehta-1199-2012
[11] https://www.hl7.org/implement/standards/cda.cfm

Code system	Version	Date of publication	Source
Australian and New Zealand Standard Classification of Occupations (ANZSCO)	First edition, revision 1	25 June 2009	http://www.abs.gov.au/ausstats/abs@.nsf/Lookup/1220.0Main+Features1First%20Edition,%20Revision%201
Australian Vaccines codes	-	-	http://www.humanservices.gov.au/health-professionals/services/australian-childhood-immunisation-register/acir-vaccine-code-formats
Clinical specialty codes	-	21 Nov 2013	http://meteor.aihw.gov.au/content/index.phtml/itemId/329673
HL7 identifier types	V2.6	2007	https://www.hl7.org/special/committees/vocab/V26_Appendix_A.pdf
HL7 service delivery role types	V3	7 Aug 2008	https://www.hl7.org/implement/standards/fhir/v3/ServiceDeliveryLocationRoleType/index.html
PBS codes (Item codes and Manufacturer codes)	Mar 2017	1 Mar 2017	http://www.pbs.gov.au/browse/downloads
SNOMED CT-AU reference set	20170228	28 Feb 2017	https://www.healthterminologies.gov.au/ncts/#/learn?content=documentlibrary

7 HL7 v2 validation scope

HL7 v2 conformance requirements are documented in Table 5.

Table 5 - Scope of schema profiles published by the Agency

HL7 version	Message types	Specification
2.3.1	MDM, ACK	https://collaborate.digitalhealth.gov.au/display/SMT/MDM%5ET02+Specification
2.4	REF, RRI, ORM, ORR, ORU, ACK	https://confluence.hl7australia.com/pages/viewpage.action?pageId=1279290

The Validator's in-scope tests include many of these conformance requirements.

Table 6 specifies which of these conformance requirements are validated by the Validator. Conformance requirements not listed in the table are not in scope for this version of the Validator.

Developers will need to run their own tests to determine conformance with the full set of requirements for HL7 v2 messages.

The following classification is used in Table 6:

- V: Fully validated by the Validator based on the schema profiles and code tables provided
- PV: Partially validated by the Validator with comments outlining what the validator checks and what it doesn't

Table 6 - Scope of tests for HL7 v2 validation

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:000003	ORM^O01, ORU^R01, REF^I12, ORR^O02	To ensure the uniqueness of the Entity Identifier (EI) in OBR-2 (*Placer Order Number)* for a request identifier across different organisations, the Entity identifier (first component) in addition to the Namespace ID (second component) and Universal ID (third component) and Universal ID type (4th component) must be populated.	V	ORC-2 placer order number is the same as OBR-2 placer order number. *Does not apply for Referrals as ORC and OBR exist in different segment groups not possible to determine link given the cardinality* Field optionality is still allowed ie. field may be completely unvalued
HL7au:000004.1	ORM^O01, ORU^R01, REF^I12, ORR^O02	To ensure the uniqueness of the Entity Identifier (EI) in OBR-3 (*Filler Order Number*) for a request identifier across different organisations, the Entity identifier (first component) in addition to the Namespace ID (second component) and Universal ID (third component) and Universal ID type (4th component) must be populated.	V	ORC-3 filler order number is the same as OBR-3 filler order number. *Does not apply for Referrals as ORC and OBR exist in different segment groups not possible to determine link given the cardinality.* *For Referrals, OBR-3 is mandatory.*
HL7au:000005	ORM^O01, ORU^R01, REF^I12, ORR^O02	To ensure the uniqueness of the Entity Identifier (EI) in ORC-2 (*Placer Order Number)* for a request identifier across different organisations, the Entity identifier (first component) in addition to the Namespace ID (second component) and Universal ID (third component) and Universal ID type (4th component) must be populated.	V	ORC-2 placer order number is the same as OBR-2 placer order number. *Does not apply for Referrals as ORC and OBR exist in different segment groups not possible to determine link given the cardinality* Field optionality is still allowed ie. field may be completely unvalued

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:000006	ORM^O01, ORU^R01, ORR^O02	To ensure the uniqueness of the Entity Identifier (EI) in ORC-3 (*Filler Order Number*) for a result identifier across different organisations, the Entity identifier (first component) in addition to the Namespace ID (second component) and Universal ID (third component) and Universal ID type (4th component) must be populated.	V	
HL7au:000007	ORM^O01, ORU^R01, REF^I12, ORR^O02	To ensure the uniqueness of the Entity Identifier (EI) in ORC-4 (*Placer Group Number*) for a request identifier across different organisations, the Entity identifier (first component) in addition to the Namespace ID (second component) and Universal ID (third component) and Universal ID type (4th component) must be populated.	V	
HL7au:000008	ORU^R01, REF^I12	The message must contain at least one OBX display segment per OBR/OBX group.	V	
HL7au:000008.1	ORU^R01, REF^I12	Display segments must use the appropriate valid values within the AUSPDI coding system in OBX-3 for the content that is represented in it: OBX||ED|HTML^Display format in HTML^AUSPDI||^text^HTML^A^<?xml OBX||ED|PDF^Display format in PDF^AUSPDI||^application^pdf^Base64^ OBX||ED|RTF^Display format in RTF^AUSPDI||… OBX||FT|TXT^Display format in text^AUSPDI||… (Referral Level 1 receivers are only required to support PDF display segments. See HL7au:000008.3.1)	V	

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:000008.1.2	ORU^R01	An OBX display segment is identified using OBX-3 Identifier (CE-1) and Name of Coding System (CE-3) components. The text component of the CE may be blank and only CE1 and CE-3 components need to match.	PV	Only be tested for Name of Coding System (CE-3) being either "LN" (LOINC) or "SCT" (SNOMED-CT) or "AUSPDI". For LN and SCT, this is done using a call to the configured terminology server to determine if the code exists. For AUSPDI, this is done using <hl7table="SPDI/>
HL7au:000008.1.3	ORU^R01	In an OBX display segment, the OBX-2 Value Type field must match its corresponding display format specified in OBX-3 Identifier (ST) component as per table Display Format codes in Section 4.5.	PV	This can only be tested for "ED" and "FT": OBX||ED|HTML^Display format in HTML^AUSPDI||^text^HTML^A^<?xml OBX||ED|PDF^Display format in PDF^AUSPDI||^application^pdf^Base64^ OBX||ED|RTF^Display format in RTF^AUSPDI||… OBX||FT|TXT^Display format in text^AUSPDI||…
HL7au:000008.1.4	ORU^R01	In an OBX display segment, the OBX-3 <name of coding system (IS)> must be valued "AUSPDI".	PV	This only applies if OBX-2 is "ED" and "FT"
HL7au:000008.1.5		The OBX display segment(s) must be the last in a set of OBX segments in each OBR/OBX group, with the exception of digital signature OBX(s) which may be after the display segments OBXs. (Display segments can be identified by having AUSPDI OBX-3 <name of coding system>)	V	
HL7au:000008.3.1	REF^I12	For Referrals Level 1: The single OBR/OBX group of the message must contain an OBX display segment in PDF format. For other profiles: Each OBR/OBX group of the message must contain at least one of the following OBX display segments HTML, PDF, TXT (HL7 FT).	V	

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:000019	All	A single message of up to 16 MB (16,777,216 bytes) must be able to be received by both the transmitters and receivers of messages.	V	Warning message will be displayed for messages larger than 16MB
HL7au:000020	All	All message types and trigger event codes beginning with the letter “Z” are reserved for locally-defined messages and must NOT be used.	V	
HL7au:000021	ORU^R01, REF^I12	Data type TX must NOT be used as a value in the OBX-2 Value Type field.	V	
HL7au:000023	All	The NTE segment must NOT be used in messages.	V	
HL7au:000024.2	All	FHS, BHS, and MSH segments must specify the Components separator character as '^'	PV	BHS and FHS are not supported
HL7au:000024.3	All	FHS, BHS, and MSH segments must specify the Sub-components separator characteras '&'	PV	BHS and FHS are not supported
HL7au:000024.4	All	FHS, BHS, and MSH segments must specify the repeat separator character as '~'	PV	BHS and FHS are not supported
HL7au:000024.5	All	FHS, BHS, and MSH segments must specify the escape separator character as '\'	PV	BHS and FHS are not supported
HL7au:000028	ORU^R01	When there are multiple OBR segments in an ORU message, the OBR-3 *Filler order number* must be unique within messages.	V	
HL7au:000028.2	REF^I12	When there are multiple OBR/OBX groups in a REF message, each OBR-3 Filler order number pair must be unique for each OBR/OBX group.	V	

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:000032	ORU^R01	In the ORU message the field OBR-24 "Diagnostic serv sect ID" must be valued and must have values from HL7 table 0074 - diagnostic service section.	V	
HL7au:000032.2	REF^I12	In the REF message the field OBR-24 "Diagnostic serv sect ID" must be valued and must have values from HL7 table 0074 - diagnostic service section appropriate for the content in the OBR/OBX group.	V	
HL7au:000034.1	ORM^O01, ORU^R01, REF^I12	When using CE data types in an OBX segment in either OBX-3 (Observation Identifier) or as an Observation Value, if the system transmits both the public (e.g. LOINC) and local terminology, then the public (eg LOINC) code must appear in the identifier.	V	If the data type declared in OBX-2 to determine datatype in OBX-5 is CE then "name of coding system" has to be either LOINC or SNOMED. OBX-3 will always be CE data type
HL7au:000034.2	ORM^O01, ORU^R01, REF^I12	When using CE data type in an OBX segment, in OBX-3 (Observation Identifier), if the system transmits both a public (e.g. LOINC) and a local terminology, then the local terminology must be transmitted in the second CE triplet i.e. the alternate identifier.	V	If the data type declared in OBX-2 to determine datatype in OBX-5 is CE, then the local terminology must be transmitted in the second CE triplet i.e. the alternate identifier. OBX-3 will always be CE data type
HL7au:000040.1	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	Senders conforming to this specification must specify "2.4" as the value of version ID (ID) component of MSH-12 Version ID (VID)	V	
HL7au:000040.2	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	MSH-12 Version ID <internationalization code (CE)> component must be valued "AUS&Australia&ISO3166_1"	V	

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:000040.3	ORM^O01, ORU^R01, ORR^O02, ACK^R01, ACK^O01	MSH-12 Version ID <internal version ID (CE)> component must be valued as "HL7AU-OO-201701&&L". (Note that the number scheme used in this identifier is HL7 date format: YYYYMM)	V	
HL7au:000040.4	REF^I12, RRI^I12	MSH-12 Version ID <internal version ID (CE)> component must be valued as "HL7AU-OO-REF-SIMPLIFIED-201706&&L" (for Level 2) or "HL7AU-OO-REF-SIMPLIFIED-201706-L1&&L" (for Level 1). (Note that the number scheme used in this identifier is HL7 date format: YYYYMM)	V	Only Level 1 Referrals supported
HL7au:000041	All	MSH-17 Version ID must be valued as "AUS"	V	
HL7au:000040.5	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	MSH-19 must be valued as "en^English^ISO639".	V	
HL7au:00044.1.1	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12	CX <ID (ST)> component must be specified and valid according to the identifier scheme of selected by the Identifier type code and Assigning Authority components.	V	
HL7au:00044.1.2	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12	CX <assigning authority (HD)> component must be valued	PV	Enforced through profile. If <universal ID type> is "VDI", a warning should be displayed if the assigining authority <namespace id> is not found in table 0363, rather than an error

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:00044.1.3	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12	CX <identifier type code (ID)> component must be valued with a valid value from HL7 Table 0203 - Identifier type.	V	
HL7au:00044.3.1	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12	The EI Entity identifier component must be valued	V	
HL7au:00044.4.1	All	When an <identifier (ST)> component is specified, the <name of the coding system> must also be specified.	V	Conditional processing applied to all CE data types where no constant values exists
HL7au:00044.4.2	All	If no <identifier (ST)> component is specified then no <name of coding system> (primary coding system) must be specified	V	Conditional processing applied to all CE data types where no constant values exists
HL7au:00044.4.5	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	When an <alternate identifier (ST)> component is specified, the <name of alternate coding system> must also be specified.	V	Conditional processing applied to all CE data types where no constant values exists
HL7au:00044.4.6	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	If no <alternate identifier (ST)> component is specified then no <name of alternate coding system> must be specified	V	Conditional processing applied to all CE data types where no constant values exists

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:00044.4.8	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	Alternate coding system must be a different from the primary coding system. As the 2 codes must describe the same concept the alternate text is optional.	V	Conditional processing applied to all CE data types where no constant values exists
HL7au:00044.6.1	ORM^O01, ORU^R01, REF^I12, ORR^O02	When an <identifier (ST)> component is specified, the <name of the coding system> must also be specified.	V	Conditional processing applied to all CWE data types where no constant values exists
HL7au:00044.6.2	ORM^O01, ORU^R01, REF^I12, ORR^O02	If no <identifier (ST)> component is specified then no <name of coding system> must be specified	V	Conditional processing applied to all CWE data types where no constant values exists
HL7au:00044.6.4	ORM^O01, ORU^R01, REF^I12, ORR^O02	When an <alternate identifier (ST)> component is specified, the <name of alternate coding system> must also be specified.	V	Conditional processing applied to all CWE data types where no constant values exists
HL7au:00044.6.5	ORM^O01, ORU^R01, REF^I12, ORR^O02	If no <alternate identifier (ST)> component is specified then no <name of alternate coding system> must be specified	V	Conditional processing applied to all CWE data types where no constant values exists
HL7au:00044.7.1	All	XCN <ID (ST)> component must be specified and valid according to the identifier scheme of selected by the Identifier type code and Assigning Authority components.	PV	Syntax enforced through profile, but semantic validity not enforced.
HL7au:00044.7.2	All	XCN <assigning authority (HD)> component must be valued and valid.	V	
HL7au:00044.7.3	All	XCN <name type code (ID)> component must be valued and valid from HL7 Table 200.	V	
HL7au:00044.7.4	All	XCN <identifier type code (ID)> component must be valued with a valid value from HL7 Table 203.	V	
HL7au:00044.7.5	All	XCN <family name (FN)> :<surname (ST)> sub-component must to be valued.	V	

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:00044.7.6	All	XCN <given name (ST)> should be valued.	V	
HL7au:00044.10.1.1	ORU^R01, REF^I12, MDM^T02, ORM^O01	ED <type of data (ID)> must be valued.	V	
HL7au:00044.10.1.2	ORU^R01, REF^I12, MDM^T02, ORM^O01	ED <data subtype (ID)> must be valued.	V	
HL7au:00044.10.1.3	ORU^R01, REF^I12, MDM^T02, ORM^O01	ED <encoding (ID)> must be valued.	V	
HL7au:00044.10.1.4	ORU^R01, REF^I12, MDM^T02, ORM^O01	ED <data (ST)> must be valued.	V	
HL7au:00044.10.1.5	ORU^R01, REF^I12, MDM^T02, ORM^O01	When the ED <subtype (ID)> component is valued with a MIME sub-type value, then the corresponding MIME type must be used in the <Type of data (ID)> component.	V	
HL7au:00044.10.1.6	ORU^R01, REF^I12, MDM^T02, ORM^O01	When the ED <subtype (ID)> component is valued with a HL7 2.4 defined <Subtype (ID)> (Table 0291) value, then the corresponding HL7 2.4 type of data (Table 0191) must be used in the <Type of data (ID)> component.	V	
HL7au:00046.1.1	All	Senders must escape | characters as '\F\' in all fields, components, subcomponents	V	

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:00046.1.2	All	Senders must escape '^' characters as '\S\' in all HL7 fields, components and subcomponents	V	
HL7au:00046.1.3	All	Senders must escape '&' characters as '\T\' in all HL7 fields, components and subcomponents	V	
HL7au:00046.1.4	All	Senders must escape '~' characters as '\R\' in all HL7 fields, components and subcomponents	V	
HL7au:00046.1.5	All	Senders must escape '\' characters as '\E\' in all HL7 fields, components and subcomponents	V	
HL7au:00046.3	All	All fields required by HL7 segments table must be validly valued.	V	
HL7au:00046.4	All	Receiving implementations when receiving HL7 messages and converting their contents to data values must ignore segments, fields, components, sub-components, and extra repetitions of a field that are present but were not expected.	V	Provides warning (not error) when condition met.
HL7au:00046.5	All	Receiving implementations when receiving HL7 messages and converting their contents to data values must treat segments that were expected but are not present as consisting entirely of fields that are not present	V	Provides warning (not error) when condition met.
HL7au:00046.6	All	Receiving implementations when receiving HL7 messages and converting their contents to data values must treat fields and components that are expected but were not included in a segment as not present.	V	Provides warning (not error) when condition met.

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:00047.1	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	MSH-15 Accept acknowledgement type (ID) must be valued AL	V	
HL7au:00047.2	All	MSH-16 Application acknowledgement type (ID) must be valued AL	V	
HL7au:00048.1	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	When MSH-18 is unvalued or valued as "ASCII" the message must contain only characters in the range ASCII 32 to ASCII 127 and cursor return ASCII 13 which must only be used as segment separator.	V	
HL7au:00048.2	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	When MSH-18 is valued and is not "ASCII" encoding the message must not contain characters less than ASCII 32, except for ASCII 13 which must only be used as segment separator.	V	
HL7au:00048.3.1 (r3)	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	MSH-18 must only contain one of the following values "", "ASCII" or by site agreement "UNICODE UTF-8", "8859/1" may be used.	V	It is hoped to allow "UNICODE UTF-8" in a future version, but this depends on widespread receiver support. Receivers are encouraged to develop capability for UTF-8. ** errata "UNICODE UTF-8" was incorrectly added as "UTF-8"

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:00048.3.2	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	MSH-18 must only contain one of the following values "", "ASCII".	PV	Assume "" to mean ASCII.
HL7au:00048.3.3	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	The encoding of characters in the message must match the value specified in MSH-18	V	
HL7au:00049.1	All	MSH-9 Message type <message type (ID)>component must be valued.	V	
HL7au:00049.2	All	MSH-9 Message type <trigger event (ID)> component must be valued.	V	
HL7au:00049.3	All	MSH-9 Message type <message structure (ID)>component must be valued.	V	
HL7au:00050.1.5	ORU^R01	The OBX-6 (Units) <name of coding system (IS)> component must be "UCUM".	V	
HL7au:00060.1	All	HL7 message elements with a usage of R (required) must be valued.	V	
HL7au:00060.3	ORM^O01, ORU^R01, REF^I12, ORR^O02, RRI^I12, ACK^I12, ACK^R01, ACK^O01	HL7 message elements with a usage of C (conditional) must be valued when the associated predicate is satisfied.	PV	Predicates aren't included in profile, so only conditional validation through code where applicable.

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:00101.1	ORU^R01	Each OBR/OBX group may contain arbitrary encapsulated data attachments as per 4.26 Encapsulated data attachments.	V	
HL7au:00101.2	ORU^R01, REF^I12, MDM^T02	Encapsulated data attachments must use Base64 encoding.	V	
HL7au:00104.1.1	REF^I12, RRI^I12	There must be exactly one PRD with a PRD-1 value of "AP" (Authoring Provider) in the REF message.	V	
HL7au:00104.2.1	REF^I12, RRI^I12	There must be exactly one PRD with a PRD-1 value of "IR" (Intended Recipient) in the REF message.	V	
HL7au:00104.7.0	REF^I12, RRI^I12	PRD-7 must have at least 1 repeat.	V	
HL7au:00104.7.2.1	REF^I12, RRI^I12	PRD-7 <type of ID number (IS)> must be valued from User-defined Table 0363 - Assigning Authority.	V	
HL7au:00104.7.3.1	REF^I12, RRI^I12	<other qualifying info (ST)> must be a valued from HL7 Table 0203 - Identifier Type.	V	
HL7au:00110.1	All	Senders must populate the MSH-3 Sending application (HD) field components as per the values specified in the provider directory of their secure messaging system being used.	PV	Check that the MSH-3 fields are populated from a syntax (i.e. values exist as per conformance profile), but cannot check the semantics (i.e. the values are as they appear in the provider directory)

HL7au Identifier	Trigger event applicability	Conformance point text	Validator classification	Comments
HL7au:00110.2	All	Senders must populate the MSH-4 Sending facility (HD) field components as per the values specified in the provider directory of their secure messaging system being used.	PV	Check that the MSH-4 fields are populated from a syntax (i.e. values exist as per conformance profile), but cannot check the semantics (i.e. the values are as they appear in the provider directory)

7.1 Scope of schema profile validation

The Validator dynamically loads the profiles structure and rules from the XML schema profiles and code tables provided to it at runtime. The actual tests for clinical messages depend entirely on the XML schema profiles and code tables provided to the Validator by the developer.

The Agency publishes XML schema profiles as listed in Table 7 below:

Table 7 - Scope of schema profiles published by the Agency

HL7 version	Trigger event	Description
2.3.1	MDM^T02	Original document notification and content message
2.3.1	ACK^T02	T02 acknowledgement message
2.4	REF^I12	Patient referral message
2.4	RRI^I12	Referral response message
2.4	ACK^I12	I12 acknowledgement message
2.4	ORM^O01	Order message
2.4	ORR^O02	Order response message
2.4	ACK^O01	O01 acknowledgement message
2.4	ORU^R01	Unsolicited observation message
2.4	ACK^R01	R01 acknowledgement message

7.1.1 Tests that must be done manually

The following types of tests are not (and cannot be) automated by the Validator:

- Tests for the equivalence between clinical information in a clinical message in atomic format versus the clinical information in the narrative blocks of a clinical message.

- Tests that can only be performed by looking up an external software system (e.g. the healthcare identifiers service, provider directory).
- Tests that can only be performed by correctly displaying and rendering the clinical information contained with the message (e.g. correctly displaying pathology results in a clinical information system.

7.2 Scope of clinical terminology validation

The values of codes for code systems that are tested by the Validator are listed below:

Table 8 - Code systems

Code system	Version	Source
LOINC	2.68	https://www.healthterminologies.gov.au/specs/v3/content-types/loinc/
SNOMED CT-AU	AU - Latest	https://www.healthterminologies.gov.au/specs/v3/content-types/snomed-ct/

The Validator not only checks that the code exists in the code system but it also compares the text in the message versus that stored in the clinical terminology service as reports warnings if anomalies are found.

8 Downloads

The Clinical Package Validator is available in the Products area of the Agency's website, or simply through the following link:

https://developer.digitalhealth.gov.au/products/clinical-package-validator

9 References

Reference	Description
SchematronISO	*Information technology — Document Schema Definition Languages (DSDL) — Part 3: Rule-based validation, Schematron*, International Standard ISO/IEC 19757-3:2016, Geneva, Switzerland : ISO https://standards.iso.org/ittf/PubliclyAvailableStandards/c055982_ISO_IEC_19757-3_2016.zip
Schematron15	*Information technology — Document Schema Definition Languages (DSDL) — Part 3: Rule-based validation, Schematron*, International Standard ISO/IEC 19757-3:2006, Geneva, Switzerland : ISO

Publication date: 16 June 2023

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

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