



Customer Information Quality Specifications Version 3.0 – Committee Specification 02 Package Overview

20 September 2008

Specification URIs:

This Version:

<http://docs.oasis-open.org/ciq/v3.0/cs02/supp/ciq-package-overview-v3.html>
<http://docs.oasis-open.org/ciq/v3.0/cs02/supp/ciq-package-overview-v3.doc>
<http://docs.oasis-open.org/ciq/v3.0/cs02/supp/ciq-package-overview-v3.pdf>

Previous Version:

<http://docs.oasis-open.org/ciq/v3.0/prd03/supp/ciq-package-overview-v3.html>
<http://docs.oasis-open.org/ciq/v3.0/prd03/supp/ciq-package-overview-v3.doc>
<http://docs.oasis-open.org/ciq/v3.0/prd03/supp/ciq-package-overview-v3.pdf>

Latest Version:

<http://docs.oasis-open.org/ciq/v3.0/cs02/supp/ciq-package-overview-v3.html>
<http://docs.oasis-open.org/ciq/v3.0/cs02/supp/ciq-package-overview-v3.doc>
<http://docs.oasis-open.org/ciq/v3.0/cs02/supp/ciq-package-overview-v3.pdf>

Technical Committee:

OASIS Customer Information Quality

Chair(s):

Ram Kumar (kumar.sydney@gmail.com)

Editor(s):

Ram Kumar (kumar.sydney@gmail.com)

Related work:

This version of the CIQ specifications replaces or supercedes OASIS CIQ V3.0 Committee Specification released in November 2007

Abstract:

This document provides an overview of the CIQ Specification V3.0 Committee Specification 02 package that is available for download from the OASIS CIQ Technical Committee (TC) web site (<http://www.oasis-open.org/committees/ciq>).

Status:

This document was last revised or approved by the OASIS CIQ TC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at www.oasis-open.org/committees/ciq.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (www.oasis-open.org/committees/ciq/ipr.php).

The non-normative errata page for this specification is located at www.oasis-open.org/committees/ciq.

Notices

Copyright © OASIS® 1993–2008. All Rights Reserved. OASIS trademark, IPR and other policies apply.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", "CIQ", "xNL", "xAL", "xNAL", "xPIL", "xPRL", "xCIL", "xCRL", "Genericcode", and "UBL" are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	ABOUT THE PACKAGE.....	5
1.1	NAME OF THE PACKAGE	5
1.2	FOR EXISTING USERS OF OASIS CIQ V3.0 COMMITTEE SPECIFICATION RELEASED IN NOVEMBER 2007	5
1.2.1	<i>If no changes to any files of “OASIS CIQ V3.0.zip”</i>	<i>5</i>
1.2.2	<i>If changes were done to file(s) of “OASIS CIQ V3.0.zip”</i>	<i>5</i>
1.3	FOR NEW USERS - EXTRACTING THE PACKAGE	6
1.4	CIQ SPECIFICATION ENTITY XML SCHEMAS USING DEFAULT/STANDARD CODE LIST APPROACH.....	6
1.5	CIQ SPECIFICATION ENTITY XML SCHEMAS USING GENERICODE BASED CODE LIST APPROACH	8
1.6	VERIFYING TEST RESULTS BY TESTING THE CUSTOMISED TEST FILES OF CODE LISTS PROVIDED	15
1.7	VERIFYING TEST RESULTS BY TESTING THE CUSTOMISED TEST FILES OF CODE LISTS PROVIDED	16
1.8	VERIFYING THE TEST RESULTS OF METADATA ATTRIBUTES OF THE CUSTOMISED CODE LISTS	17
2	CUSTOMISING YOUR CODE LISTS / ENUMERATIONS	18
2.1	USING OPTION 1 FOR CODE LISTS (DEFAULT)	18
2.2	USING OPTION 2 FOR CODE LISTS (GENERICODE APPROACH)	18
2.2.1	<i>Pre-requisites.....</i>	<i>18</i>
2.2.2	<i>XML Parsers.....</i>	<i>18</i>
2.2.3	<i>Known Bug</i>	<i>18</i>
2.2.4	<i>Steps to “Prepare and Test” the two pass validation when modifying the supplied default package to customise default genericode.....</i>	<i>18</i>
2.2.4.1	<i>Steps to test two pass validation when the default package is not modified</i>	<i>19</i>
A.	ACKNOWLEDGEMENTS	20
B.	INTELLECTUAL PROPERTY RIGHTS, PATENTS, LICENSES AND ROYALTIES	21
C.	REVISION HISTORY	22

1 About the Package

The purpose of this document is to assist users who have downloaded the “**OASIS CIQ V3 CS02.zip**” (Committee Specification 02 of OASIS CIQ V3.0) package from the OASIS CIQ TC web site (<http://www.oasis-open.org/committees/ciq>) to understand the contents of the package and how to install and use them.

1.1 Name of the Package

The name of the package is “**OASIS CIQ V3.0 CS02.zip**”.

1.2 For existing users of OASIS CIQ V3.0 Committee Specification released in November 2007

This section is only applicable to users who have downloaded and are using CIQ Version 3.0 Committee specification package (**OASIS CIQ V3.0.zip**) that was released in November 2007. This section is not applicable to users of this specification package.

In January 2008, few minor issues with CIQ Version 3.0 Committee Specifications were identified and logged in a change log document. The changes made to the files of the above package (OASIS CIQ V3.0.zip) are now included as part of this revised specification. Details about the issues are documented in “**ciq-v3-change-log**” file (.doc, .pdf or .html) under “supp” directory. If you are already using OASIS CIQ V3.0 Committee Specification package (**OASIS CIQ V3.0.zip**) that was released in November 2007, this package supercedes it as it includes the changed files and therefore, these fixes **SHOULD** have an impact on your implementation and use. Details about the impact and how to implement the changes are documented in this section.

1.2.1 If no changes to any files of “OASIS CIQ V3.0.zip”

If no changes to any files of OASIS CIQ V3.0.zip were done, replace your current installation with this new installation as described from section 1.3 onwards.

1.2.2 If changes were done to file(s) of “OASIS CIQ V3.0.zip”

Following changes could have been possibly done by you:

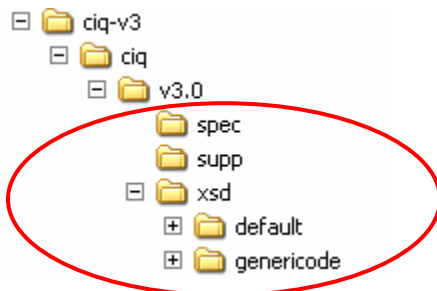
- Changes the file path names in the batch or shell files
- Adding/Updating/Deleting data in any of the enumeration files (codelist - option 1) or genericcode related files (.bat, .sh, .cva, .gc, .sch, .xml).
- Changes to example files or adding or deleting example files
- Changes to .xsd files

Read “**installing-fixes-to-ciqv3-cs**” document (.doc or .pdf or .html) under “supp” package on how to install the changed files as part of the current implementation of OASIS CIQ V3.0 specifications package that was released in November 2007.

Existing users of OASIS CIQ V3.0 specification do not need to read the remaining sections of this document.

1.3 For new users - Extracting the Package

Extracting the downloaded package (use C: root drive to install) creates the following directory structure:



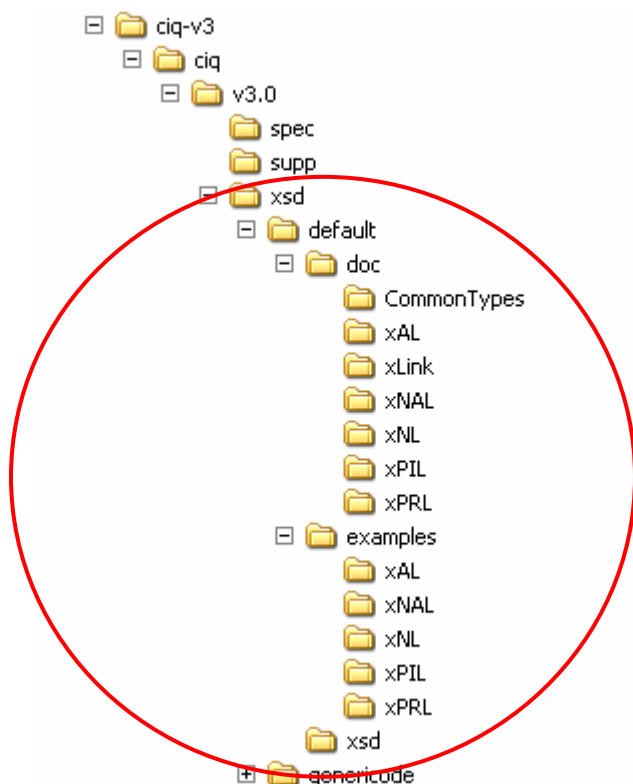
Directory Name	Contents
spec	Contains the document describing the Name, Address, Name and Address, and Party specifications
supp	Contains supporting documents namely, introduction to CIQ TC, technical overview, release notes, this document, and technical FAQ
xsd	Contains the directory for CIQ entity XML schemas. Classified into two parts, 1. Default CIQ entity XML schemas using Option 1 of Code List, 2. CIQ entity XML schemas using Option 2 of Genericcode based Code List
xsd/default	Contains default CIQ entity XML schemas (xNL, xAL, xNAL, and xPIL) using Option 1 of Code List and XML schema documentation (HTML)and sample XML document instances for entities
xsd/genericcode	Contains CIQ entity XML schemas using Option 2 of Genericcode based Code List and XML schema documentation and sample XML document instances for entities. Also contains all code lists represented using genericcode, utilities to run the two pass validation, and batch/shell files to prepare two pass validation

1.4 CIQ Specification Entity XML Schemas using Default/Standard Code List Approach

CIQ Specification entity XML schemas are available in two types:

- One set uses default code list approach (Option 1 – all code lists are represented as XML schemas (*xNL-types.xsd*, *xAL-types.xsd*, *xNAL-types.xsd*, and *xPIL-types.xsd*) and “included” in entity XML schemas (*xNL.xsd*, *xAL.xsd*, and *xPIL.xsd*).
- The other set uses genericcode based code list approach (Option 2 – all code lists are represented in genericcode)

This section outlines the structure of Option 1. Users who are not interested in genericcode based code list approach, should concentrate on the following marked directory structure only.

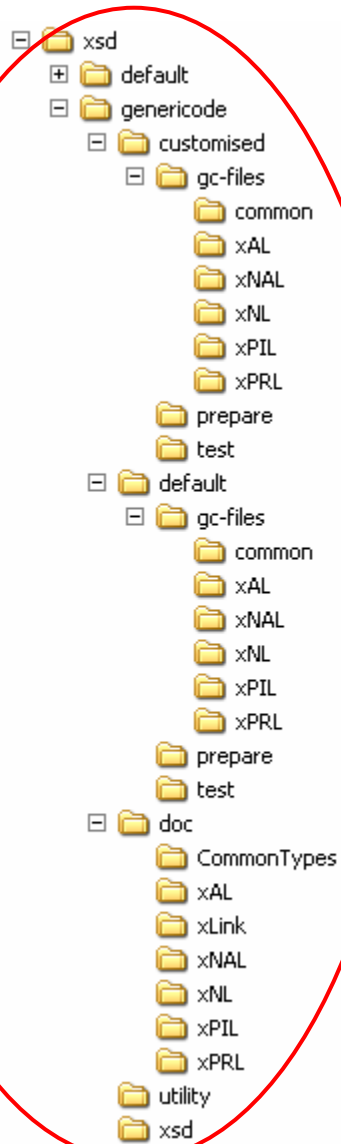


56

Directory Name	Contents
xsd/default/xsd	<p>Contains the default entity XML schemas for Name, Address and Party.</p> <ul style="list-style-type: none"> • xNL.xsd – xNL schema for Name entity. <u>Users must not modify</u> this file. • xAL.xsd – xAL schema for Address entity. <u>Users must not modify</u> this file. • xPIL.xsd – xPIL schema for Party entity. <u>Users must not modify</u> this file. • CommonTypes.xsd – Schema reused by all the above entity schemas. <u>Users must not modify</u> this file. • xNL-types.xsd – Defines all code lists and values for xNL.xsd. <u>Users can modify</u> this file. • xAL-types.xsd – Defines all code lists and values for xAL.xsd. <u>Users can modify</u> this file. • xNAL-types.xsd – Defines all code lists and values for xNAL.xsd. <u>Users can modify</u> this file. • xPIL-types.xsd – Defines all code lists and values for xPIL.xsd. <u>Users can modify</u> this file.
xsd/default/doc	<p>Provides HTML documentation for all default XML schemas (xNL, xAL, xNAL, xPIL, CommonTypes and xLink) in individual sub directories</p>
xsd/default/examples	<p>Contains XML sample files for Name, Address, Name and Address, and Party Schemas using Option 1 for code lists</p>

1.5 CIQ Specification Entity XML Schemas using Genericcode based Code List Approach

This section outlines the structure of Option 2. Users who are interested in genericcode based code list approach, should concentrate on the following marked directory structure only.



Directory Name	Contents
xsd/genericcode/default	This directory contains all default genericcode files along with files for preparing genericcodes and test files. <u>Users should not modify files under this structure</u> as everything is prepared for the user as part of this package. Users should only apply constraints on these default genericcodes and this is done in a separate directory (xsd/genericcode/customised)
xsd/genericcode/default/gc-files	This sub-directory contains all the default genericcode files to support CIQ Specification entity schemas namely, xNL, xAL, xNAL, and xPIL
xsd/genericcode/default/common	Contains the common genericcode code list files (2) used by Name, Address and Party XML schemas.
xsd/genericcode/default/xAL	Contains the common genericcode code list files (32) used by Address XML schema (xAL.xsd).
xsd/genericcode/default/xNL	Contains the common genericcode code list files (13) used by Name XML schema (xNL.xsd).
xsd/genericcode/default/xNAL	Contains the common genericcode code list file (2) used by Name and Address XML schema (xNAL.xsd).
xsd/genericcode/default/xPIL	Contains the common genericcode code list files (60) used by Party XML schema (xPIL.xsd).
xsd/genericcode/default/xPRL	Empty Directory
xsd/genericcode/default/prepare	<p>This directory provides all files required to prepare the default genericcode files. Users should not modify any of the files in this directory as it has already been prepared for this as part of this package. However, if the default genericcode files are changed, then the files to prepare for validation should be used.</p> <ul style="list-style-type: none"> • prepare-default-cl.bat (.sh) – editable windows batch file (or shell file) to prepare the genericcode representation of the default code lists for two pass validation. This is the “main” program that executes other batch programs. This file <u>does not require modification</u> by the user as it has been already updated. Users should define appropriate relative paths in this file if they change the default directory structures • default-cl-constraints.cva – lists all of the genericcode expressions of agreed-upon default value list value enumerations, and lists all of the default contexts in which the value enumerations are used. All constraints for CIQ are already defined and <u>requires no modifications</u> to this file by the user. Users should define appropriate relative paths in this file if they change the default directory

Directory Name	Contents
	<p>structures</p> <ul style="list-style-type: none"> • DefaultCodeList.sch – Defines the default code list namespace constraints. This file <u>requires no modification</u> as all required constraints have been added. Users should define appropriate relative paths in this file if they change the default directory structures <p>Other files not listed above– All the other files in this directory are automatically generated when prepare-default-cl.bat/sh file is executed and must <u>NOT</u> be modified by user and so, do not touch them</p>
xsd/genericcode/default/test	<p>This directory provides files to test the default genericcode lists by performing two pass validations. <u>Users can modify the .xml files to test different cases.</u> Sample test files have been provided for users to test them.</p> <ul style="list-style-type: none"> • test-all.bat (.sh)– editable windows batch file (or shell file) to test xNL, xAL, xNAL, and xPIL sample document instance (using default genericcode based code lists) using two pass validation. Users should define appropriate relative paths in this file if they change the default directory structures • test-default-xnl.bat (.sh)– editable windows batch file (or shell file) to test xNL sample document instance (using default genericcode based code lists) using two pass validation. Users should define appropriate relative paths in this file if they change the default directory structures • test-default-xal.bat (.sh)– editable windows batch file (or shell file) to test xAL sample document instance (using default genericcode based code lists) using two pass validation. Users should define appropriate relative paths in this file if they change the default directory structures • test-default-xnal.bat (.sh)– editable windows batch file (or shell file) to test xNAL sample document instance (using default genericcode based code lists) using two pass validation. Users should define appropriate relative paths in this file if they change the default directory structures • test-default-xpil.bat (.sh)– editable windows batch file (or shell file) to test xPIL sample document instance (using default genericcode based code lists) using two pass validation. Users should define appropriate relative paths in this file if they change the default directory structures • xAL-default.xml – <u>User modifiable</u> sample test

Directory Name	Contents
	<p>file for xAL default genericcode based code lists</p> <ul style="list-style-type: none"> • xPIL-default.xml - <u>User modifiable</u> sample test file for xPIL default genericcode based code lists • xNL-default.xml – User modifiable sample test file for xNL default genericcode based code lists • xNAL-default.xml – User modifiable sample test file for xNAL default genericcode based code lists
xsd/genericcode/xsd	<p>Contains the genericcode list based entity XML schemas for Name, Address and Party. Note: No <i>xNL-types.xsd</i>, <i>xAL-types.xsd</i>, <i>xNAL-types.xsd</i>, and <i>xPIL-types.xsd</i> exist in this directory as genericcode approach for code list is used.</p> <ul style="list-style-type: none"> • xNL.xsd – xNL schema for Name entity and is customised (extra attributes for genericcode based code list metadata information) from the default xNL.xsd. <u>Users must not modify</u> this file. • xAL.xsd – xAL schema for Address entity and is customised(extra attributes for genericcode based code list metadata information) from the default xAL.xsd. <u>Users must not modify</u> this file. • xNAL.xsd – xNAL schema for Name and Address entity and is customised (extra attributes for genericcode based code list metadata information) from the default xNAL.xsd. <u>Users must not modify</u> this file. • xPIL.xsd – xPIL schema for Party entity and is customised(extra attributes for genericcode based code list metadata information) from the default xPIL.xsd. <u>Users must not modify</u> this file. • CommonTypes.xsd – Schema reused by all the above entity schemas and is customised(extra attributes for genericcode based code list metadata information) from the default CommonTypes.xsd. <u>Users must not modify</u> this file. • xlink-2003-12-21.xsd – Same schema as the default version and <u>must not be modified by user</u>.
xsd/default/doc	<p>Provides HTML documentation for all genericcode based XML schemas (xNL, xAL, xNAL, xPIL, CommonTypes and xLink) in individual sub directories</p>
xsd/genericcode/utility	<p>This directory provides a set of utility files to prepare genericcode files such as XML parsers, creation of schematron patterns and XSLTs.</p> <ul style="list-style-type: none"> • prepare-cva.bat (.sh) – editable windows

Directory Name	Contents
	<p>batch file (or shell file) to prepare context/value associations, and <u>users are allowed to modify</u> it to include relative paths if the default directory structure is changed and to turn documentation generation feature on or off (default is off). Do not run this file on its own.</p> <ul style="list-style-type: none"> • prepare-gc.bat (.sh) – editable windows batch (or shell file) file to prepare genericodes, and <u>users are allowed to modify</u> it to include relative paths if the default directory structure is changed and to turn documentation generation feature on or off (default is off). Do not run this file. • twopass.bat (.sh)– editable windows batch file (or shell file) to perform two-pass structure/lexical validation and value validation, and <u>users are allowed to modify</u> it to include relative paths if the default directory structure is changed. Do not run this file. • w3cschema.bat (.sh)– editable windows batch file (or shell file) that calls the appropriate java files to perform XML parsing. <u>Users are allowed to modify</u> this file to define the relative paths if the default directory structure is changed. Do not run this file. • xslt.bat (.sh) – editable windows batch file (or shell file) that calls the appropriate java file to create XSLT. <u>Users are allowed to modify</u> this file to define the relative paths if the default directory structure is changed. Do not run this file. <p>Other files not listed above – <u>do not touch them</u></p>
xsd/genericode/customised	This directory contains all customised genericode files (for demonstration purposes to show how default genericode files supplied in this package can be customised) along with files for preparing customised genericodes and test files. Users can modify the files under this structure to apply constraints on default genericode files in order to meet their specific requirements. (xsd/genericode/customised)
xsd/genericode/customised/gc-files	This sub-directory contains all the customised genericode files (for demonstration purposes) from default genericode files
xsd/genericode/customised/gc-files/common	Provides the directory to store genericode code list file that is customised for use by CommonTypes schema (CommonTypes.xsd). This directory is currently empty.
xsd/genericode/customised/gc-files/xAL	Provides the directory to store genericode code list files that is customised for use by Address XML schema (xAL.xsd). This directory has some sample

Directory Name	Contents
	test genericcode files to demonstrate customisation. <u>Users can modify</u> the sample genericcode files or add more genericcode files
xsd/genericcode/customised/gc-files/xNL	Provides the directory to store genericcode code list files that is customised for use by Name XML schema (xNL.xsd). This directory has some sample test genericcode files to demonstrate customisation. <u>Users can modify</u> the sample genericcode files or add more genericcode files
xsd/genericcode/customised/gc-files/xPIL	Provides the directory to store genericcode code list files that is customised for use by Party XML schema (xPIL.xsd). This directory has some sample test genericcode files to demonstrate organizations. <u>Users can modify</u> the sample genericcode files or add more genericcode files
xsd/genericcode/customised/gc-files/xPRL	Empty Directory
xsd/genericcode/customised/prepare	<p>This directory provides all files required to prepare the customised genericcode files.</p> <ul style="list-style-type: none"> • customised-cl-business-rules.sch – Defines the business rules to constraint the use of code lists using schematron language and is <u>modifiable by user</u>. A sample business rules for demonstration purpose is currently defined and <u>requires modification</u> to this file by the user to meet their specific requirements. Users should define appropriate relative paths in this file if they change the default directory structures • customised-cl-constraints.cva – list all of the genericcode expressions of agreed-upon customised value list value enumerations, and lists all of the customised contexts in which the value enumerations are used. This file is modifiable by user. Sample constraints for demonstration purposes are currently defined and <u>requires modifications</u> to this file by the user to meet their specific requirements. Users should define appropriate relative paths in this file if they change the default directory structures • CustomisedCodeList.sch – Defines the customised code list namespace constraints. This file is modifiable by user as they define constraints on default code lists. Users should define appropriate relative paths in this file if they change the default directory structures • prepare-customised-cl.bat (.sh)– editable windows batch file (or shell file) to prepare the customised genericcode files to constrain the default genericcode code lists for two pass validation. This is the “main” program that executes other batch programs. This file

Directory Name	Contents
	<p><u>requires modification</u> by the user as the contents in this file are for demonstration purposes only. Users should define appropriate relative paths in this file if they change the default directory structures</p> <p>Other files not listed above– All the other files in this directory are automatically generated when prepare-ciq.bat/sh file is executed and are to be <u>NOT modified by user and so , do not touch them</u></p>
xsd/genericcode/customised/test	<p>This directory provides files to test the customized genericcode lists (for demonstration purposes) by performing two pass validations. <u>Users can modify the .xml files to test different cases.</u> Sample test files have been provided for users to test them.</p> <ul style="list-style-type: none"> • test-all.bat (.sh)– editable windows batch file (or shell file) to test xNL, xAL, and xPIL sample document instance (using customised genericcode based code lists for demonstration purposes) using two pass validation. This file is <u>modifiable by user</u>. Users should define appropriate relative paths in this file if they change the default directory structures • test-customised-xnl.bat (.sh)– editable windows batch file (or shell file) to test xNL sample document instance (using customised genericcode based code lists for demonstration purposes) using two pass validation. This file is <u>modifiable by user</u>. Users should define appropriate relative paths in this file if they change the default directory structures • test-customised-xal.bat (.sh)– editable windows batch file (or shell file) to test xAL sample document instance (using default genericcode based code lists for demonstration purposes) using two pass validation. This file is <u>modifiable by user</u>. Users should define appropriate relative paths in this file if they change the default directory structures • test-customised-xpil.bat (.sh)– editable windows batch file (or shell file) to test xPIL sample document instance (using default genericcode based code lists) using two pass validation. This file is <u>modifiable by user</u>. Users should define appropriate relative paths in this file if they change the default directory structures • xAL-customised.xml – <u>User modifiable</u> sample test file for xAL customised (for demonstration purposes) genericcode based code lists • xNL-customised.xml – <u>User modifiable</u>

Directory Name	Contents
	<p>sample test file for xNL customised (for demonstration purposes) genericcode based code lists</p> <ul style="list-style-type: none"> • xPIL-customised.xml – <u>User modifiable</u> sample test file for xPIL customised (for demonstration purposes) genericcode based code lists • test-metadata.bat (.sh)- editable windows batch file (or shell file) to test sample metadata attributes (using customised genericcode based code lists for demonstration purposes) using two pass validation. This file is <u>modifiable by user</u>. Users should define appropriate relative paths in this file if they change the default directory structures

1.6 Verifying Test Results by testing the customised test files of code lists provided

By running the “test-all.bat” or “test-all.sh” in the “genericcode\customised\test” directory, the following result should occur.

```
C:\ciq-v3\ciq\v3.0\xsd\genericcode\default\test>test-all
:: ***** Testing xml documents that use default CIQ Genericcode based Code Lists *****
:: ***** Testing xNL-default.xml document *****
:: Validating xNL-default.xml with ..\..\xsd\xNL.xsd and ..\prepare\DefaultCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
Bad name component 'NativePlaceName' for a Person.: /n:PersonName/n:NameElement[7]/@n:ElementType
Processing terminated by xsl:message at line 138
:: ***** Testing xAL-default.xml document *****
:: Validating xAL-default.xml with ..\..\xsd\xAL.xsd and ..\prepare\DefaultCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
Bad type 'Tower' for Premises.: /a:Address/a:Premises/@a:Type
Processing terminated by xsl:message at line 138
:: ***** Testing xNAL-default.xml document *****
:: Validating xNAL-default.xml with ..\..\xsd\xNAL.xsd and ..\prepare\DefaultCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
:: No value validation errors
:: ***** Testing xPIL-default.xml document *****
:: Validating xPIL-default.xml with ..\..\xsd\xPIL.xsd and ..\prepare\DefaultCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
Bad type 'UnparsedData' for contact number data.: /p:Party/p:ContactNumbers/p:ContactNumber/p:ContactNumberElement/@p:Type
Processing terminated by xsl:message at line 138
C:\ciq-v3\ciq\v3.0\xsd\genericcode\default\test>_
```


1.7 Verifying Test Results by testing the customised test files of code lists provided

By running the “test-all.bat” or “test-all.sh” in the “genericcode\customised\test” directory, the following result should occur.

```
C:\ciq-v3\ciq\v3.0\xsd\genericcode\customised\test>test-all

:: ***** Testing xml documents that use customised CIQ Genericcode based Code Lists *****

:: ***** Testing xNL-customised.xml document *****

:: Validating xNL-customised.xml with ..\..\xsd\xNL.xsd and ..\prepare\CustomisedCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
Must have exactly one FirstName component: /n:PersonName
Must have exactly one LastName component: /n:PersonName
Bad type 'Title' for a Person name data.: /n:PersonName/n:NameElement/@n:ElementType
Bad type 'MiddleName' for a Person name data.: /n:PersonName/n:NameElement[4]/@n:ElementType

Processing terminated by xsl:message at line 139

:: ***** Testing xAL-customised.xml document *****

:: Validating xAL-customised.xml with ..\..\xsd\xAL.xsd and ..\prepare\CustomisedCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
Invalid element present. Only FreeTextAddress element allowed.
: /a:Address/a:AdministrativeArea
Invalid element present. Only FreeTextAddress element allowed.
: /a:Address/a:Locality
Invalid element present. Only FreeTextAddress element allowed.
: /a:Address/a:Thoroughfare
Invalid element present. Only FreeTextAddress element allowed.
: /a:Address/a:Premises
Invalid element present. Only FreeTextAddress element allowed.
: /a:Address/a:PostCode
Bad type 'Province' for Administrative area in address data.: /a:Address/a:AdministrativeArea/@a:Type
Bad type 'Area' for Locality in address data.: /a:Address/a:Locality/@a:Type

Processing terminated by xsl:message at line 139

:: ***** Testing xNAL-customised.xml document *****

:: Validating xNAL-customised.xml with ..\..\xsd\xNAL.xsd and ..\prepare\CustomisedCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
Invalid element present. Only FreeTextAddress element allowed.
: /na:Postallabel/a:Address/a:Locality
Invalid element present. Only FreeTextAddress element allowed.
: /na:Postallabel/a:Address/a:Thoroughfare
Invalid element present. Only FreeTextAddress element allowed.
: /na:Postallabel/a:Address/a:PostCode
Bad type 'CareOf' for dependency in party name and address data.: /na:Postallabel/na:Addressee/na:DependencyName/@na:Type

Processing terminated by xsl:message at line 139

:: ***** Testing xPIL-customised.xml document *****

:: Validating xPIL-customised.xml with ..\..\xsd\xPIL.xsd and ..\prepare\CustomisedCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
Bad type 'Title' for a Person name data.: /p:Party/p:PartyName/n:PersonName/n:NameElement/@n:ElementType
Bad type 'MiddleName' for a Person name data.: /p:Party/p:PartyName/n:PersonName/n:NameElement[3]/@n:ElementType
Bad type 'SKYPE' for electronic address identifier in party data.: /p:Party/p:ElectronicAddressIdentifiers/p:ElectronicAddressIdentifier[

Processing terminated by xsl:message at line 139

C:\ciq-v3\ciq\v3.0\xsd\genericcode\customised\test>
```


1.8 Verifying the test results of metadata attributes of the customised code lists

By running the “test-metadata.bat” or “test-metadata.sh” in the “genericcode\customised\test” directory, the following result should occur.

```
C:\ciq-v3\ciq\v3.0\xsd\genericcode\customised\test>test-metadata
:: Validating XML-meta-bad-1.xml with ..\..\xsd\xML.xsd and ..\prepare\CustomisedCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
Bad type 'NativePlaceName' for a Person name data.: /n:PersonName/n:NameElement[3]/@n:ElementType
Processing terminated by xsl:message at line 139

:: Validating XML-meta-good-1.xml with ..\..\xsd\xML.xsd and ..\prepare\CustomisedCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
:: No value validation errors

:: Validating XML-meta-good-2.xml with ..\..\xsd\xML.xsd and ..\prepare\CustomisedCodeList.xsl
:: Pass 1 XSD structure/lexical validation...
:: No structure/lexical validation errors.
:: Pass 2 XSL value validation...
:: No value validation errors

C:\ciq-v3\ciq\v3.0\xsd\genericcode\customised\test>_
```

2 Customising your Code Lists / Enumerations

In this section, we explain how to customise and execute the customised code lists using the two Options for code lists provided, to meet your specific requirements.

2.1 Using Option 1 for Code Lists (Default)

Modify enumeration lists in xNL-types.xsd, xAL-types.xsd, xPIL-types.xsd, CommonTypes.xsd as required to add/delete code list values. This is a pretty straight forward approach that requires no further work.

2.2 Using Option 2 for Code Lists (Genericcode approach)

This approach requires quite a bit of effort to set it up.

2.2.1 Pre-requisites

Following skills are required to use this approach:

- Good knowledge and understanding of the OASIS Code List Representation scheme
- Good knowledge and understanding of the OASIS Code List “Context Value Association” Methodology
- Experience in creating/updating windows batch files/Unix/Linux shell files
- Knowledge of writing schematron patterns
- The default XML parsers used in this package are Java parsers and hence, the user environment should have Java runtime environment to run the programs.

2.2.2 XML Parsers

The XML and XSLT parser provided with this package under the “utility” directory (clvv/utility) are only sample parsers. Users can use any parsers (not necessarily written in Java) of their choice that can do this job. There is no restriction.

2.2.3 Known Bug

There is a known bug in the xjparse java program (in “utility” directory). For this program to run correctly, the “prepare-gc.bat/prepare-gc.sh” under “utility” directory needs to provide the absolute path of where this java program is located for the program to run. The “prepare-gc.bat” file lists the code below.

2.2.4 Steps to “Prepare and Test” the two pass validation when modifying the supplied default package to customise default genericcode

Following are the steps to prepare and test the files for validation using the code list value validation methodology if changes are done to the supplied default package:

1. Create the .gc files to restrict or add to the code lists in the default .gc files
2. Update the “prepare-customised-cl.bat/prepare-customised-cl.sh” file to include the organization .gc files for validation
3. Update the “customised-cl-constraints.cva” file to define appropriate constraint rules reflecting step 1
4. Update the “customised-cl-business-rules.sh” file to define any specific business rules (using Schematron language) to constrain the use of code lists

5. If the default directory structure provided by the CIQ Specification package is changed, ensure that the relative paths in the following files are updated accordingly:
 - .gc files (in genericcode/default/gc-files and genericcode/customised/gc-files directories)
 - default-cl-constraints.cva (in genericcode/default/prepare directory)
 - customised-cl-constraints.cva (in genericcode/customised/prepare directory)
 - DefaultCodeList.sh (in genericcode/default/prepare directory)
 - CustomisedCodeList.sh (in genericcode/customised/prepare directory)
 - customised-cl-business-rules.sh (in genericcode/customised/prepare directory)
 - prepare-gc.bat/prepare-gc.sh (in genericcode/utility directory)
 - prepare-cva.bat/prepare-cva.sh (in genericcode/utility directory)
 - xslt.bat/xslt.sh (in genericcode/utility directory)
 - prepare-sh.bat/prepare-sh.sh (in genericcode/utility directory)
 - twopass.bat/twopas.sh (in genericcode/utility directory)
6. Change the absolute path coded in “prepare-gc.bat/prepare-gc.sh” in genericcode/utility directory to the correct absolute path where this package is installed
7. Run “prepare-default-cl.bat/prepare-default-cl.sh”. The output should produce no errors.
8. Run “prepare-customised-cl.bat/prepare-customised-cl.sh” if the code lists are rganizati. The output should produce no errors.
9. Now test two pass validations by using the “test-default-xal.bat/test-default-xal.sh”, “test-default-xnl.bat/test-default-xnl.sh”, and “test-default-xpil.bat/test-default-xpil.sh”.
10. Play with the sample xml files used in the default testing to check whether two pass validation produces no errors.
11. To test the customised file, run “test-customised-xnl.bat/test-customised-xnl.sh”, “test-customised-xal.bat/test-customised-xal.sh”, and “test-customised-xpil.bat/test-customised-xpil.sh” files.
12. Use the sample xml files or create sample xml files to test the validation of values

2.2.4.1 Steps to test two pass validation when the default package is not modified

Following are the steps to test the supplied default files for validation using the code list value validation methodology if no changes are done to the supplied default package:

1. Change the absolute path coded in “prepare-gc.bat/prepare-gc.sh” to the correct absolute path where this package is installed
2. Now test two pass validations by using the “test-default-xal.bat/test-default-xal.sh”, “test-default-xnl.bat/test-default-xnl.sh”, and “test-default-xpil.bat/test-default-xpil.sh”.
3. Play with the sample xml files used in the default testing to check whether two pass validation produces no errors
4. To test the customised file (if default code lists were customised, first execute “prepare-customised-cl.bat/prepare-customised-cl.sh”. Then, run test-customised-xnl.bat/test-customised-xnl.sh, test-customised-xal.bat/test-customised-xal.sh, and test-customised-xpil.bat/test-customised-xpil.sh files. To test all these files, run “test-all.bat” or “test-all.sh” file. To test the metadata attributes, run “test-metadata.bat” or “test-metadata.sh” file
5. Use the sample xml files or create sample xml files to test the validation of values

A. Acknowledgements

The following individuals have participated in the creation of version 3.0 of CIQ specifications and are gratefully acknowledged:

Participants:

Colin Wallis	New Zealand Government	Voting Member, CIQ TC
David Webber	Individual	Voting Member, CIQ TC
Fulton Wilcox	Colts Neck Solutions LLC	Voting Member, CIQ TC
Graham Lobsey	Individual	Voting Member, CIQ TC
Joe Lubenow	Individual	Voting Member, CIQ TC
John Glaubit	Vertex, Inc	Voting Member, CIQ TC
Michael Roytman	Vertex, Inc	Voting Member, CIQ TC
Ram Kumar	Individual	Chair and Voting Member, CIQ TC
George Farkas	XBI Software, Inc	Former Member, CIQ TC
Hidajet Hasimbegovic	Individual	Former Member, CIQ TC
John Putman	Individual	Former Member, CIQ TC
Mark Meadows	Microsoft Corporation	Former Member, CIQ TC
Max Voskob	Individual	Former Member, CIQ TC
Robert James	Individual	Former Member, CIQ TC

OASIS CIQ Technical Committee (TC) sincerely thanks the public (this includes other standard groups, organizations and end users) for their continuous feedback and support that helps the TC to work toward improving the CIQ specifications.

Special thanks to Mr.Ken Holman, Chair of OASIS Code List TC (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=codelist) for his support, guidance and generic code implementation assistance to the TC in releasing the OASIS Code List version of CIQ V3.0 XML Schemas.

Special thanks to Mr.Hugh Wallis, Director of Standards Development of extensible Business Reporting Language (xBR) International Standards Group (<http://www.xbrl.org>) for working closely with the CIQ TC in jointly implementing W3C xLink specification that is now used by both xBR and CIQ Specifications to enable interoperability between the two specifications.

Special thanks to Mr.Carl Reed, Chief Technology Officer of Open Geospatial Consortium (OGC – <http://www.opengeospatial.org>) for his guidance and assistance to the TC in referencing the work of OGC on GeoRSS and Geo-Coordinates for addresses/locations as part of CIQ Address Specifications.

OASIS CIQ TC also acknowledges the contributions from other former members of the TC since its inception in 2000.

B. Intellectual Property Rights, Patents, Licenses and Royalties

CIQ TC Specifications (includes documents, schemas and examples¹ and ²) are free of any Intellectual Property Rights, Patents, Licenses or Royalties. Public is free to download and implement the specifications free of charge.

¹**xAL-AustralianAddresses.xml**

Address examples come from AS/NZ 4819:2003 standard of Standards Australia and are subject to copyright

²**xAL-InternationalAddresses.xml**

Address examples come from a variety of sources including Universal Postal Union (UPU) website and the UPU address examples are subject to copyright.

xLink-2003-12-31.xsd

This schema was provided by the xBRL group in December 2006.

C. Revision History

Revision	Date	Editor	Changes Made
V3.0 PRD 01	13 April 2006	Ram Kumar and Max Voskob	Prepared 60 days public review draft from Committee Draft 01
V3.0 PRD 02	15 June 2007	Ram Kumar	Prepared second round of 60 days public review draft from Committee Draft 02 by including all public review comments from PRD 01. Also included is implementation of OASIS Code list specification
V3.0 PRD 02 R1	18 September 2007	Ram Kumar	Inclusion of comments from Public Review 02
V3.0	15 November 2007	Ram Kumar	Final Version
V3.0 PRD 03	08 April 2008	Ram Kumar	Revised to include V3.0 Change Logs
V3.0 CS02	20 September 2008	Ram Kumar	Final Version