



Guideline



Project Acronym: PEPPOL Grant Agreement number: 224974

Project Title: Pan-European Public Procurement Online



PreAward - eCat ICT - Models

PEPPOL Guidelines PEPPOL Test and Validation Guidelines for eCatalogue



Version: 1.1

Status: Work in progress



Editors:

Giancarlo De Stefano (Consip) Leonardo Bertini (Consip) Hugo Oliveira (ANCP) Andrea Leutgeb (BBG)





Project co-funded by the European Commission within the ICT Policy Support Programme				
	Dissemination Level			
Р	Public	Х		
С	Confidential, only for members of the consortium and the Commission Services			





Revision History

Version	Date	Editor	Org	Description
0.0.1	01.05.2011	Hugo Oliveira	ANCP	First Version
1.1	29.08.2011	Leonardo Bertini	Consip	Approved Version

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Statement of copyright



This deliverable is released under the terms of the **Creative Commons Licence** accessed through the following link: http://creativecommons.org/licenses/by/3.0/.

In short, it is free to

Share — to copy, distribute and transmit the work

Remix — to adapt the work

Under the following conditions

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).





Contributors

Organisations

Consip, Italy, http://www.eng.consip.it/on-line/en/Home.html
BBG, Bundesbeschaffung GmbH Austria, www.bbg.gv.at
ANCP, Agencia Nacional de Compras Publicas, http://www.ancp.gov.pt/EN/ANCP/PAGES/HOME.ASPX

Persons

Giancarlo De Stefano (Consip) Leonardo Bertini (Consip) Andrea Leutgeb (BBG) Hugo Oliveira (ANCP)





Table of Contents

1	Intro	duction		6
	1.1	Cross-Cor	mmunity definition	6
2	Over	all PEPPOL	QA and Testing Landscape	7
	2.1		he PEPPOL WP3 test	
	2.2		A & testing methodology	
	2.3		erminology used within this document	
	2.4		ls	
	2.5	Description	n of Test levels	7
		2.5.1	Unit test	
		2.5.2	Integration test – phase 1	7
		2.5.3	Integration test – phase 2	7
		2.5. <i>4</i>	Systems test	
		2.5.5	Acceptance test	8
3	Test	Process an	nd Schedule	9
	3.1		esting Process	
	3.2		l Testing Scope	
		3.2.1	Functional Testing Scope WP3	
	3.3	Test environ	onment and organizational test preparation	12
		3.3.1	Preconditions for interoperability testing	12
		3.3.2	Organizational test preparation	12
		3.3.3	Functional test preparation	13
4	Test	tools		14
	4.1	Test Mana	agement Tool	14
		4.1.1	How to use the tool for Test case administration	14
	4.2	Validation	Tool	17
		4.2.1	Schematron files	17
		4.2.2	XSLT files	
		4.2.3	How to use validation artefacts	18
	4.3	Bug Repo	rting	26
	4.4	Templates	s and Reporting	26
5	Appe	endix – Abb	previationsErrore. Il se	egnalibro non è definito.





1 Introduction

1.1 Cross-Community definition

The scope of this document is to provide common understanding about testing in the PEPPOL WP3 eCatalogue pre-award. The target group of this document are all the members of PEPPOL WP3. This document aims only for the PEPPOL WP3 eCatalogue pre-award scope.





2 Overall PEPPOL QA and Testing Landscape

2.1 Scope of the PEPPOL WP3 test

The aim of the WP3 test is to find potential bugs and problematic issues within the functionalities before the pilot. Besides that, the testing – as all software testing activities – shall save time and money by detecting problems as soon as possible and therefore a crucial measure of QA within the general PEPPOL quality framework.

Therefore the scope of the test is:

- A measure of software quality assurance to make sure the defined specifications of WP3 eCatalogue pre-award.
- To do the PEPPOL project internal step of evaluation of the Pilot implementation within the scope of WP3 eCatalogue pre-award.

2.2 General QA & testing methodology

The general methodology and terminology is aligned with the document "CEN ISSS WS/BII Test Guidelines, Version 1.0", and with the chapter "testing guidelines" of the document "PEPPOL Development infrastructure and guidelines". General definitions are taken from these two documents.

2.3 Testing Terminology used within this document¹

2.4 Test Levels

The proposed test guidelines are based on the V-model. A V-model generally provides a test and quality assurance framework for the development of an IT solution. The V-model also describes the tests to be conducted, and in what order these should be executed.

2.5 Description of Test levels²

2.5.1 Unit test

Is done by technical staff to ensure, that the code works as defined; generally executed only for one component (e.g. national validation).

2.5.2 Integration test - phase 1

Is done to ensure, that the overall technical and functional design of one solution/solution package works as defined; generally executed for more than one component/function (e.g. national validation and individual backend processing).

2.5.3 Integration test – phase 2

Is done to ensure, that the solution/solution package works together with external interfaces and data exchange (e.g. exchange of documents with second individual solution).

2.5.4 **Systems test**

Is done to ensure, that the whole business process works together with external interfaces and data exchange (e.g. Catalogue workflow with transport infrastructure).

² Short version, adapted from the document CEN ISSS WS/BII Test Guidelines version 1.0 – details see there



7

¹ Taken from the document CEN ISSS WS/BII Test Guidelines, Version 1.0



2.5.5 Acceptance test

Is done to ensure, that users will accept the solution for their business purposes (e.g. suppliers).





3 Test Process and Schedule

3.1 Overall Testing Process

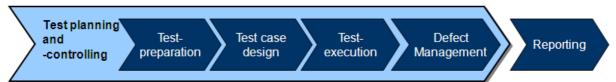


Figure 1: Holistic QA and testing process

Test planning

- Requirements analysis: Determine what aspects of a design are testable and with what parameters those tests work.
- Test concept creation.
- Test environment and infrastructure definition: e.g. Hardware, Software.
- Definition of to be tested functions or components by using of requirements analysis.
- Optional: definition of training for the testers: each tester must at least be familiar with the test process, the tools and the defect workflow.
- Definition of acceptance criteria.
- Organize the test team and establish a plan: Who, when, how, allocation of resources and budget...

Test preparation

- Kick-off workshop for test team.
- Organize and setup the test environment and infrastructure.
- Definition of test dataset and test scripts to use in testing software.

Test case design

Write, review and release the test cases.

Test execution and documentation

- Testers execute the defined test cases based on the plan and report any errors found to the development team.
- Defect Retesting: Once a defect has been solved by the development team, it is retested by the testing team.

Defect management

- Record the defects and monitor bug fixing.
- Defect Retesting: Once a defect has been solved by the development team, it is retested by the testing team.
- Regression testing: It is common to have a small test program built of a subset of tests, for each integration test phase of new, modified, or fixed software, in order to ensure that the latest delivery has not ruined anything, and that the software product as a whole is still working correctly.

Reporting on outcome per phase





Once the test meets the exit criteria, the activities such as capturing the key outputs, results, logs, documents related to the project are to be reported.

3.2 Functional Testing Scope

3.2.1 Functional Testing Scope WP3 eCatalogue pre-award

General:

Basis of the functional testing are the Pilot requirements such as agreed on within the WP3 project team – each of the requirements has to be tested.

#	Pilot Case Criteria	Mandatory					
1	Content and Validation of a catalogue template (pre-award)						
1.1	The Catalogue must be based on PEPPOL BIS12a specifications	У					
1.2	The CA can validate, that the Catalogue template fulfils all PEPPOL BIS12a requirements:						
	 Validate CORE Catalogue template as CEN/BII conformant 	У					
	 Validate CORE Catalogue template as CEN/BII and extensions conformant 	У					
	 Validate additional contractual content, as extensions 	n					
1.3	The EO can validate, that the Catalogue template fulfils all PEPPOL BIS12a requirements:						
	 Validate CORE Catalogue template as CEN/BII conformant 	У					
	 Validate CORE Catalogue template as CEN/BII and extensions conformant 	У					
	 Validate additional contractual content, as extensions 	n					

3.2.1.1 Functional testing dimensions WP3

Functional testing dimensions define a functional process or setting; it is useful to define these dimensions as "headlines" for different sets of test cases.

- Validation following the Template Validation
 - Technical structure (XSD)
 - Core data set (CEN BII) (Schematron)
 - ▶ BIS data set 12a (Schematron)
 - Optional customization: cross boarder/domestic/industry specific (Schematron)
 - Optional bilateral schemes (Schematron)
- Visualization (optional)
 - Display on screen
 - Display on print-out
 - Translation of texts (where applicable)
 - Verify all BII Core fields are present

3.2.1.2 Object-to-test dimensions WP3

Object-to-test dimensions define the type of quality of the data or process, which has to be tested.

- Data structure / involvement of partners -> applicable for all functionalities
 - dummy data / dummy partners
 - real-life simulation / demo client with simulated CA/EO
 - real-life data / pilot partner
- Validation (required)
 - Test each validation layer with valid and invalid files
- Document creation (optional)





- manual
- application generated

3.2.1.3 Test scenarios WP3

Test scenarios define overall processes, which allow an integration testing of different artefacts and interfaces along a real-life or real-life-simulated process chain.

Scenario 1 – Create Catalogue Template (CA)

Generate Catalogue template
 Expected result: Catalogue PEPPOL format

Validate PEPPOL Catalogue template

Expected result: validation report with a clear "valid" or "invalid"

Scenario 2 – Fulfil Catalogue (EO)

Fulfil Catalogue

Expected result: Catalogue PEPPOL format

Validate PEPPOL Catalogue

Expected result: validation report with a clear "valid" or "invalid"

3.2.1.4 Test Packages WP3

These are the test packages, each participant should perform based on the previously defined dimensions and scenarios:

- Generate Catalogue template
- Validation Catalogue template
- Perform full process (with a partner) as a sender (EO)
- Perform full process (with a partner) as a receiver (CA)

3.2.1.5 Validation testing for WP3

BII has drawn the validation architecture framework as a pyramid with multiple levels of validation. The base of the pyramid has the structural requirements for the document model while the top of the pyramid handles specific company requirements. This pyramid of validation can be mapped to the different levels of interoperability defined within the European Interoperability Framework:

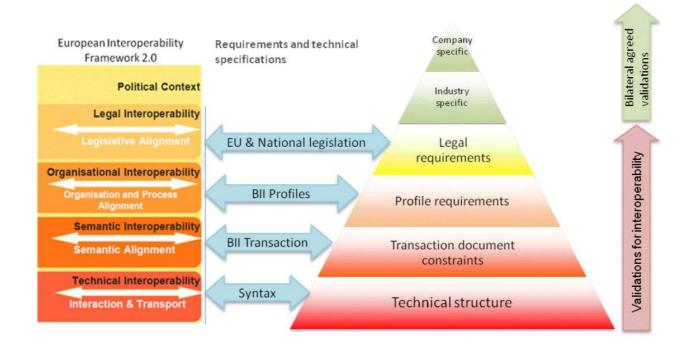






Figure 2 – Pyramid of validation

Where the technical validation artefacts provide the technical interoperability level, the semantic interoperability is provided by the CEN BII syntax neutral basic document constraints; the organizational interoperability can be validated using the profile requirements and finally, the legal interoperability is validated using legal requirement artefacts.

Different artefacts have been created for the validation of an XML instance in these different layers. The validation of a document instance is applied bottom-up in the pyramid, so the order of validation is:

BII BIS conformance

The first levels (technical, transaction and profile) are the basis for interoperability and define the compliance to the BII profile specifications. Every document exchanged must be valid according to these three levels of validation, and everyone in the exchange community must be able to process all received elements in a valid instance.

- 1. Validate that the technical structure of the document is correct.
- 2. Validate the **basic common** rules for the document to ensure the semantics of the instance are valid
- 3. Validate the process rules as the document instance is used within an specific profile

Processing a document instance does not mean being required to process it automatically. The information in a received instance can be partially processed electronically and partially viewed manually e.g. by use of style sheets. The validation levels in the profile conformance are:

The purpose of PEPPOL is to create at least one validation artefact for each validation layer, so users get technical material to ensure their implementations are conforming to the standards.

3.3 Test environment and organizational test preparation

3.3.1 Preconditions for interoperability testing

- software artefacts MUST be unit tested on an individual (e.g. national) level
- software SHOULD be integration tested with other national components (if necessary and if an end-to-end process has been chosen for specific test areas)

3.3.2 Organizational test preparation

The Interoperability testing is organized in test tracks. These are dedicated and scheduled time slots, where participants will execute a beforehand defined number of test cases referring to the test packages.

It is useful to follow this procedure for the planning and execution of each test track:

- 1. Choose the test packages you want to test
- 2. Decide for the BIS-number(s) to test
- 3. Choose the object-to-test-dimensions
- 4. Choose and/or design the test cases you want to execute
- 5. optional: Choose a partner you want to execute the test cases with
- 6. Be sure, that the resources (testing team, bug-fixing team, hardware and software resources, granted access to the systems you need,...) you need for testing are available during the planned test track
- 7. Execute the test cases
- 8. Document the test executions (test case tool, test report); report, fix and re-test bugs
- 9. Share lessons learned

The precondition for individual software tests is to have an up and running test environment with test datasets, supporting tools, equipment and technical support for the testers. This environment has to be established on time. On an individual (national) level the responsibility to provide the test environment





including a transparent and useful release management for updates lies within the implementation team of the components to be tested.

3.3.3 Functional test preparation

In order to provide quality assurance for all relevant functional requirements for the structuring of your overall testing scope the following main steps are suggested:

>> Step one - Create a Catalogue template

Verify requirements of the Catalogue Template.

Details:

The following details on each process step are provided by using the example of a Catalogue exchange.

Step one - Create a Catalogue template

- The objective is to test the creation and fulfil the Catalogue template.
 - 1. Create test Catalogue Template
 - 2. Test the Catalogue Template
 - 3. Validate Catalogue Template
 - 4. Fulfil the Catalogue Template
 - 5. Test the Catalogue Template after fulfil
 - 6. Validate the Catalogue Template after fulfil





4 Test tools

Several tools are provided to establish a testing process WP3:

- Overall WP3 Test plan
- Test Management Tool (web interface) containing:
 - Test cases (test case management)
 - Test execution (documentation of results on basis of test cases)
 - Shared documents (such us e.g. test guidelines, lessons learned documentation,...)
 - Link with defects on OSOR
- Validation Tool (web interface) containing:
 - UBL validation rules
 - CEN BII rules
 - PEPPOL rules (according to BIS specifications)
- Test report template:
 - Needs to be used to report the outcome of a each test phase (sum of all test executions for a certain test type) by participant
- OSOR forum:
 - Report bugs (defect management)
 - Share experiences on detailed technical questions (cross-WP)
- ▶ Skype/Adobe Connect meetings:
 - Share and discuss lessons learned

4.1 Test Management Tool

Test cases describe the steps to conduct the test execution including preconditions to verify the require condition.

There must be good cases (describing what has to happen) and bad cases (non-valid constellations) to be sure not only functional transactions, but also all validation rules and error handling is working correctly. A standardized test case template makes sure, that the relevant information will be contained in all test cases and makes test cases reusable (for different test phases, levels ...).

4.1.1 How to use the tool for Test case administration

The PEPPOL test tool is located at http://www.phloc.com/peppol/config It is provided free of charge and contains the central location where all test cases are stored. The following screenshots provide a quick look into the tool5.





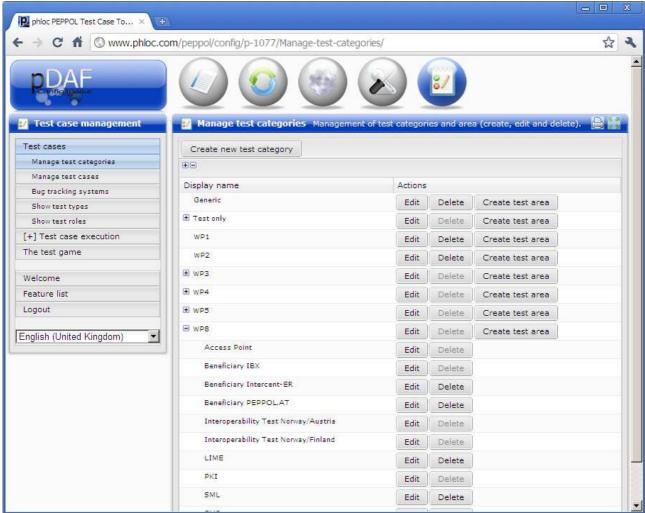


Figure 3: Manage test categories and test areas







Figure 4: Create a new test case





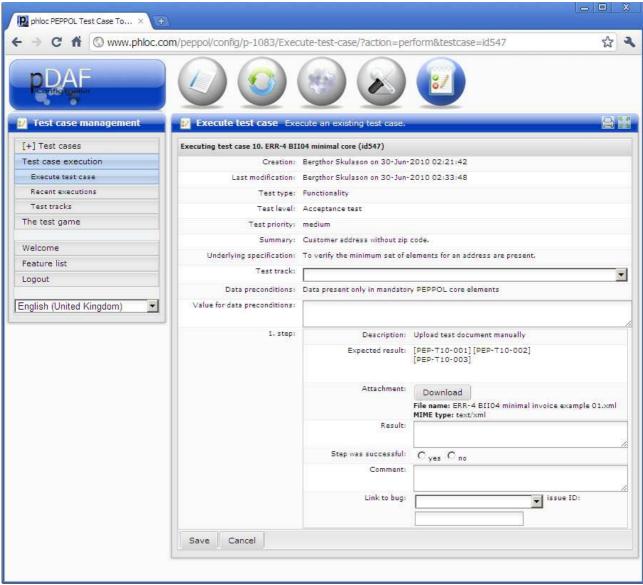


Figure 5: Execute a test case

4.2 Validation Tool

4.2.1 Schematron files

Technical tools produced in PEPPOL WP5 to validate XML Invoice and Credit Note instances are based on Schematron6. The Schematron validation artefacts are packaged using the following structure:

/ /abstract /SYNTAX1 /SYNTAX2 /codelist

Main folder

Root folder for the validation artefacts has the main Schematron file which imports the abstract, code list and syntax binding artefacts. It has to be invoked when trying to validate an XML instance. The name of the main Schematron file placed in the root validation directory follows the pattern:





<RULESET>-<SYNTAX>-<TRANSACTION>.SCH

Where:

RULESET - is the identifier for the rule set. PEPPOL has defined 4 main rule sets in the interoperability levels of the validation pyramid: BIICORE, BIIRULES, BIIPROFILES or EUGEN. Additionally, some other rule sets have been identified for countries such as Austria and Norway. They are ATNAT, NONAT, ATGOV and NOGOV.

SYNTAX - is the identifier for the XML syntax to which the business rules are bound.

TRANSACTION - is the identifier for the transaction (Txx) as defined in CEN BII.

Abstract folder

Abstract folder has the abstract Schematron artefacts. The abstract folder is not syntax specific, so the filenames for the Schematron files placed in that directory follow the pattern:

<RULESET>-<TRANSACTION>.sch

Schematron fragments in the abstract folder group all the assertions per each context in a single pattern.

Codelist folder

If the rule set has code list rules, there is a codelist folder that contains all codelists and associations from codelists to actual values in the XML instance in a single Schematron fragment.

The codelist Schematron fragment is generated from Genericode⁷ files and the associations defined in Context Value Association (CVA) files.

Syntax specific folder

For every syntax there is a folder with the actual mapping from the abstract rules to real XPATH expressions for that specific syntax.

PEPPOL has defined the binding to UBL so there is only a folder for UBL in the artefacts provided by PEPPOL.

4.2.2 XSLT files

There are several ways of implementing Schematron validators8. PEPPOL has used the validator implementation defined by Schematron to create XSLT files that parse and validate XML instances, producing an SVRL output.

SVRL stands for Schematron Validation Report Language and is a reporting language defined within ISO Schematron that produces an XML instance that identifies errors or warnings.

This is just a sample implementation of Schematron, but it can be implemented in many other ways.

XSLT validation files have the same pattern names as the original Schematron files but with the file extension ".xslt".

4.2.3 How to use validation artefacts

Validation artefacts can be used in different testing environments. They can be used from within XML processing tools or in different environments supporting XSLT or Schematron engines such as web browsers.





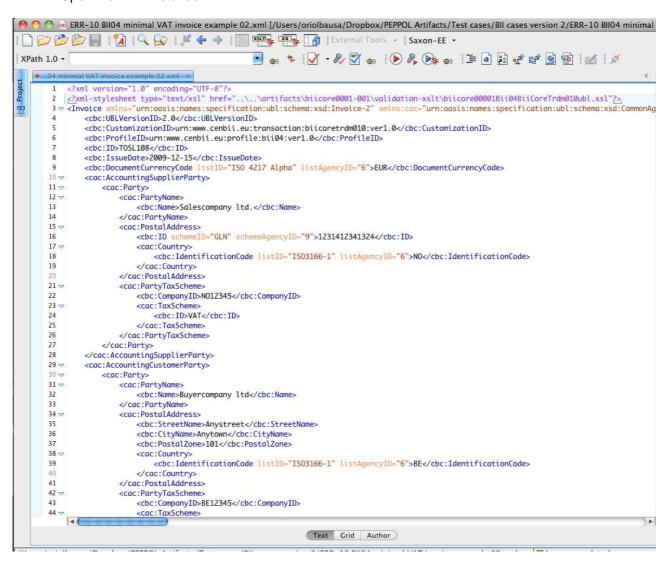
4.2.3.1 Using XML processing tools

XML processing tools such as XML Altova® or Oxygen® among others can be used to test XML instances. This kind of tools can process both Schematron and XSLT files.

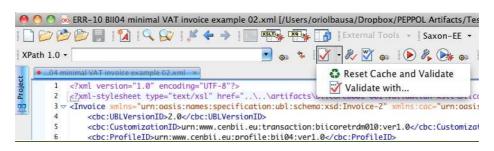
4.2.3.2 Test an instance using Schematron in Oxygen

Oxygen9 is a commercial XML IDE that has integrated Schematron support.

1. Open the XML instance



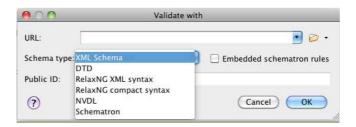
2. Select Validate with...







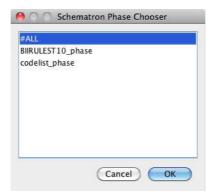
3. Select Schematron in the dropdown box



4. Select the Schematron file you want to use



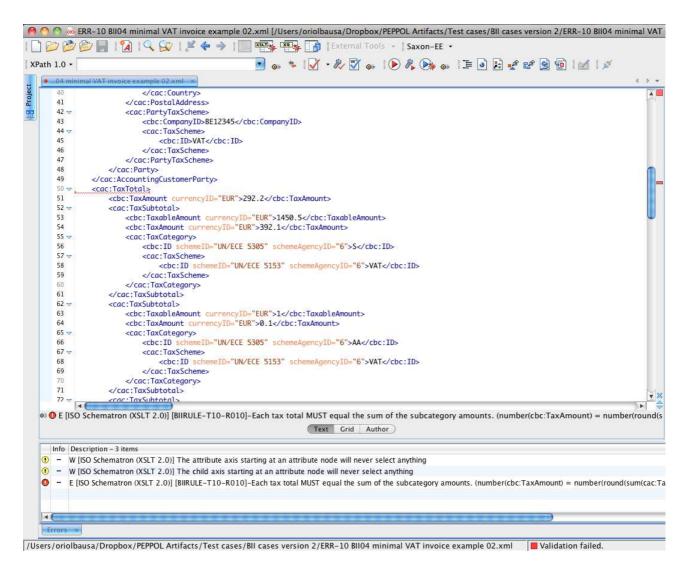
5. Select the Phase you want to run (usually there is only one phase)



6. Get the results





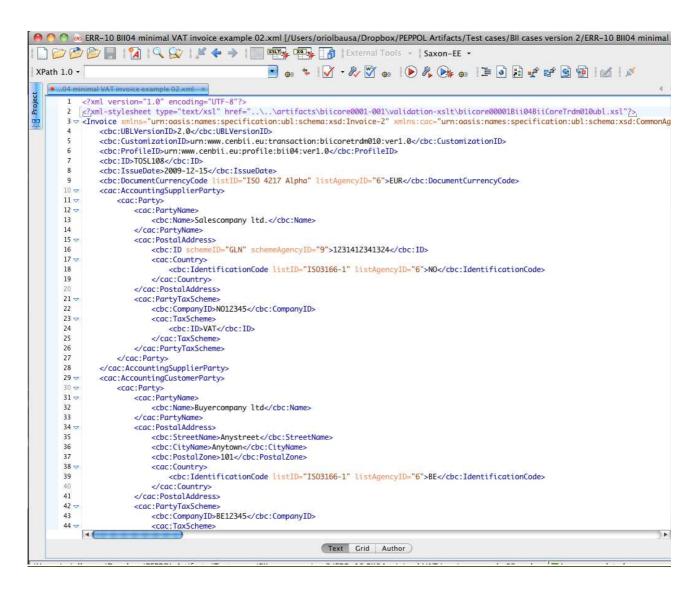


4.2.3.3 Test an instance using XSLT in Oxygen

1. Open the XML instance







2. Open the XSLT validation file



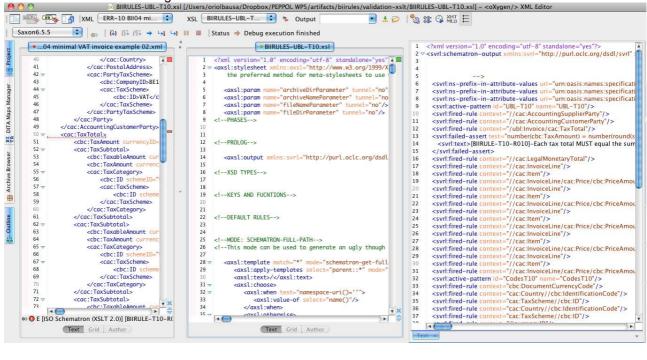


```
💩 BIIRULES-UBL-T10.xsl [/Users/oriolbausa/Dropbox/PEPPOL WP5/artifacts/biirules/validation-xslt/BIIRULES-UBL-T10.xsl] - <oXyge
□ 🙆 🙋 🖟 [ External Tools → Saxon-EE →
                                                    🗾 💩 🤏 📝 - 🎉 💟 🚳 🕒 👂 🍇 📑 🗿 🔡 🛫 😭 💁 🕍 💉
XPath 1.0 -
   ...04 minimal VAT invoice example 02.xml × BIIRULES-UBL-T10.
          the preferred method for meta-stylesheets to use where possible. -->
             <axsl:param name="archiveDirParameter" tunnel="no"/>
<axsl:param name="archiveNameParameter" tunnel="no"/>
<axsl:param name="fileNameParameter" tunnel="no"/>
<axsl:param name="fileDirParameter" tunnel="no"/>
          <!--PROLOG-->
      13
            <axsl:output xmlns:svrl="http://purl.oclc.org/dsdl/svrl" method="xml" omit-xml-declaration="no" standalone="yes" indent="yes"/>
          <!--XSD TYPES-->
      17
      19
          <!--KEYS AND FUCNTIONS-->
      21
          <!--DEFAULT RULES-->
      23
      24
          <!--MODE: SCHEMATRON-FULL-PATH-->
      26
          <!--This mode can be used to generate an ugly though full XPath for locators-->
             <axsl:template match="*" mode="schematron-get-full-path">
    <axsl:apply-templates select="parent::*" mode="schemat</pre>
      28 -
      29
                                                        mode="schematron-aet-full-path"/>
                <axsl:text>/</axsl:text>
      31 🗢
                <axs1:choose>
                   <axsl:when test="namespace-uri()=''">
      33
                      <axsl:value-of select="name()"/>
          -/avs] whan
                                                                 Text Grid Author
```

3. Go to XSLT mode pressing the XSLT button



4. Run the XSLT and get the SVRL instance





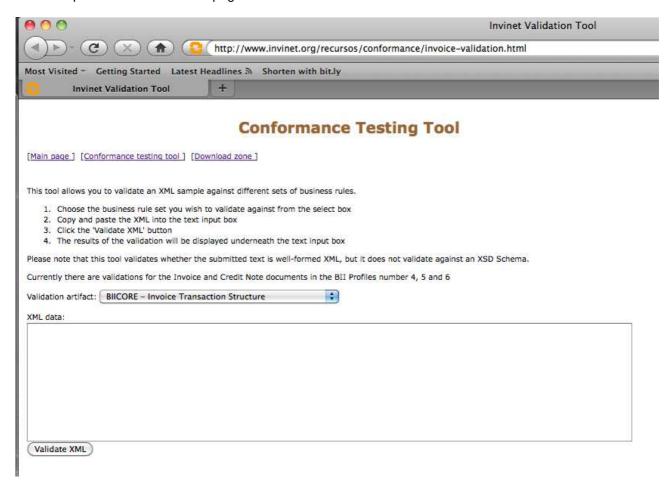


4.2.3.4 Using a web browser

Web Browsers include an XSLT engine. A web page can be set up to validate document instances using the built-in XSLT engine.

4.2.3.5 Test an instance using a web validation tool

1. Open web validation tool page



2. Copy & paste an XML invoice instance in the text area







Conformance Testing Tool

[Main page] [Conformance testing tool] [Download zone]

This tool allows you to validate an XML sample against different sets of business rules.

- 1. Choose the business rule set you wish to validate against from the select box
- 2. Copy and paste the XML into the text input box
- 3. Click the 'Validate XML' button

Validate XML

4. The results of the validation will be displayed underneath the text input box

Validation artifact: BIICORE - Invoice Transaction Structure

Please note that this tool validates whether the submitted text is well-formed XML, but it does not validate against an XSD Schema.

Currently there are validations for the Invoice and Credit Note documents in the BII Profiles number 4, 5 and 6

3. Select the validation artefact to run

Currently there are validations for the Invoice and Credit Note documents in the BII Profiles number 4,

Validation artifact:

BIICORE - Invoice Transaction Structure

BIICORE - Credit Note Transaction Structure

BIICORE - Corrective Invoice Transaction Structure

BIICORE - Credit Note Transaction Structure

BIIRULES - Invoice Business rules

BIIRULES - Credit Note Business rules

BIIRULES - Corrective Invoice Business rules

BIIRULES - Credit Note Profile rules

BIIROFILES - Credit Note Profile rules

BIIROFILES - Credit Note Profile rules

EUGEN - Invoice European General rules

EUGEN - Credit Note Surpean General rules

ATNAT - Invoice Austria National Rules

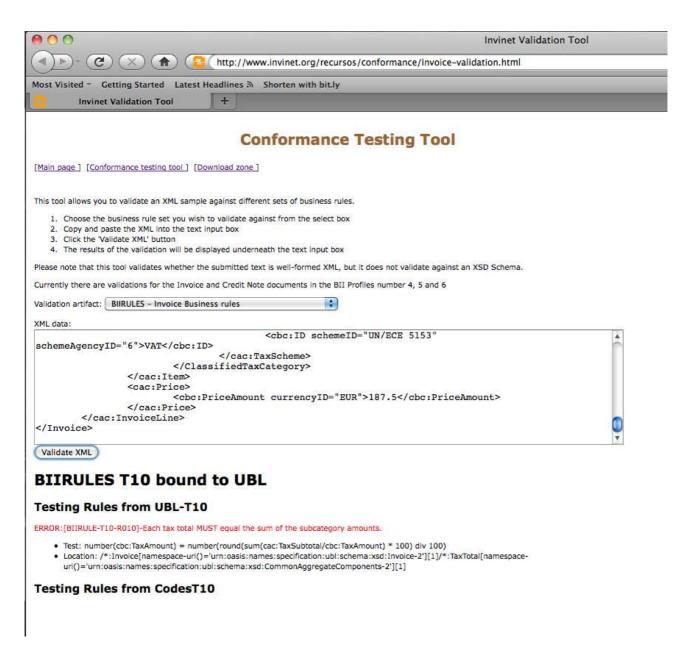
NONAT - Invoice Austria National Rules

NOGOV - Invoice Norway Government Business rules

4. Press validate button







4.3 Bug Reporting

We'll do it at OSOR: http://forge.osor.eu/tracker/?group_id=39

4.4 Templates and Reporting

- Central planning template: A central planning template (Excel) with all planned test tracks of all participants, who have scheduled testing activities
- Test reporting template A central planning an reporting template is provided in order to collect and align test plans of participants, quantitative and qualitative outcome of each test phase, and the informations to be provided to the EC for KSI reporting.
- Reporting of test tracks: A report of the executed test cases and results can be generated from the pDAf Tool by each test track (instead of generating it manually). So please do input your test cases in the tool and report the outcome of the execution you can save a lot of work!!

All templates are be published and updated for download within the pDAF Tool.

