



Doc Review Procedure

January 25, 2023



**Hewlett Packard
Enterprise**

Table of Contents

About this guide.....	1
1. Spell check	2
2. Terminology checker	4
2.1. Installation	4
2.2. Configuration.....	5
2.3. Usage	7
2.3.1. check a single file.....	7
2.3.2. check a ditamap and associated files.....	8
3. Schematron rules: installation and usage	10
3.1. check a single document	10
3.2. check a ditamap and associated files	11
4. Advanced automated review procedure	13

About this guide

- Chapter 1 provides instructions are running spell check on all of the files included in a ditamap.
- Chapter 2 includes details about using a customized terminology checker file (for the Oxygen Author XML terminology checker plugin) for automated reviews of tech documentation in Oxygen Author XML.
- Chapter 3 includes information about using Schematron rules in Oxygen Author XML.
- Chapter 4 includes a reference to a more advanced method of automated reviews.

Chapter 1. Spell check

Spell check the files in your ditamap!

- open **Check Spelling in Files** option in Oxygen XML Author to check multiple files or all of the files in a ditamap

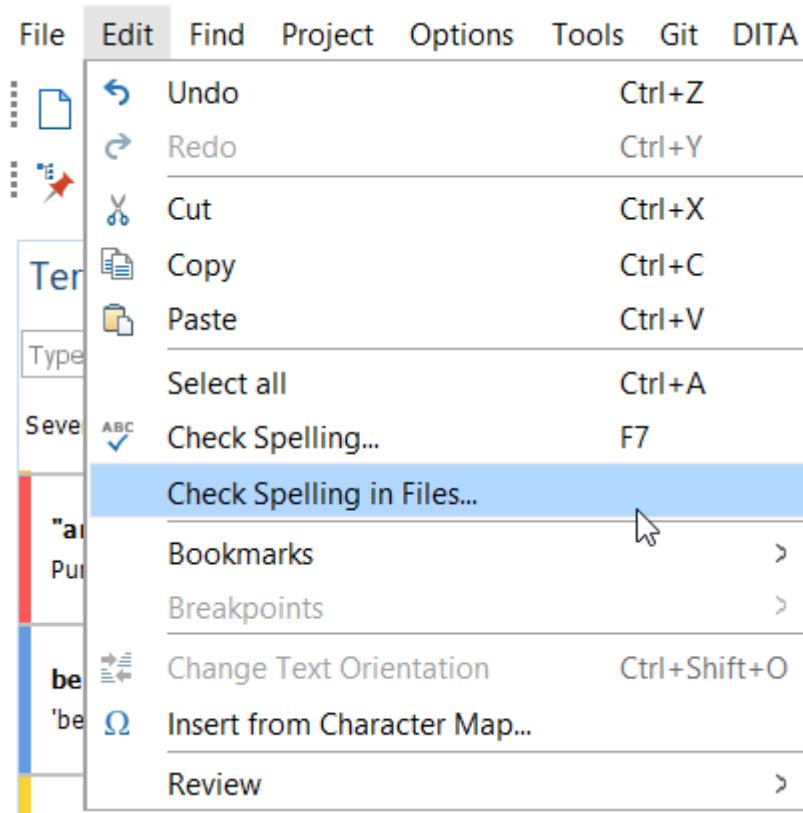


Figure 1. Spell check multiple files - Edit tab

- select spell check options - select Current DITA map hierarchy to check all of the files in a ditamap

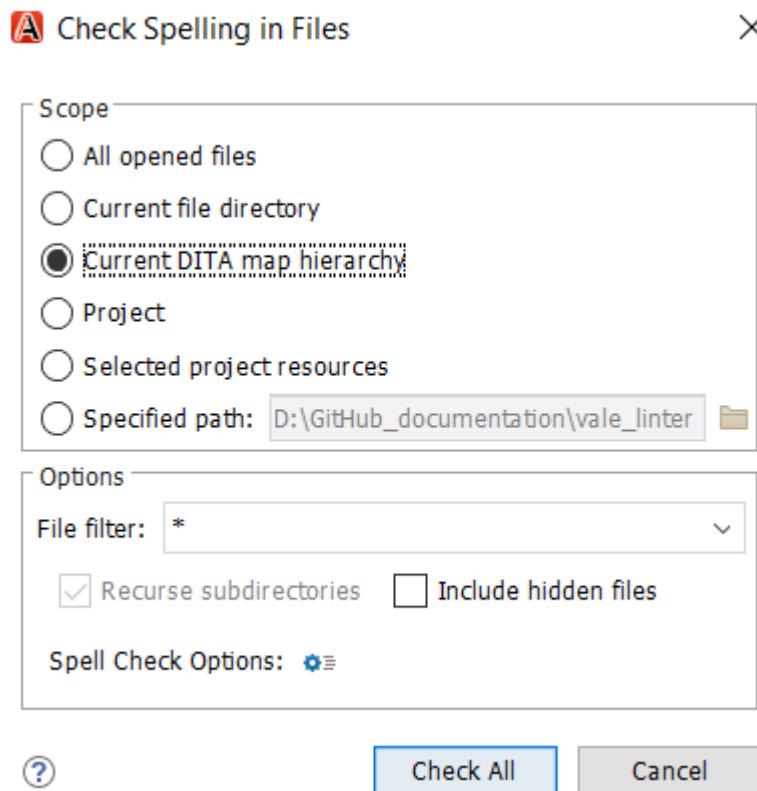


Figure 2. Spell check multiple files - options

- resolve resulting errors

Note ° clicking on an error will open the associated topic in the main panel

Results		
Description - 500 items	System ID	Location
Misspelled word: "Multi-level". Suggestions: "Mulch-level", "Multi-level", "Multilevel", "Multiplex", "Multiple", "Multitude", "Stim... D:\GitHub_documentation\slb\reuse\conventions_c.dita		42:95
↳ decomposed_deployment_c.dita (4 items)		
Misspelled word: "virtualization". Suggestions: "visualization", "actualization", "vitalization", "trivialization", "capitalization", "... D:\GitHub_documentation\slb\Release_Notes\decomposed_deployment_c.dita		5:84
Misspelled word: "MySQL". Suggestions: "Myself", "Mucky", "Muscle". D:\GitHub_documentation\slb\Release_Notes\decomposed_deployment_c.dita		29:19
Misspelled word: "TimesTen". Suggestions: "Times Ten", "Times-ten", "Timestamps", "Oftentimes", "Trimester", "Sentimen... D:\GitHub_documentation\slb\Release_Notes\decomposed_deployment_c.dita		29:43
Misspelled word: "preinstalled". Suggestions: "reinstalled", "pre installed", "uninstalled", "installed", "reinstall", "reinstate...", D:\GitHub_documentation\slb\Release_Notes\decomposed_deployment_c.dita		32:64
↳ deployment_architecture_c.dita (2 items)		
Check Spelling in Files X		

Figure 3. Spell check results

Chapter 2. Terminology checker

Note • Requires latest Oxygen XML Author!

2.1. Installation

1. Go to Help > Install new add-ons to open an add-on selection dialog box.

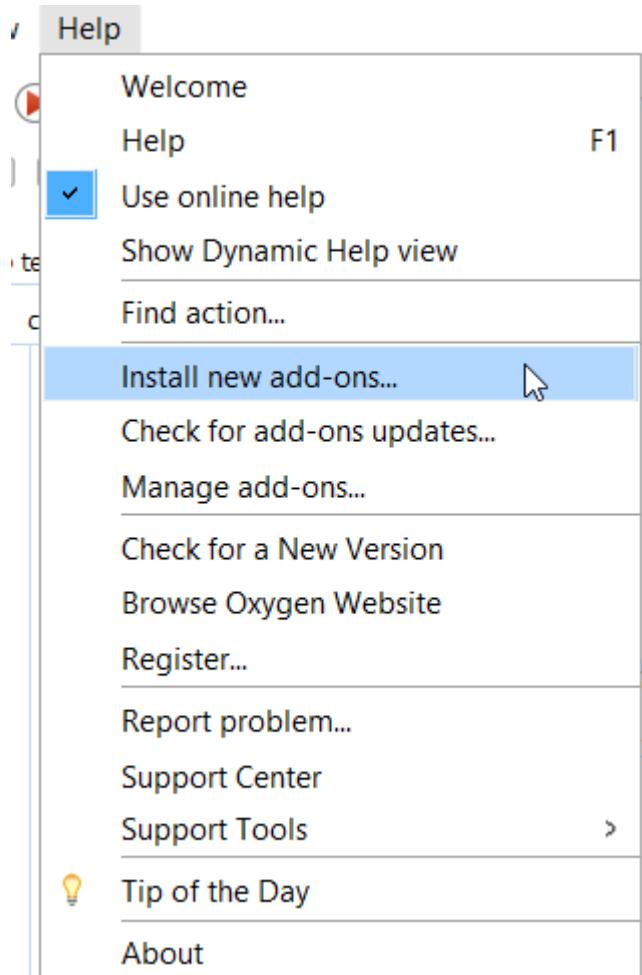


Figure 4. Install new add on

2. Enter or paste <https://www.oxygenxml.com/InstData/Addons/default/updateSite.xml> in the Show add-ons from field or select it from the drop-down menu
3. Select the Terminology Checker add-on and click Next

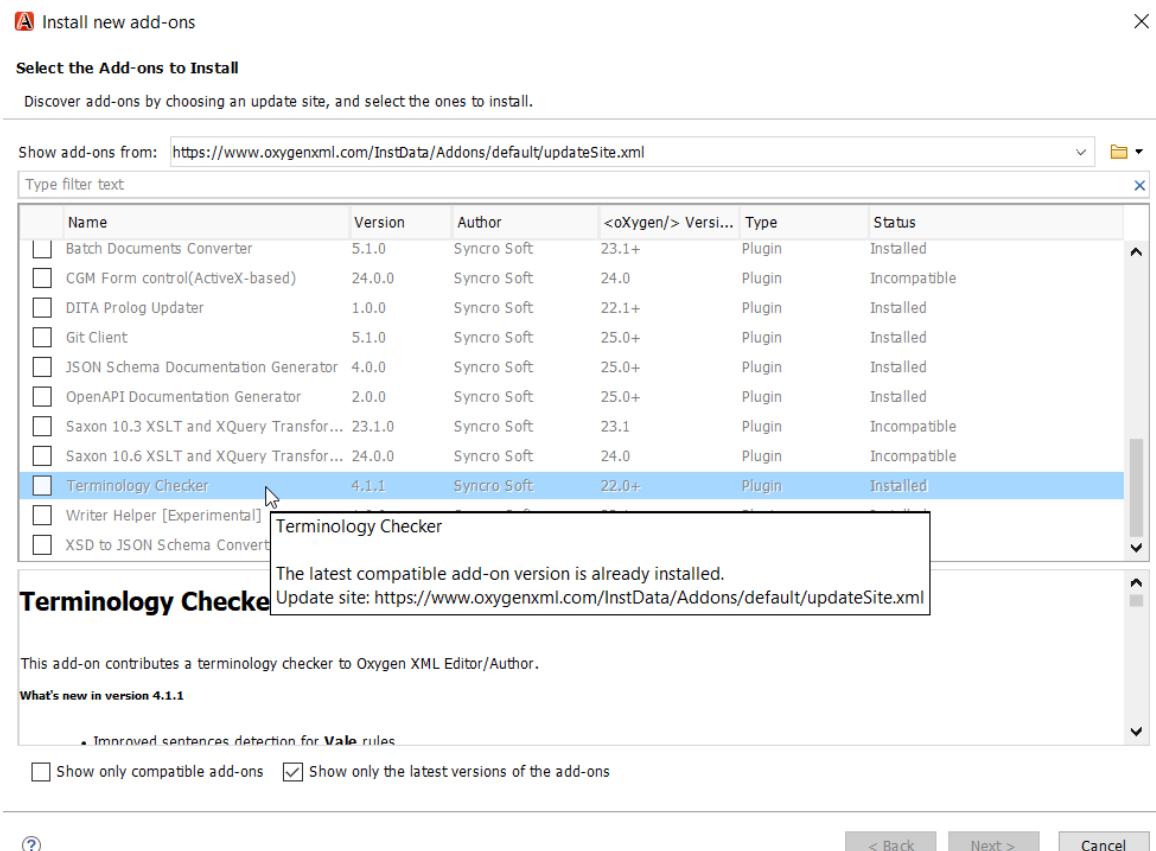


Figure 5. Install Terminology checker

4. Read the end-user license agreement and then select the I accept all terms of the end-user license agreement option and click Finish
5. Restart Oxygen XML Author

2.2. Configuration

1. Fork repo [review_automation](#) and clone it. Keep your forked repo in sync as updates may occur to the master repo.
2. Open in Oxygen XML Author the terminology file from the cloned GitHub repo
`review_automation\oxygen_xml_plugin_terminology_checker\terminology_checker.xml`.
3. Save the terminology_checker.xml file in a new subfolder named oxygen-term-checker located in the current project folder (the current project opened in Oxygen XML Author Project view)
example: c:\DITA\HPE_sources\snap\oxygen-term-checker\

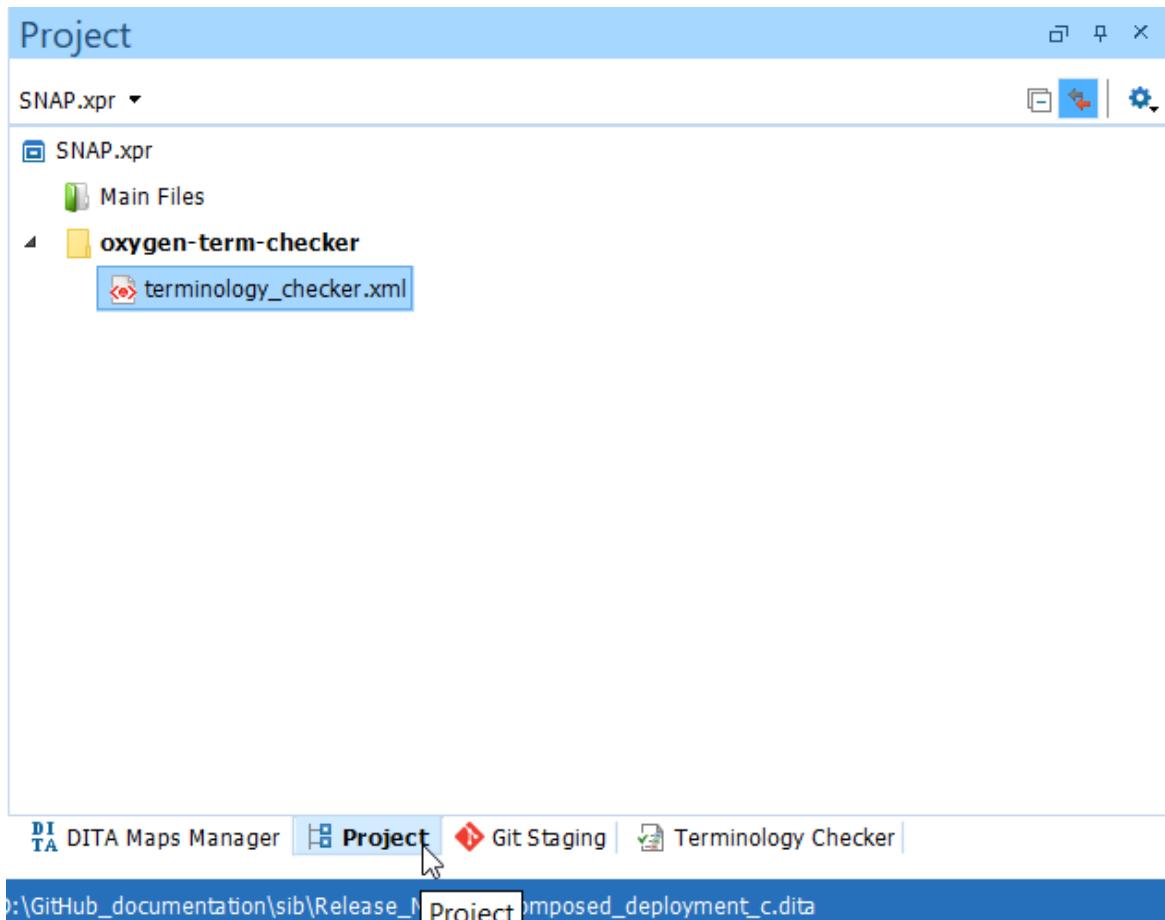


Figure 6. Project view

4. Go to the Options → Preferences → Plugins → Terminology Checker panel → Additional terminology folder and add the location to where the Vale style files are located -

\review_automation\oxygen_xml_plugin_terminology_checker\Vale_styles\

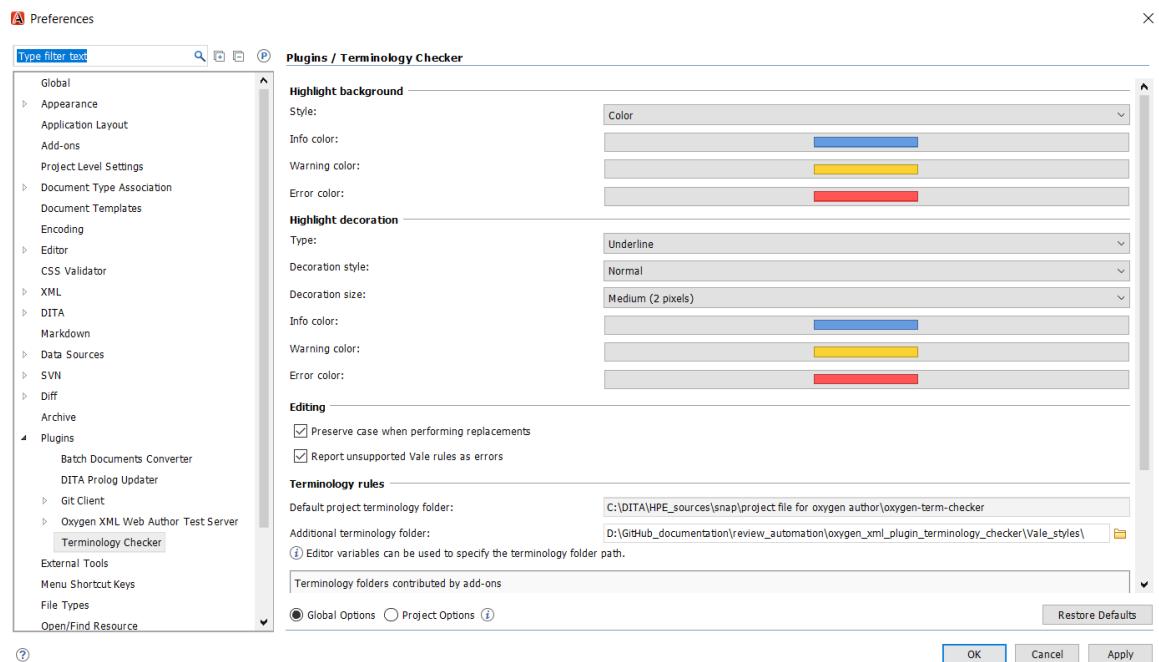


Figure 7. Additional terminology folder

5. add Terminology Checker window

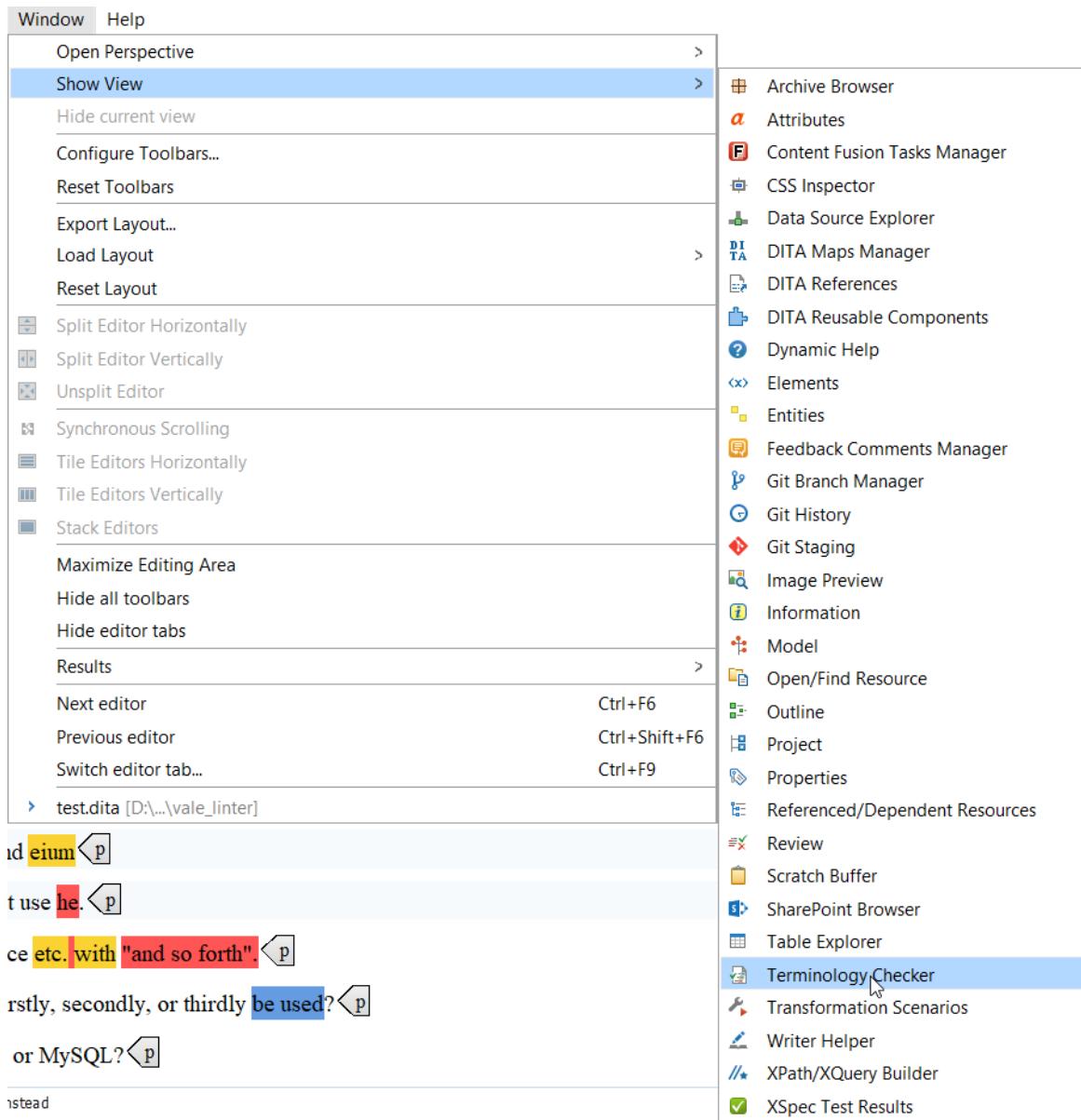


Figure 8. Terminology Checker window

2.3. Usage

2.3.1. check a single file

1. switch to the Terminology Checker window to see the warning and error messages which can also be highlighted by color in the file by configuring the proper preference options

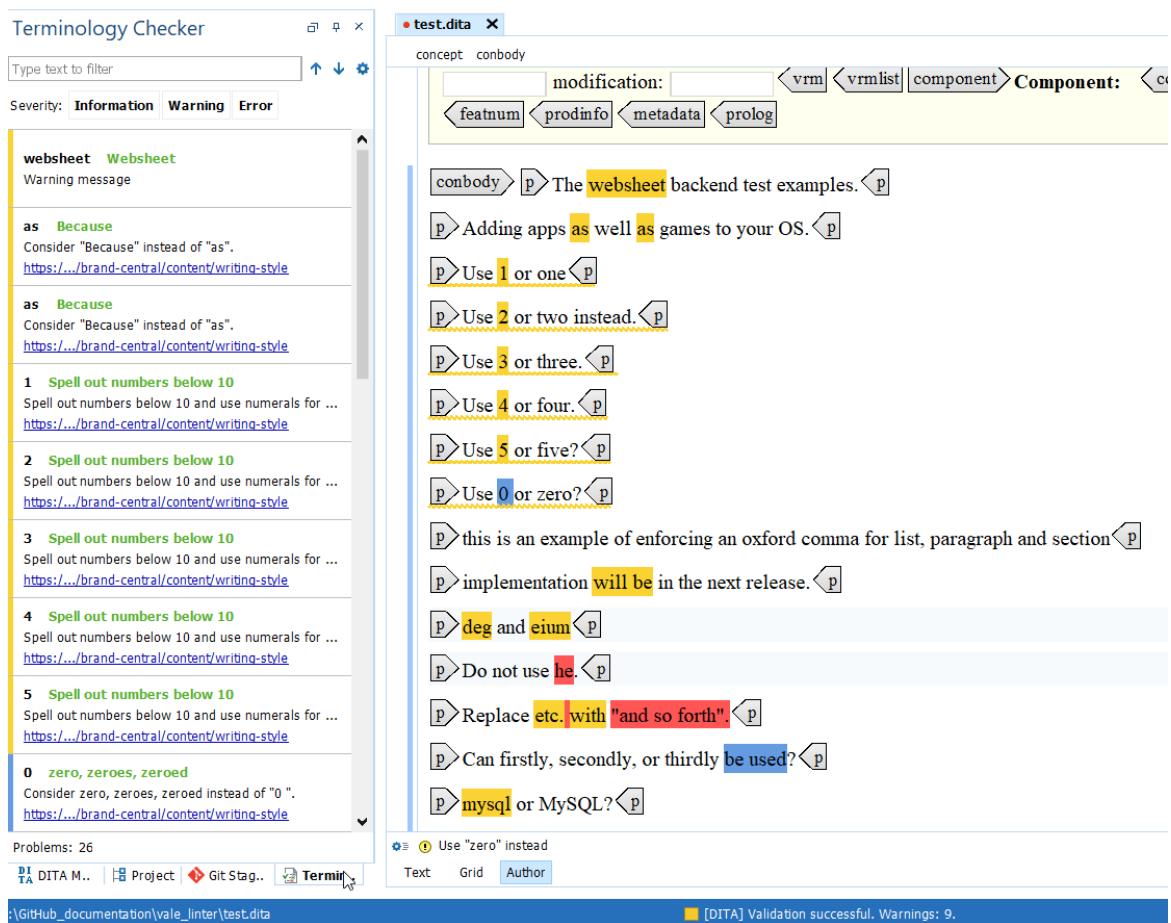


Figure 9. Terminology Checker example

2. resolve the highlighted errors

2.3.2. check a ditamap and associated files

1. to check all of the files in a ditamap, right click on ditamap and select Check terminology

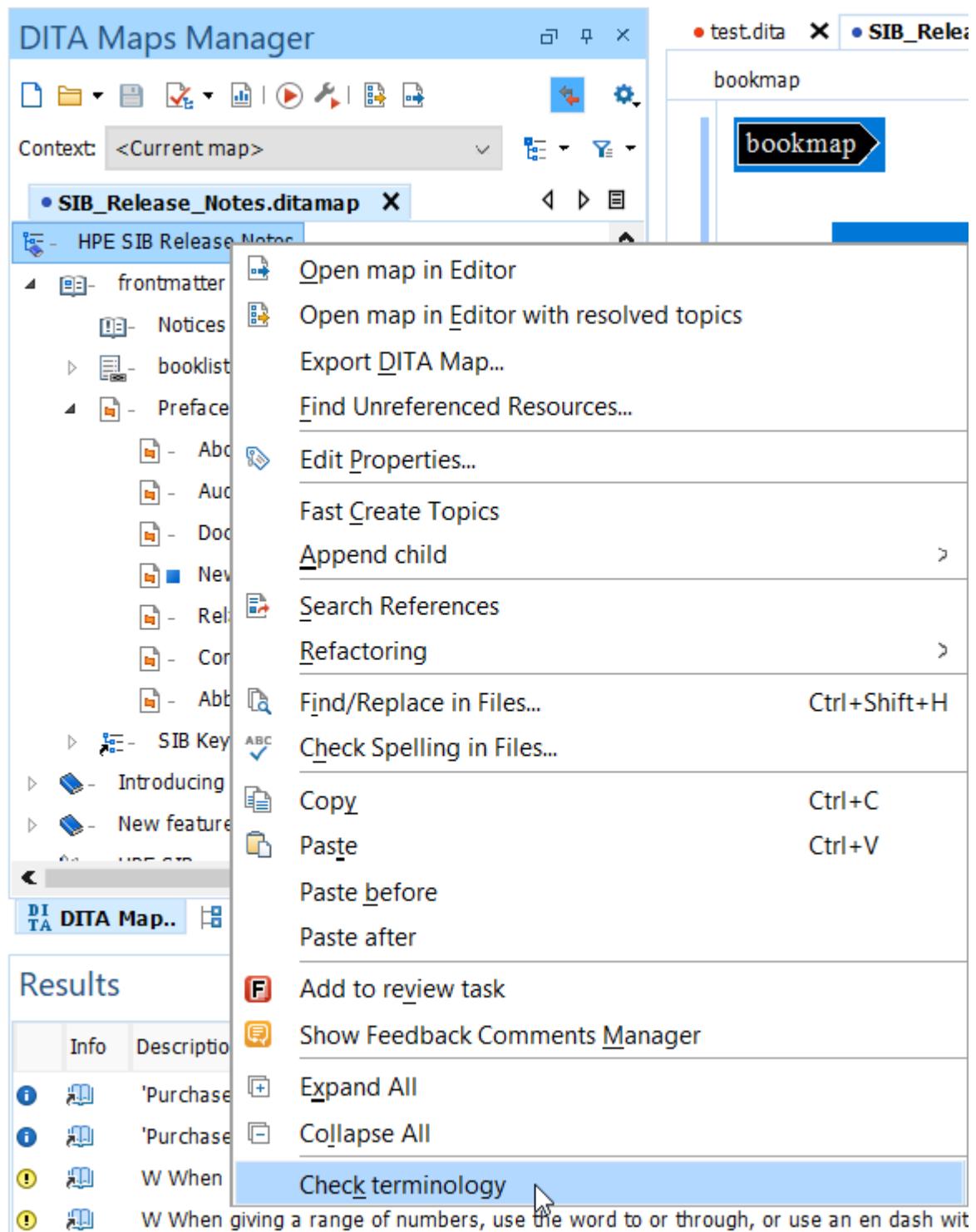


Figure 10. Terminology Checker ditamap example

2. resolve the highlighted errors noted in the files of the ditamap

Results			
Info	Description	Resource	System ID
!	'Purchase' looks like passive voice.	decomposed_deployment_c...	D:\GitHub_documentation\slb\Release_Notes\decomposed_deployment_c.dita
!	'EMS' has no definition.	decomposed_deployment_c...	D:\GitHub_documentation\slb\Release_Notes\decomposed_deployment_c.dita
!	'SBI' has no definition.	decomposed_deployment_c...	D:\GitHub_documentation\slb\Release_Notes\decomposed_deployment_c.dita
!	'EMS' has no definition.	decomposed_deployment_c...	D:\GitHub_documentation\slb\Release_Notes\decomposed_deployment_c.dita
W	When giving a range of numbers, use the word to or through, or use an en dash with a space before it.	cloud_native_deployment_f...	D:\GitHub_documentation\slb\overview_guide\cloud_native_deployment_for_all_hpe_sib_use_cases_c.dita
W	When giving a range of numbers, use the word to or through, or use an en dash with a space before it.	cloud_native_deployment_f...	D:\GitHub_documentation\slb\overview_guide\cloud_native_deployment_for_all_hpe_sib_use_cases_c.dita

Figure 11. Terminology Checker ditamap results example

Note ° clicking on an error will open the associated topic file in the main panel

Chapter 3. Schematron rules: installation and usage

3.1. check a single document

- configure toolbar by right clicking in the toolbar area and selecting Configure Toolbars...

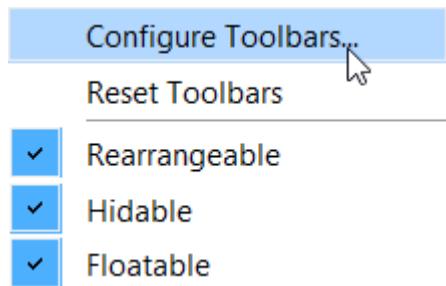


Figure 12. Configure toolbar

- select Document → Associate Schema

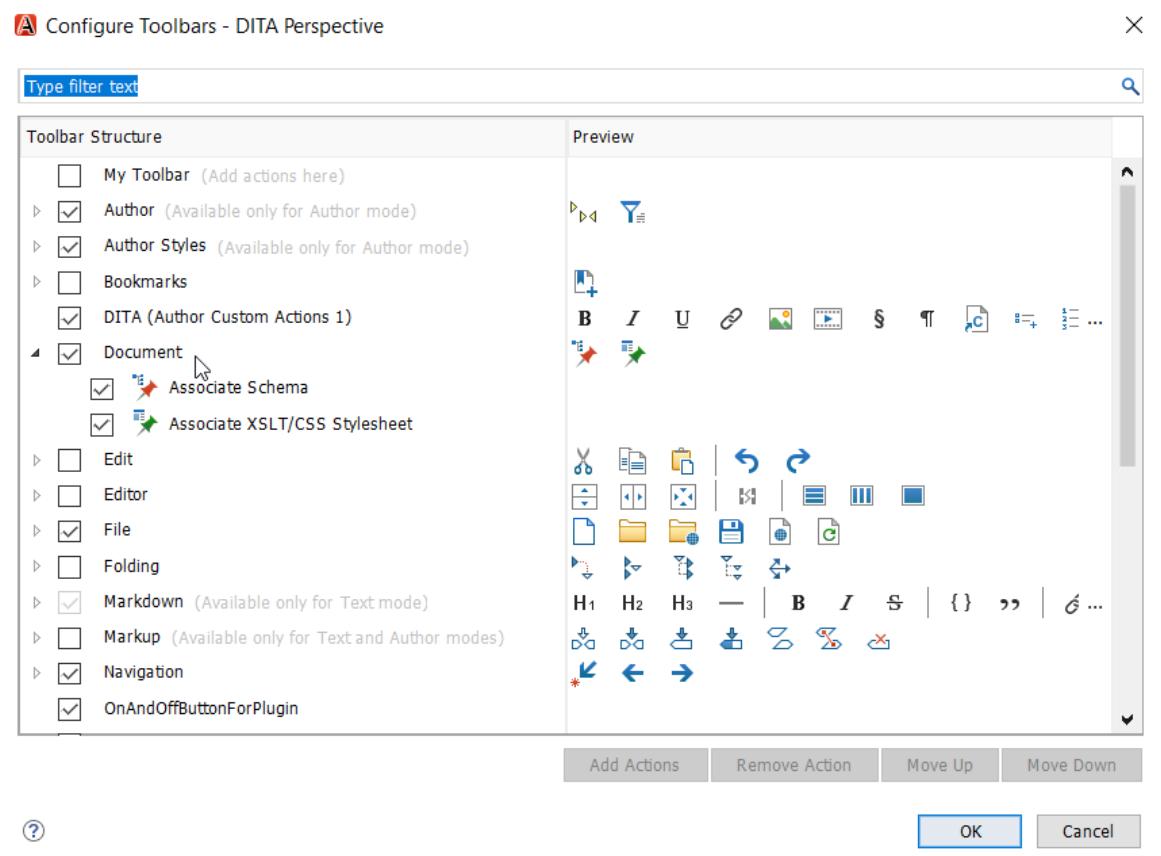


Figure 13. Add Associate Schema toolbar

- to run Schematron for that particular file click on the Associate Schema icon



Figure 14. Associate Schema button

- choose the master_schematron_rules.sch file for the URL path and click Ok

d:\DTC Enterprise\Global Team - Digital Identity\HPE MPC

Common\Schematron_rules\master_schematron_rules.sch

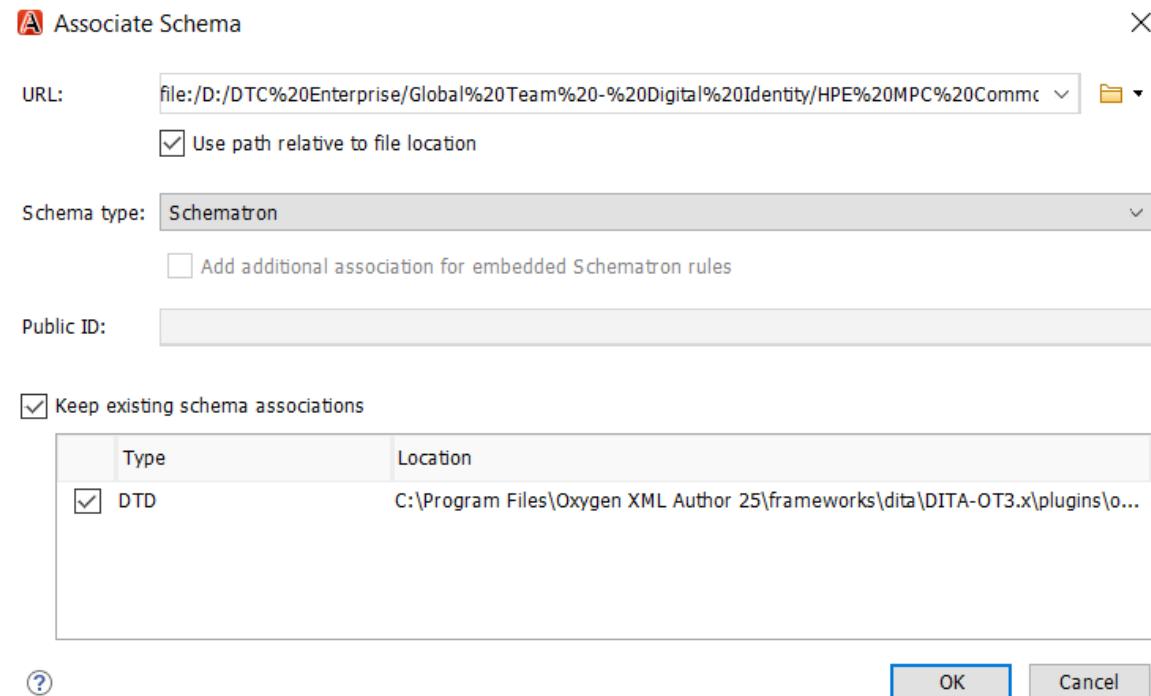


Figure 15. Associate Schema options

- resolve the resulting errors

Schematron example:

when Schematron is applied to a file then error messages are shown both on the right side of the main panel and at the bottom



Figure 16. Schematron messages

3.2. check a ditamap and associated files

- select Validate and Check for Completeness under DITA Maps Manager

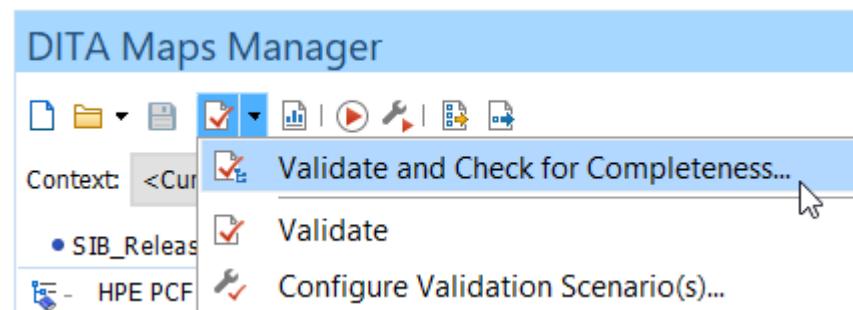


Figure 17. Validate and Check for Completeness

- add Schematron rules under Additional Schematron checks (\DTC Enterprise\Global Team - Digital Identity\HPE MPC Common\Schematron_rules\master_schematron_rules.sch) and click Check

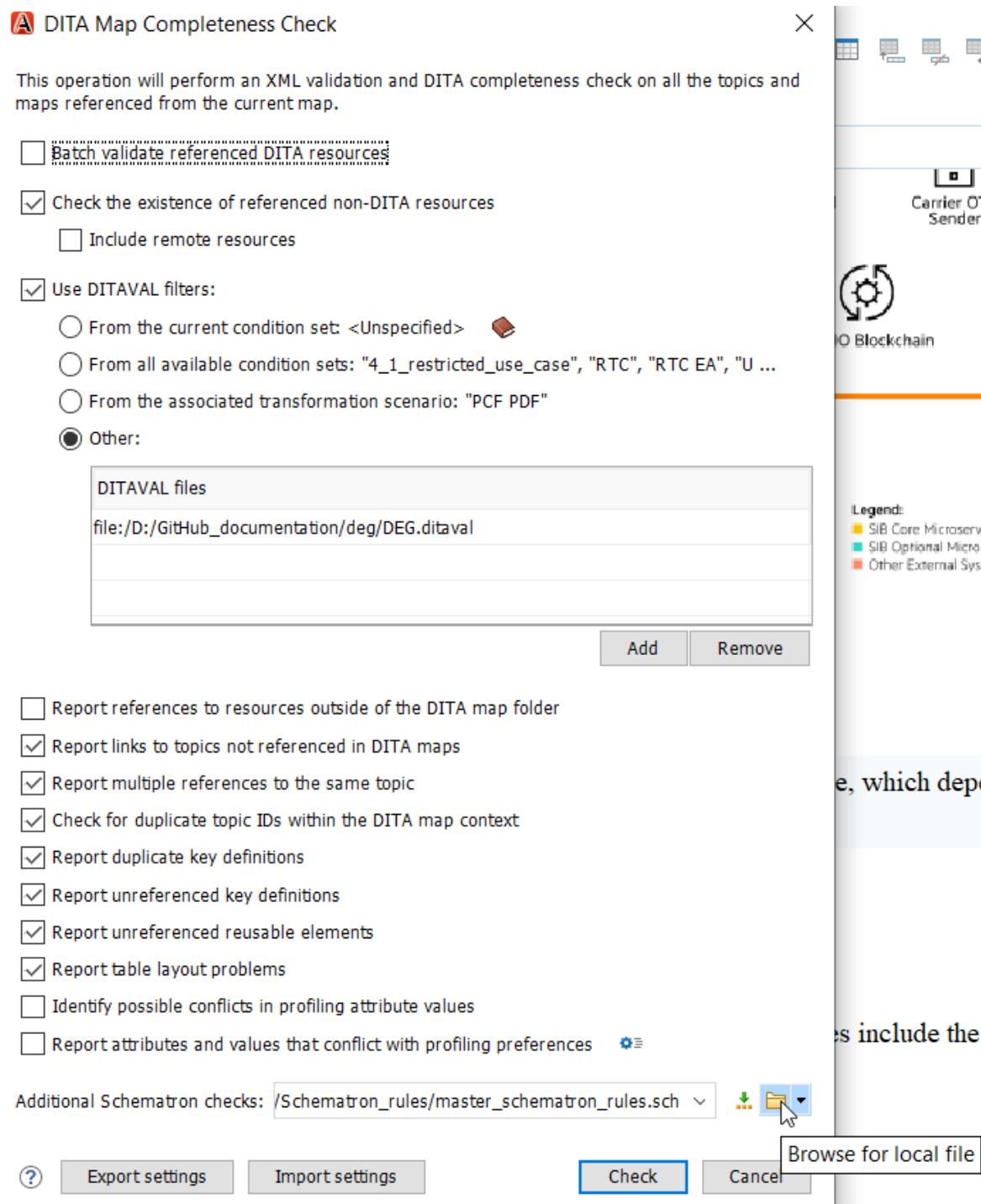


Figure 18. Validate and Check for Completeness options

- resolve the resulting errors

example error results

Results						
Info	Type	Description - 249 items	Resource	System ID	Location	
○	-	W [ISO Schematron] Do not put spaces as the first or last characters of uicontrol elements.	conventions_c_dita	D:\GitHub_documentation\sb1_reuse\conventions_c_dita	49:36	
○	-	W [ISO Schematron] Every table needs a title.	Reusable.dita	D:\GitHub_documentation\sb1_reuse\Reusable.dita	9:69	
○	-	W [ISO Schematron] If you only have one list item, you don't need a list.	Reusable.dita	D:\GitHub_documentation\sb1_reuse\Reusable.dita	853:100	
○	-	W [ISO Schematron] Do not put spaces as the first or last characters of b elements.	Reusable.dita	D:\GitHub_documentation\sb1_reuse\Reusable.dita	1231:49	
○	-	W [ISO Schematron] Do not put spaces as the first or last characters of keyword elements.	Reusable.dita	D:\GitHub_documentation\sb1_reuse\Reusable.dita	1234:20	
○	-	W [ISO Schematron] Do not put spaces as the first or last characters of keyword elements.	keywords_SIB.ditamap	D:\GitHub_documentation\sb1\keywords_SIB.ditamap	93:15	
○	-	W [ISO Schematron] Do not put spaces as the first or last characters of keyword elements.	keywords_SIB.ditamap	D:\GitHub_documentation\sb1\keywords_SIB.ditamap	632:15	

Figure 19. Validate and Check for Completeness results

- Note • clicking on an error will open the associated topic file in the main panel

Chapter 4. Advanced automated review procedure

This procedure uses the Vale linting tool and lefthook (a GitHub hook tool).

1. [Vale linter installation \(CLI\) and system configuration](#)
2. [Vale integration with other interfaces](#)
3. Running Vale on an individual file or folder: [Vale usage](#)
4. [Online tool for building Vale rules](#) — Use if you need to create or update *.yml files to check for additional conditions.
5. Review / Vale linter automation
 1. [Automating Vale linter checks with lefthook](#)
 2. [Vale usage with a continuous integration \(CI\) service](#)
 3. [Vale GitHub action](#). It has to be enabled by writing this email address - scm_ghe_admins@groups.ext.hpe.com - to ask the admins to enable it for your repo.
6. [Understanding the Vale command line tool output](#)
7. If Vale was installed on your machine using the choco command, then to update Vale to a new release run the command choco upgrade vale.
8. Vale rules in this repo were built to follow the [HPE Style Guide](#)
Rules used to convert the HPE Style Guide were based on YAML styles that can be found in these locations:
 - <https://github.com/topics/vale-linter-style>
 - <https://github.com/errata-ai/styles>
 Note * Some styles have been modified from their original version.
9. Extension points

Check	Implementations
existence	HPE: AM_PM.yml, Ampersands, Begin_sentence.yml, But.yml, Capitalize_file_extensions.yml, Dates_ordinal.yml, Dates.yml, Directional_language.yml, Disabilities.yml, EmDash.yml, EnDash.yml, Ensure.yml, Exclamation.yml, Heading_capitalization.yml, headingPunctuation.yml, HPE_a_an.yml, HPE_Ellipses.yml, HPE_hyphen.yml, Indefinite_modifiers.yml, Jargon.yml, MeaningfulLinkWords.yml, MeaningfulWords_colors_shapes.yml, Negative_numbers.yml, Numbers_ranges.yml, Numeric_dates, Passive.yml, Person.yml, Prepositions_end_sentence.yml, Prepositions.yml, Quote_punctuation.yml, Semicolons.yml, Slash.yml, Spaces.yml, Tense.yml, Time_ranges.yml, Units_space, Wordiness.yml Base: Capitalize_list_start.yml, Customer_names.yml, Empty_reference.yml, So.yml, Validate_wordiness.yml

Check	Implementations
substitution	HPE: 24_7.yml, above_below.yml, Advice_advise.yml, Appendixes, HPE_Avoid.yml, Backup.yml, British.yml, Contractions.yml, End_user.yml, Gender.yml, HPE_name.yml, HPE_possessive.yml, InclusionCultural.yml, InclusionOther.yml, Latin.yml, Logon_login.yml, Noon_midnight.yml, Numbers.yml, Please.yml, Positive.yml, Pull_down.yml, Redundant.yml, Setup.yml, Should.yml, Terms_ignorecase_false.yml, Terms_ignorecase_true.yml, Very.yml, Word_choice.yml, Zero.yml Base: Backend.yml, Term_Consistency_yml, Typography.yml
occurrence	Commas_morethanthree.yml, Sentence_length.yml
repetition	Repetition.yml
consistency	Consistency.yml
capitalization	Capitalization.yml
readability	N/A
conditional	HPE: Acronyms_abbreviations.yml Base: UnexpandedAcronyms.yml
spelling	Spelling.yml

For more details, see the [README file](#).