PWGSC Open Data Tool

User Guide

Version 6.14.0

November 23, 2022

# Contents

[PWGSC Open Data Tool 1](#_Toc122072624)

[User Guide 1](#_Toc122072625)

[Contents 2](#_Toc122072626)

[Introduction 5](#_Toc122072627)

[Tool Limitations 5](#_Toc122072628)

[Installation 5](#_Toc122072629)

[System Requirements (Windows) 5](#_Toc122072630)

[Installing the WPSS Tool 6](#_Toc122072631)

[Optional Requirements 6](#_Toc122072632)

[Uninstall 8](#_Toc122072633)

[Uninstalling Perl 8](#_Toc122072634)

[Uninstalling Python 8](#_Toc122072635)

[Using the WPSS Open Data Tool 9](#_Toc122072636)

[Open Data Tab 9](#_Toc122072637)

[Configuration Tab 11](#_Toc122072638)

[Profiles 13](#_Toc122072639)

[Results Window 14](#_Toc122072640)

[Stopping the Analysis 15](#_Toc122072641)

[Reporting Passes and Fails 15](#_Toc122072642)

[Saving Results 15](#_Toc122072643)

[Command Line Interface 16](#_Toc122072644)

[Status and Progress 16](#_Toc122072645)

[Language Switching 16](#_Toc122072646)

[Troubleshooting 18](#_Toc122072647)

[Perl Command Line Interpreter Error 18](#_Toc122072648)

[500 Internal Server Error 18](#_Toc122072649)

[Test Cases 20](#_Toc122072650)

[OD\_URL - Open data URL 20](#_Toc122072651)

[OD\_ENC - Character encoding 20](#_Toc122072652)

[OD\_REG – Open Data registry 21](#_Toc122072653)

[OD\_VAL – Validation 21](#_Toc122072654)

[OD\_DATA – Data Quality 22](#_Toc122072655)

[TBS\_QRS\_Structured 24](#_Toc122072656)

[TBS\_QRS\_Online 26](#_Toc122072657)

[TBS\_QRS\_Tidy 26](#_Toc122072658)

[TBS\_QRS\_Honest 27](#_Toc122072659)

[TBS\_QRS\_International 27](#_Toc122072660)

[TP\_PW\_OD\_BOM – UTF-8 BOM 27](#_Toc122072661)

[TP\_PW\_OD\_CONT – Content 28](#_Toc122072662)

[TP\_PW\_OD\_CONT\_CONSISTENCY – Content 29](#_Toc122072663)

[TP\_PW\_OD\_CONT\_DUP – Duplicate Content 29](#_Toc122072664)

[TP\_PW\_OD\_DATA – PWGSC Data Quality 29](#_Toc122072665)

[TP\_PW\_OD\_DD – Data Dictionary 31](#_Toc122072666)

[Checks by File Type 32](#_Toc122072667)

[CSV Files 32](#_Toc122072668)

[JSON-CSV Files (JSON encoded CSV files) 34](#_Toc122072669)

[JSON Dataset Description Files 36](#_Toc122072670)

[JSON Files 36](#_Toc122072671)

[MARC Files 37](#_Toc122072672)

[TXT Files 37](#_Toc122072673)

[XML Data Dictionaries 37](#_Toc122072674)

[XML Files 38](#_Toc122072675)

[Zip Files 39](#_Toc122072676)

# Introduction

The PWGSC WPSS Open Data Tool provides a method to perform tests on datasets and dictionaries being submitted for the Open Data Registry to ensure they are compliant with Open Data standards. The WPSS Open Data Tool reviews all of the documents and analyses each one for compliance to standards.

## Tool Limitations

Some of the output of the Open Data Tool is in English only. The tool is using third party software components. The source of these components is available only in the language that it was authored.

# Installation

The WPSS Open Data Tool requires Perl, Python and Java distributions installed. A number of optional components have additional requirements.

## System Requirements (Windows)

To install the WPSS Open Data Tool on Windows, the follow requirements must be met:

* Windows 7 or later,
* Oracle Java runtime environment 1.8.0 (other versions may not work), or OpenJDK version 11 or newer,
  + Oracle Java is available at <https://www.java.com/en/download/>
  + OpenJDK is available at <https://openjdk.java.net/>
* Latest version of Python 2.7.x (does not work with Python 3),
  + available from http://python.org/downloads/
* Strawberry Perl 5.26.1 or later, either 32 or 64 bit, (does not work with ActiveState Perl),
  + available from <http://strawberryperl.com>
  + Only one installation of Perl on the system. Multiple installations may cause problems.

## Installing the WPSS Tool

To install the WPSS Open Data Tool, double-click the WPSS\_Install.exe file and follow the instructions on the screen.



The default installation folder for the WPSS Open Data Tool is C:\Program Files\WPSS\_Tool.

## Optional Requirements

The following are the requirements for optional components of the WPSS Open Data Tool.

### Chrome Headless User Agent

The following are requirements for using the Chrome headless user agent. This user agent is newer and more feature rich than the default [PhantomJS](https://phantomjs.org/) user agent.

* Node version 8 or newer
  + available from <https://nodejs.org/en/download/>
* Chrome browser version 69 or newer
  + available from <https://www.google.com/chrome/>
* ChromeDriver node module that matches the Chrome browser version
  + List locally installed chromedriver version ‘npm list chromedriver -g’
  + List all available chromedriver versions ‘npm view chromedriver versions’
  + Remove module ‘npm uninstall -g chromedriver’
  + Install a specific version ‘npm install -g [chromedriver@89.0.0](mailto:chromedriver@89.0.0)’
* Puppeteer-core node module. Installed via command prompt.
  + npm install –g puppeteer-core

The above node module checks and installation can be performed using the **install\_puppeteer.pl** script located in the top level folder of the installation. Google Chrome and Node must be installed manually.

### Pa11y accessibility tool

The following are requirements for using the [Pa11y accessibility tool](https://pa11y.org/). This tool is required for the Pa11y testcase profile of the WPSS Open Data Tool.

* Pa11y node module (required for optional Pa11y accessibility tool) installed via command prompt.
  + npm install –g pa11y
  + <https://github.com/pa11y/pa11y>

The above node module checks and installation can be performed using the **install\_pa11y.pl** script located in the top level folder of the installation.

### Deque AXE accessibility tool

The following are requirements for using the [Deque AXE accessibility](https://github.com/dequelabs/axe-core-npm/tree/develop/packages/cli) command line tool. This tool is required for the Deque AXE testcase profile of the WPSS Open Data Tool.

* Deque AXE core module and command line interface (CLI) module. Installed via command prompt.
  + npm install -g @axe-core/cli
  + Install the required chromedriver module (see above).

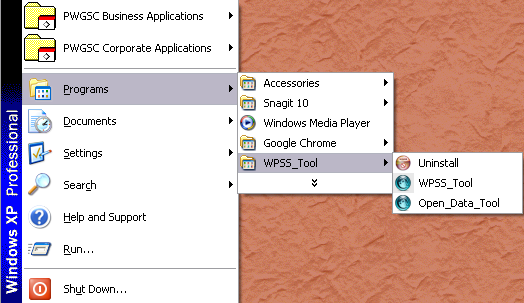
The above node module checks and installation can be performed using the **install\_deque\_axe.pl** script located in the top level folder of the installation.

# Uninstall

To remove the WPSS Open Data Tool from a workstation, run the uninstall script.

**To uninstall the Open Data Tool:**

Go to **Start > Programs > WPSS\_Tool > Uninstall**.



## Uninstalling Perl

**To remove the Perl installation:**

1. Go to **Start > Settings > Control Panel**.
2. Click **Add or Remove Programs**.
3. Locate Strawberry Perl and click **Remove**.

## Uninstalling Python

**To remove the Perl installation:**

1. Go to **Start > Settings > Control Panel**.
2. Click **Add or Remove Programs**.
3. Locate Pythonand click **Remove**.

# Using the WPSS Open Data Tool

**To start the PWGSC WPSS Open Data Tool:**

1. Go to **Start > Programs > Web and Open Data Validator**.
2. Click the **Open Data Tool** icon.

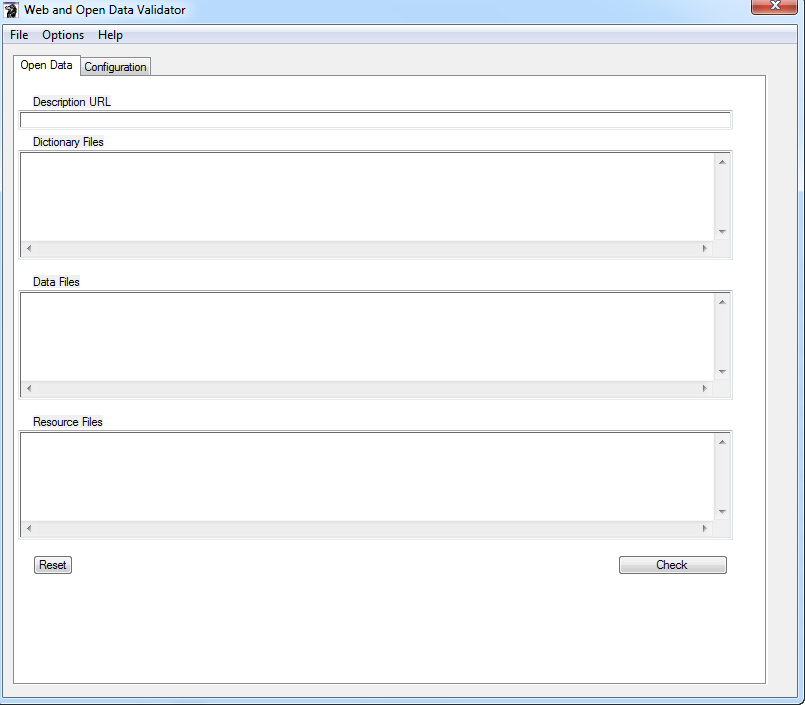
Alternatively, using Windows Explorer, navigate to the **C:\Program Files\WPSS\_Tool** folder and double click the **open\_data\_tool.pl** file.

The main window consists of two tabs:

* **Open Data** is for entering the dictionary, data and resource file references for analysis.
* **Configuration** is for configuring the WPSS Open Data Tool profile.

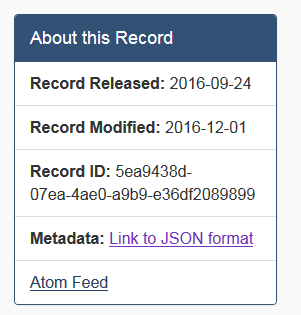
## Open Data Tab

To analyse the documents identified for the Open Data Registry, enter the URLs for the dictionary, data and resource files into the WPSS Open Data Tool. Use the **Open Data** tab to enter this information.



### Description URL

The URL to a JSON object that describes the dataset details (e.g. dictionary file, data files, etc.). A link to the JSON object can be found on the dataset details page on the <http://open.canada.ca> site (e.g. <http://open.canada.ca/data/en/dataset/5ea9438d-07ea-4ae0-a9b9-e36df2089899> ). The link appears below the “About this Record” right hand menu in a link labelled “Link to JSON format”



### Dictionary Files

The list of URLs for data dictionary files.

### Data Files

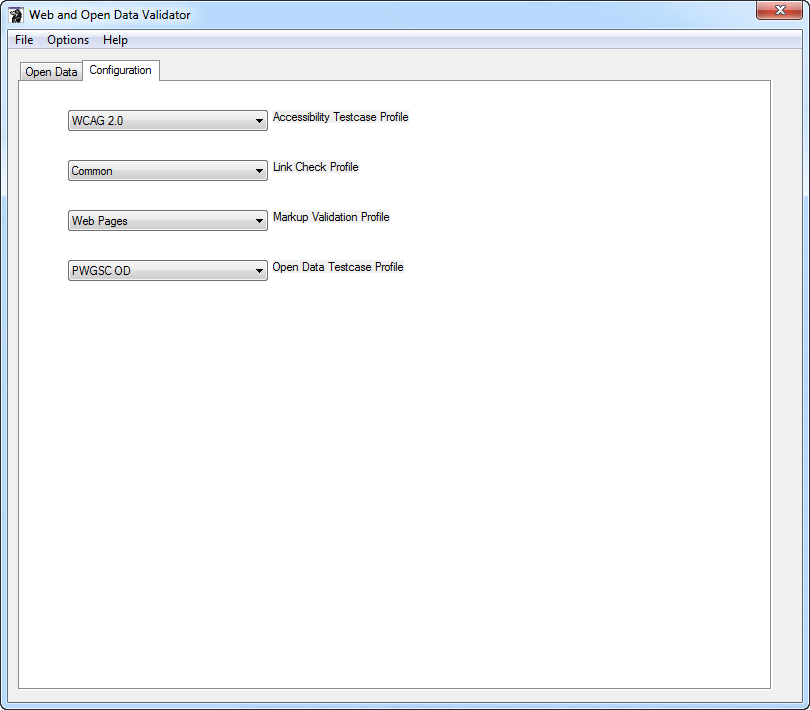
The list of URLs for data files (e.g. CSV).

### Resource Files

The list of URLs for dataset resource files (e.g. supporting documentation).

## Configuration Tab

The **Configuration** tab enables you to select the option for the analysis.



**Accessibility Testcase Profile** – Controls the Web Accessibility tests that are performed on any HTML supporting files that are part of the dataset. Options include:

* WCAG 2.0 – WCAG 2.0 A and AA checks
* Deque AXE accessibility – Deque AXE rules (<https://dequeuniversity.com/rules/axe>)
* Pa11y accessibility – Pa11y accessibility test tool rule (<https://github.com/pa11y/pa11y>)
* All accessibility – all WCAG, Deque AXE and Pa11y checks.
* None – no accessibility checking

**Link Check Profile** – Controls the link checking that is performed on any supporting files that are part of the dataset. Options include:

* Common – Broken links, cross language links
* All – all link violations
* Errors – report broken links only
* IPV4 – report IP addresses used as domain names
* None

**Markup Validation Profile** – Controls the markup validation that is performed on any supporting files that are part of the dataset. This option does not affect the markup validation of data files. Options include:

* Web Pages – only web pages are validated, does not include supporting files such as CSS, JavaScript
* All – all files
* None – no validation of supporting files.

**Open Data Testcase Profile** – Controls the set of checks performed on data dictionary and data files in the dataset. Options include:

* PWGSC OD – PWGSC specific checks as well as all common checks
* PWGSC OD (core) – PWGSC specific checks as well as all common checks except for duplicate data cells and data inconsistency checks.
* Common – TBS and best practise checks
* None – No open data checks

# Profiles

You can save the open data details in a configuration profile for sharing, or easy access if you want to use the WPSS Open Data Tool again.

**To save the data profile:**

1. Go to **File > Save Open Data Config**.
2. Select a folder and file name for the configuration file.
3. Click **OK**.

To best manage the configuration files, it is suggested that you save the configuration files to the **C:\Program Files\WPSS\_Tool\profiles** folder.

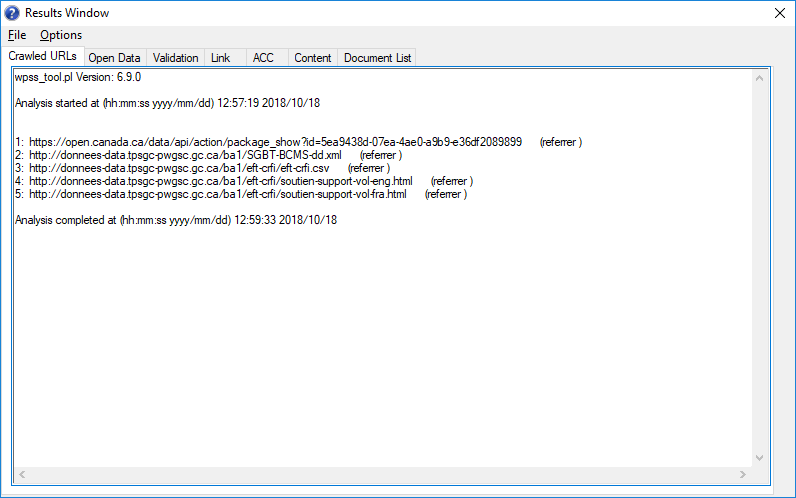
You can load a previously saved site configuration. Loading a saved configuration file loads the **Open Data** tab fields. Once loaded, you can modify the information if required.

**To open a saved profile:**

1. Go to **File > Load Open Data Config**.
2. Locate the folder and file.
3. Click **OK**.

# Results Window

The **Results Window** includes seven tabs containing the output of an individual analysis from the WPSS Open Data Tool. The output in each tab includes a header that lists information and the time and date when the analysis started.



**Crawled URLs** tab – Provides a list of the URLs the WPSS Open Data Tool analysed. It lists the referrer page to indicate how the crawler reached a particular page. Use this tab to monitor the WPSS Open Data Tool to ensure it is actively crawling and analysing the files.

**Open Data** tab – Lists the documents that contain open data check violations.

**Validation** tab – Contains the output of the HTML/XHTML, XML, CSV, MARC21 and JSON validation tools.

**Link** tab – Lists the link violations found in HTML supporting files.

**ACC** tab – Contains the output of the accessibility check.

**Content** tab – Contains the results of content checks.

**Document List** tab – The Open Data Tool writes information to this tab after completing the site analysis. It contains the sorted list of files reviewed.

The WPSS Open Data Tool includes the time and date at the end of the report in each tab.

## Stopping the Analysis

If you need to stop the analysis while it is running, in the Results Window, go to **Options > Stop Crawl**. This stops the WPSS Open Data Tool after processing the current document. The results include a note at the bottom of each output tab in the Results Window indicating that the analysis was aborted.

## Reporting Passes and Fails

The default behaviour of the analysis tools is to report only URLs that fail checks. You can view results for both passes and fails. To see both passes and fails, in the WPSS Open Data Tool window, go to **Options > Report Fails and Passes**. The URL for documents that pass checks are recorded in the results output.

To see only failed pages, go to **Options > Report Fails Only**.

## Saving Results

To save the analysis results, in the Results Window, go to **File > Save As**. Select the file name and folder path in the file chooser dialog box. The results are stored in a number of files, one for each result tab. Each file name contains a suffix identifying the report type.

For example, you save the results in the file od\_results, the actual files are:

* od\_results\_acc.txt
* od\_results\_cont.txt
* od\_results\_crawl.txt
* od\_results\_od.txt
* od\_results\_urls.txt
* od\_results\_val.txt

There will be 2 additional CSV files created, one being a detailed listing of all the dataset file URLs and the other is a CSV version of the test case results.

* od\_results\_file\_inventory.csv
* od\_results\_rslts.csv

It is suggested that you use the same name for the results as the open data profile name, and save the results in the **\WPSS\_Tool** folder.

# Command Line Interface

The WPSS Open Data Tool is available from the GUI interface and from the command prompt.

**To access the command line version:**

1. Go to **Start > Programs > Accessories > Command Prompt**.
2. Change to the **Program Files\WPSS\_Tool** directory.
3. Run the program **open\_data\_tool.pl** with the command:

open\_data\_tool.pl –cli –o <profile file>

where <profile file> is the path to the profile file containing the details of the open data to be analysed.

## Status and Progress

As the command line WPSS Open Data Tool runs and analyses documents, the URLs of the documents appear in the console window. Use this to monitor the WPSS Open Data Tool to ensure it is actively crawling and analysing the data files.

## Language Switching

You can toggle the language of the analysis results using a command line option.

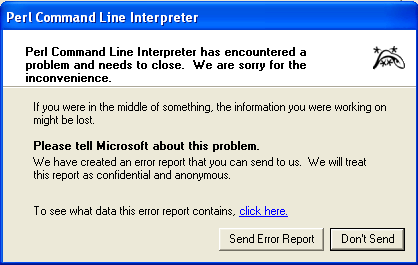
|  |  |
| --- | --- |
| **Option** | **Result** |
| -eng | Analysis results are provided in English. |
| -fra | Analysis results are provided in French. |

If no language selection is made, the language is determined from the operating system.

# Troubleshooting

## Perl Command Line Interpreter Error

The WPSS Open Data Tool uses a number of Perl modules. These modules may have errors that cause the program to fail. You may encounter the following message:



## 500 Internal Server Error

When analysing data, the WPSS Open Data Tool may report a “500 Internal server error”. This may be due to a limitation on the Web server and not the WPSS Open Data Tool. The problem may be due to the Web server not handling the “Range” setting in the HTTP GET operation. The WPSS Open Data Tool sets a size limit on GET operations to avoid getting extremely large documents.

To avoid this error, you can change the WPSS Open Data Tool configuration file setting to not include the “Range” setting in a HTTP GET operation.

Using a simple text file editor, such as WordPad:

1. Open the file c:\Program Files\wpss\_tool\conf\wpss\_tool.config.
2. Locate the following lines in the file:

#  
# Max User Agent Size limits the size of files accepted in a GET  
# request. A value of 0 means we can accept documents of any size.  
# A value of 0 also removes the Range field from the HTTP header.  
#  
#User\_Agent\_Max\_Size 0

1. Remove the leading ‘#’ character from the #User\_Agent\_Max\_Size 0.

This functionality is **not** available through the user interface, only by directly editing the configuration file.

# Test Cases

This section describes the criteria the Open Data Tool uses to verify the syntax and content of Open Data dataset files.

## OD\_URL - Open data URL

Check URLs of the open data files.

### Checks performed

* That the dataset file (dictionary, data and resource) URL exists (i.e. the HTTP response code is 200).
* Checks that the mime-type and file suffix are consistent (e.g. a .csv file has a mime type of text/csv or text/x-comma-separated-values).
* Checks that the format attribute for a dataset file in the dataset JSON description structure is consistent with the mime-type provided in the HTTP response header (e.g. a dataset file with format “csv” has a mime-type of text/csv or text/x-comma-separated-values).
* That there is the same number of language specific data files in a dataset.
* Checks that the dataset JSON description URL has a “application/json” mime-type.

### Does not check

* That the dataset file is available to the public. For example, available on the Internet.

## OD\_ENC - Character encoding

Check the character encoding used in files.

### Checks performed

* That the content of the dataset files is UTF-8 encoded.
* Checks XML files to ensure the ‘encoding’ attribute of the <xml> tag specifies UTF-8 encoding.
* Check that all JSON files specify a charset in the HTTP::Response or contain a UTF-8 BOM (Byte Order Mark) at the beginning of the file.

­

## OD\_REG – Open Data registry

Check open data registry field values.

### Checks performed

* Check that there is a non-blank value for required registry fields (e.g. portal\_release\_date, date\_published).
* Check that dataset resources have description, format, type and URL values.
* Check that all JSON files specify a charset in the HTTP::Response or contain a UTF-8 BOM (Byte Order Mark) at the beginning of the file.

## OD\_VAL – Validation

Check validity of content mark-up or syntax.

### Checks performed

**All file types:**

* That the file has content.

**CSV files:**

* That CSV data files are well formed, properly delimited and quoted.
* That CSV files conform to [Common Format and MIME Type for Comma-Separated Values (CSV) Files](http://www.rfc-base.org/txt/rfc-4180.txt) (RFC-4180)
* Check that a .csv file contains comma separators rather than tab separators for fields.

**JSON files:**

* That JSON files are well formed.
* That the JSON file contains a $schema name/value pair to specify the schema.
* If the file contains a $schema name/value pair, it checks that the JSON content conforms to that schema.

**MARC files:**

* That MARC files are well formed and validate to the [Library of Congress MARC21 specification](https://www.loc.gov/marc/specifications/).

**XML Files:**

* That the file is well formed with properly nested and closed tags.
* That the file includes either a DOCTYPE of Schema specification.
* If the file includes an ‘xsi:schemaLocation’ attribute or xsi:noNamespaceSchemaLocation attribute to specify a schema, it checks that the XML file conforms to that schema.
* If the file includes a DOCTYPE tag, it checks that the XML file conforms to that doctype.

**ZIP Files:**

* That the ZIP file is a valid ZIP archive.

**JSON Dataset Description Files:**

* That required fields are in the JSON dataset description file.

## OD\_DATA – Data Quality

Data Quality checks.

### Checks performed

**All Files:**

* If there is a data dictionary, and the data dictionary specifies a regular expression for data values, then check that the data value matches the expression.

**CSV Files:**

* That CSV data files have the same number of fields in each row.
* There is data in each row (i.e. the row does not have only empty fields).
* There are no duplicate column headings.
* If there is a data dictionary, and the data dictionary specifies a data pattern, then check that the data value matches the pattern.
* If there is a data dictionary, and the data dictionary specifies a data condition (defined by CSV Schema 1.0), then check that the data value matches the condition.
* If there is a data dictionary, check that the heading order matches in language variants of data files.
* If there is a data dictionary, check that the language appropriate label is used for headings with multiple language specific labels.
* If the first line of a multi-line field value contains whitespace only.
* Text fields for possible WCAG 2.0 plain text technique accessibility errors (e.g. lists in cells).
* Check that column types (e.g. numeric, text) match for all language variations of the same data file.
* Check that the sum of the values for numeric data columns match for all language variations of the same data file.
* Check that the number of non-blank cells in each column matches for all language variations of the same data file.
* Check that the sum of the values for date data (YYYY-MM-DD) columns match for all language variations of the same data file.

**JSON-CSV Files (JSON encoded CSV files):**

* Check that each data array items contains the same number of fields in the same order.
* If there is a data dictionary, and the data dictionary specifies a data pattern, then check that the data value matches the pattern.
* Check that language variants of JSON-CSV data files have the same number of fields in each data array item.
* Check that language variants of JSON-CSV data files have the same number of items the data arrays.
* Check that the number of JSON-CSV data array items matches the number of data rows in the corresponding CSV data file.
* Check that the number of fields in a JSON-CSV data array items matches the number of columns in the corresponding CSV data file.
* Check that the data type (e.g. numeric, text) of fields in a JSON-CSV data array items matches the data type of columns in the corresponding CSV data file.
* Check that the number of non-blank of fields in a JSON-CSV data array items matches the number of non-blank cells in the corresponding columns in the corresponding CSV data file.
* Check that the sum of the values for numeric fields in a JSON-CSV data array items matches the sum of the values for numeric cells in the corresponding columns in the corresponding CSV data file.
* Check that the field names in JSON-CSV data array items match the column names in the corresponding CSV data file.
* Check that the data values in a JSON-CSV data array item matches the data values of the values of the corresponding row in the corresponding CSV data file. The order of data array items must match the order of rows in the corresponding CSV file.

**XML Files:**

* If there is a data dictionary, and the data dictionary specifies a data pattern, and an XML tag name matches a data dictionary term, then check that the data value matches the pattern.

## TBS\_QRS\_Structured

Your data is made available in a structured format.

### Checks performed

**Portal Metadata:**

* Check that there is 1 or more resources of type dataset that are not alternate format data files (i.e. resource title does not contain the word alternate).

**TBS\_QRS\_Open**

Your data is made available in a nonproprietary structured format.

### Checks performed

**Portal Metadata:**

**TBS\_QRS\_Noticed**

Receive 5 or more User Ratings.

### Checks performed

None

**TBS\_QRS\_Popular**

Receive 10 or more User Ratings with an average score of 3 or higher.

### Checks performed

None

**TBS\_QRS\_Timely**

Dataset is updated in accordance with user-defined frequency.

### Checks performed

**Portal Metadata:**

* Check that the date published or date modified value is consistent with the update frequency of the dataset (e.g. datasets should be updated within 1.5 times the frequency value).

**TBS\_QRS\_Documented**

Valid supporting documentation.

### Checks performed

**Portal Metadata:**

* Check that there are 1 or more resources of type “guide” (e.g. supporting documentation, data dictionary, etc.).

**TBS\_QRS\_Connected**

Valid maintainer email supplied for the owning department.

### Checks performed

**Portal Metadata:**

* Check that the maintainer email address is a valid address and not a TBS address (for non TBS datasets).

**TBS\_QRS\_Readable**

Description is at appropriate readability level.

### Checks performed

**Portal Metadata:**

* Check that the Flesch-Kincaid reading level of the dataset’s English description does not exceed the maximum allowed level. There is no similar check for the French description as no rating level algorithm has been specified.

## TBS\_QRS\_Online

The link to the dataset is valid.

### Checks performed

**Dataset Resources**

* Checks that dataset resources links are valid (e.g. are not broken links).

## TBS\_QRS\_Tidy

Dataset is successfully validated via a linting tool and returns no errors.

### Checks performed

**Dataset Resources**

* Checks that CSV files validate with the [csvlint](https://github.com/Clever/csvlint) tool and Perl CSV parser.
* Check that JSON files validate with Perl JSON module.
* If JSON files specify a schema, validate the schema and JSON file against the schema using the [jsonschema](https://pypi.python.org/pypi/jsonschema/2.6.0) tool.

## TBS\_QRS\_Honest

The supplied data file matches the indicated file type.

### Checks performed

**Portal Metadata:**

* Checks that the format value (e.g. CSV, TXT, XML) specified has the expected mime-type or file name suffix.
* Checks that the file mime-type or file name suffix has the expected format value.

## TBS\_QRS\_International

Data is supplied with UTF-8 character encoding.

### Checks performed

**Dataset Resources**

* Checks HTTP response Content-Type header for “charset=utf-8” or the X-Meta-Charset header for “UTF-8”.

## TP\_PW\_OD\_BOM – UTF-8 BOM

UTF-8 Byte Order Mark.

### Checks performed

**CSV and Text Files:**

* Check that all CSV and TXT files contain a UTF-8 BOM (Byte Order Mark) at the beginning of the file.

**JSON Files:**

* Check that file does NOT contain a UTF-8 BOM (Byte Order Mark) at the beginning of the file.

## TP\_PW\_OD\_CONT – Content

Content quality checks.

### Checks performed

**CSV Files:**

* Check that the number of columns match for all language variations of the same data file.
* Check for numeric values with a large number of digits as the value may be truncated when read by programs (e.g. in Excel).
* Check for a possible Excel formula as the content of a cell.
* Check for leading or trailing whitespace in cell content. Leading space is accepted to avoid possible interpretation of cell content as an Excel formula.
* Check the type of data (e.g. numeric, text) is consistent in a column.
* Check text columns for formatted currency values (e.g. $ 1.00 or 1,00 $).
* Check text columns for formatted numeric values with thousands separators (e.g. 1,000,000).
* Check for scientific notation values (e.g. 1.3e-005).

**JSON-CSV Files (JSON encoded CSV files):**

* Check for duplicate data array field name.
* Check that there are no duplicate data array items (i.e. the same data in 2 array items).

**XML Data Dictionaries:**

* Check for whitespace inside data dictionary terms.

## TP\_PW\_OD\_CONT\_CONSISTENCY – Content

Content quality checks for consistent punctuation and capitalization of data values.

### Checks performed

**CSV Files:**

* Check for consistent punctuation and pluralization of cell values (e.g. “St-John” versus “St. John” versus “St John”).
* Check for consistent capitalization of cell values (e.g. “St John” versus “ST JOHN”).
* Check for consistent accented characters in text values (e.g. “dommage cause à un véhicule” versus “dommage causé à un véhicule”).

## TP\_PW\_OD\_CONT\_DUP – Duplicate Content

Content quality checks for duplicate rows or columns of data.

### Checks performed

**CSV Files:**

* There are no duplicate rows (i.e. the same data in 2 rows).
* There are no duplicate columns of data (i.e. the same data in 2 columns). Duplicate columns of empty fields or fields containing a 0 (zero) are allowed.

**JSON-CSV Files (JSON encoded CSV files):**

* Check that there are no duplicate data array items (i.e. the same data in 2 array items).

## TP\_PW\_OD\_DATA – PWGSC Data Quality

PWGSC data quality checks.

### Checks performed

**All Files:**

* If the file is uncompressed it does not exceed the maximum size limit (default 50 Mb).
* If there are language specific variations of a data file, check that required languages (i.e. English, French) variations exist.

**CSV Files:**

* The CSV data files must have a header row.
* There are no blank column headers.
* There is no leading or trailing whitespace in column headers.
* If at least 25% of the fields on the first row matches the data dictionary terms, then checks to see if all fields match dictionary terms.

**JSON-CSV Files (JSON encoded CSV files):**

* Check there is a CSV data file for each JSON-CSV data file.
* All data array item field names are in the data dictionary.

**JSON Dataset Description Files:**

* The JSON content is a valid open.canada.ca JSON dataset description.
* There is at least 1 data dictionary specified.
* There is at least 1 data file specified.
* If there are language specific data files, that all required languages (e.g. English, French) are found.

**Zip Files:**

* The ZIP archive contains only 1 type of file (e.g. all CSV or all XML).

## TP\_PW\_OD\_DD – Data Dictionary

PWGSC data dictionary file.

### Checks performed

**XML Data Dictionaries:**

* That XML data dictionaries follow the [PWGSC XML Data Dictionary specification](https://masource-mysource.spac-pspc.gc.ca/eng/or-tr/communications/web/ldw-wg/do-od/Pages/xml_dd.aspx).
* That there are no duplicate headings.
* That there are no duplicate definitions.
* That definitions are provided for both official languages.

# Checks by File Type

This section lists Open Data checks by file type. The test case identifier and description is provided.

## CSV Files

* OD\_ENC – Check that the content of the file is UTF-8 encoded.
* OD\_VAL – Check that files conform to [Common Format and MIME Type for Comma-Separated Values (CSV) Files](http://www.rfc-base.org/txt/rfc-4180.txt) (RFC-4180)
* OD\_VAL - Check that file has content.
* OD\_VAL - Check that a .csv file contains comma separators rather than tab separators for fields.
* OD\_DATA - If there is a data dictionary, and the data dictionary specifies a regular expression for data values, then check that the data value matches the expression.
* OD\_DATA – Check that CSV data files have the same number of fields in each row.
* OD\_DATA – Check there is data in each row (i.e. the row does not have only empty fields).
* OD\_DATA - Check there are no duplicate column headings.
* OD\_DATA - If there is a data dictionary, and the data dictionary specifies a data pattern, then check that the data value matches the pattern.
* OD\_DATA - If there is a data dictionary, and the data dictionary specifies a data condition (defined by CSV Schema 1.0), then check that the data value matches the condition.
* OD\_DATA – Check if the first line of a multi-line field contains whitespace only.
* OD\_DATA - Check if text fields for possible WCAG 2.0 plain text technique accessibility errors (e.g. lists in cells).
* OD\_DATA - Check that column types (e.g. numeric, text) match for all language variations of the same data file.
* OD\_DATA - Check that the sum of the values for numeric data columns match for all language variations of the same data file.
* OD\_DATA - Check that the number of non-blank cells in each column matches for all language variations of the same data file.
* OD\_DATA - Check that the sum of the values for date data (YYYY-MM-DD) columns match for all language variations of the same data file.
* TP\_PW\_OD\_BOM - Check that file contains a UTF-8 BOM (Byte Order Mark) at the beginning of the file.
* TP\_PW\_OD\_CONT - Check that the number of columns match for all language variations of the same data file.
* TP\_PW\_OD\_CONT - Check for numeric values with a large number of digits as the value may be truncated when read by programs (e.g. in Excel).
* TP\_PW\_OD\_CONT - Check for a possible Excel formula as the content of a cell.
* TP\_PW\_OD\_CONT - Check for leading or trailing whitespace in cell content. Leading space is accepted to avoid possible interpretation of cell content as an Excel formula.
* TP\_PW\_OD\_CONT - Check the type of data (e.g. numeric, text) is consistent in a column.
* TP\_PW\_OD\_CONT\_CONSISTENCY - Check for consistent punctuation and pluralization of cell values (e.g. “St-John” versus “St. John” versus “St John”).
* TP\_PW\_OD\_CONT\_CONSISTENCY - Check for consistent capitalization of cell values (e.g. “St John” versus “ST JOHN”).
* TP\_PW\_OD\_CONT\_CONSISTENCY - Check for consistent accented characters in text values (e.g. “dommage cause à un véhicule” versus “dommage causé à un véhicule”).
* TP\_PW\_OD\_CONT\_DUP – Check that there are no duplicate rows (i.e. the same data in 2 rows).
* TP\_PW\_OD\_CONT\_DUP - Check that there are no duplicate columns of data (i.e. the same data in 2 columns). Duplicate columns of empty fields or fields containing a 0 (zero) are allowed.
* TP\_PW\_OD\_DATA - If the file is uncompressed it does not exceed the maximum size limit (default 50 Mb).
* TP\_PW\_OD\_DATA - If there are language specific variations of a data file, check that required languages (i.e. English, French) variations exist.
* TP\_PW\_OD\_DATA - Check that the CSV data file has a header row.
* TP\_PW\_OD\_DATA - Check that there are no blank column headers.
* TP\_PW\_OD\_DATA – Check that there is no leading or trailing whitespace in column headers.
* TP\_PW\_OD\_DATA – Check if at least 25% of the fields on the first row matches data dictionary terms, then checks to see if all fields match dictionary terms.

## JSON-CSV Files (JSON encoded CSV files)

A JSON-CSV is a CSV data formatted in JSON syntax. The JSON file contains a schema specification and a “data” array that contain items which represent rows of CSV data. Each item contains fields corresponding to the CSV column headings and values for the cell values.

An example JSON-CSV file:

{ "$schema": "<schema url>",

"data": [{

"field 1": "Value 1",

"field 2": "Value 2",

....

"field n": "Value n"

},

{

"field 1": "Value 1",

"field 2": "Value 2",

....

"field n": "Value n"

}

...

]

}

Checks for JSON-CSV files:

* OD\_ENC – Check that the content of the file is UTF-8 encoded.
* OD\_ENC - Check that files specify a charset in the HTTP::Response or contain a UTF-8 BOM (Byte Order Mark) at the beginning of the file.
* OD\_VAL - That files are well formed.
* OD\_VAL - That the file contains a $schema name/value pair to specify the schema.
* OD\_VAL - If the file contains a $schema name/value pair, it checks that the JSON content conforms to that schema.
* OD\_VAL - Check that file has content.
* OD\_DATA - If there is a data dictionary, and the data dictionary specifies a regular expression for data values, then check that the data value matches the expression.
* OD\_DATA - Check that each data array items contains the same number of fields in the same order.
* OD\_DATA - Check that language variants of JSON-CSV data files have the same number of fields in each data array item.
* OD\_DATA - Check that language variants of JSON-CSV data files have the same number of items the data arrays.
* OD\_DATA - Check that the number of JSON-CSV data array items matches the number of data rows in the corresponding CSV data file.
* OD\_DATA - Check that the number of fields in a JSON-CSV data array items matches the number of columns in the corresponding CSV data file.
* OD\_DATA - Check that the data type (e.g. numeric, text) of fields in a JSON-CSV data array items matches the data type of columns in the corresponding CSV data file.
* OD\_DATA - Check that the number of non-blank of fields in a JSON-CSV data array items matches the number of non-blank cells in the corresponding columns in the corresponding CSV data file.
* OD\_DATA - Check that the sum of the values for numeric fields in a JSON-CSV data array items matches the sum of the values for numeric cells in the corresponding columns in the corresponding CSV data file.
* OD\_DATA - Check that the field names in JSON-CSV data array items match the column names in the corresponding CSV data file.
* OD\_DATA - Check that the data values in a JSON-CSV data array item matches the data values of the values of the corresponding row in the corresponding CSV data file. The order of data array items must match the order of rows in the corresponding CSV file.
* TP\_PW\_OD\_BOM - Check that file does NOT contain a UTF-8 BOM (Byte Order Mark) at the beginning of the file.
* TP\_PW\_OD\_CONT-DUP - Check for duplicate data array field name.
* TP\_PW\_OD\_CONT\_DUP - Check that there are no duplicate data array items (i.e. the same data in 2 array items).
* TP\_PW\_OD\_DATA - If the file is uncompressed it does not exceed the maximum size limit (default 50 Mb).
* TP\_PW\_OD\_DATA - If there are language specific variations of a data file, check that required languages (i.e. English, French) variations exist.
* TP\_PW\_OD\_DATA - Check there is a CSV data file for each JSON-CSV data file.
* TP\_PW\_OD\_DATA - Check there that all data array item field names are in the data dictionary.

## JSON Dataset Description Files

* OD\_VAL - That files are well formed.
* OD\_VAL - That required fields are in the JSON dataset description file.
* OD\_VAL - Check that file has content.
* TP\_PW\_OD\_DATA – Check that the JSON content is a valid open.canada.ca JSON dataset description.
* TP\_PW\_OD\_DATA - Check that there is at least 1 data dictionary specified.
* TP\_PW\_OD\_DATA - Check that there is at least 1 data file specified.
* TP\_PW\_OD\_DATA - If there are language specific data files, check that all required languages (e.g. English, French) are found.

## JSON Files

* OD\_ENC – Check that the content of the file is UTF-8 encoded.
* OD\_ENC - Check that files specify a charset in the HTTP::Response or contain a UTF-8 BOM (Byte Order Mark) at the beginning of the file.
* OD\_VAL - That files are well formed.
* OD\_VAL - That the file contains a $schema name/value pair to specify the schema.
* OD\_VAL - If the file contains a $schema name/value pair, it checks that the JSON content conforms to that schema.
* OD\_VAL - Check that file has content.
* TP\_PW\_OD\_BOM - Check that file does NOT contain a UTF-8 BOM (Byte Order Mark) at the beginning of the file.
* TP\_PW\_OD\_DATA - If the file is uncompressed it does not exceed the maximum size limit (default 50 Mb).
* TP\_PW\_OD\_DATA - If there are language specific variations of a data file, check that required languages (i.e. English, French) variations exist.

## MARC Files

* OD\_VAL - That files are well formed and validate to the [Library of Congress MARC21 specification](https://www.loc.gov/marc/specifications/).
* TP\_PW\_OD\_DATA - If there are language specific variations of a data file, check that required languages (i.e. English, French) variations exist.

## TXT Files

* OD\_ENC – Check that the content of the file is UTF-8 encoded.
* OD\_VAL - Check that file has content.
* TP\_PW\_OD\_BOM - Check that file contains a UTF-8 BOM (Byte Order Mark) at the beginning of the file.
* TP\_PW\_OD\_DATA - If the file is uncompressed it does not exceed the maximum size limit (default 50 Mb).
* TP\_PW\_OD\_DATA - If there are language specific variations of a data file, check that required languages (i.e. English, French) variations exist.

## XML Data Dictionaries

* OD\_ENC – Check that the content of the file is UTF-8 encoded.
* OD\_ENC - Checks that the ‘encoding’ attribute of the <xml> tag specifies UTF-8 encoding.
* OD\_VAL - That the file is well formed with properly nested and closed tags.
* OD\_VAL - That the file includes either a DOCTYPE of Schema specification.
* OD\_VAL - If the file includes an ‘xsi:schemaLocation’ attribute or xsi:noNamespaceSchemaLocation attribute to specify a schema, it checks that the XML file conforms to that schema.
* OD\_VAL - If the file includes a DOCTYPE tag, it checks that the XML file conforms to that doctype.
* OD\_VAL - Check that file has content.
* TP\_PW\_OD\_CONT - Check for whitespace inside data dictionary terms.
* TP\_PW\_OD\_DD – Check that XML data dictionaries follow the [PWGSC XML Data Dictionary specification](https://masource-mysource.spac-pspc.gc.ca/eng/or-tr/communications/web/ldw-wg/do-od/Pages/xml_dd.aspx).
* TP\_PW\_OD\_DD – Check that there are no duplicate headings.
* TP\_PW\_OD\_DD – Check that there are no duplicate definitions.
* TP\_PW\_OD\_DD – Check that definitions are provided for both official languages.

## XML Files

* OD\_ENC – Check that the content of the file is UTF-8 encoded.
* OD\_ENC - Checks that the ‘encoding’ attribute of the <xml> tag specifies UTF-8 encoding.
* OD\_VAL - That the file is well formed with properly nested and closed tags.
* OD\_VAL - That the file includes either a DOCTYPE of Schema specification.
* OD\_VAL - If the file includes an ‘xsi:schemaLocation’ attribute or xsi:noNamespaceSchemaLocation attribute to specify a schema, it checks that the XML file conforms to that schema.
* OD\_VAL - If the file includes a DOCTYPE tag, it checks that the XML file conforms to that doctype.
* OD\_VAL - Check that file has content.
* OD\_DATA - If there is a data dictionary, and the data dictionary specifies a regular expression for data values, then check that the data value matches the expression.
* TP\_PW\_OD\_DATA - If the file is uncompressed it does not exceed the maximum size limit (default 50 Mb).
* TP\_PW\_OD\_DATA - If there are language specific variations of a data file, check that required languages (i.e. English, French) variations exist.

## Zip Files

* OD\_VAL - That the file is a valid ZIP archive.
* TP\_PW\_OD\_DATA – Check that the ZIP archive contains only 1 type of file (e.g. all CSV or all XML).