TECHNICAL SUMMARY

CANADA'S WEATHER API ACCESS

This technical summary gives a detailed but general explanation of how Canada's Weather API Access - version 4.0 workflow works, as shown in Figure 1. For a more detailed review of each node configuration, it is encouraging to download the workflow from the KNIME Community Hub [1]. The workflow consists of two sub-workflows: one to write the .csv files where the data recovered or scraped from ECCC is saved, and the second that updates these files each time the workflow accesses the API. Both sub-workflows are divided into five main sections, which are:

1. API Links (metanode)
2. Execution Date&Time (metanode)
3. Filtering (metanode)
4. API Access (metanode)
5. CSV Files Writer/CSV Updating (metanode)

Sections 1, 2, 4 and 5 are metanodes. Metanodes are groups of nodes that allow for building neater workflows [2].

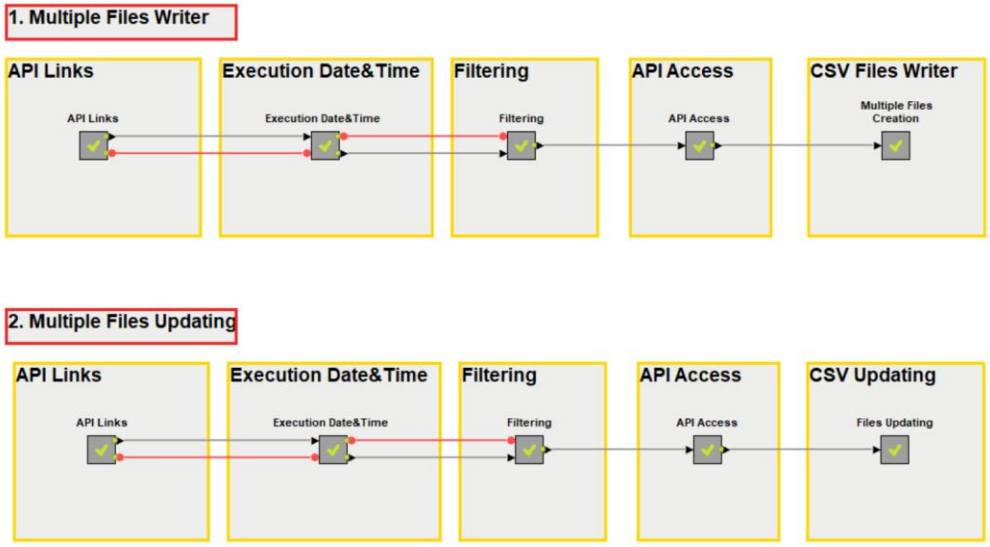

Figure 1. Canada's weather API access workflow. Created by the author [1].

API Links (Metanode)

Figure 2 shows the nodes that make up the API Links metanode. The Webpage Retriever node accesses the following link:

https://dd.weather.gc.ca/citypage_weather/xml/siteList.xml

which provides a complete list of the cities available in the ECCC API with their respective site codes and generates the information as an XML body. Subsequently, the XPath node extracts

the site code and city and province parameters from the XML body and sorts them into a table. Then, the Column Expressions node takes the values of the columns of the table and generates a link for each city with the following structure:

https://dd.weather.gc.ca/citypage_weather/xml/{XX}/{SiteNameCode}_{L}.xml

where XX is a 2-letter provincial or territorial code indicating the area covered by the forecasts, SiteNameCode corresponds to the site codes used in the API city site list, and L corresponds to a single letter indicating the language of the file. It can be either f (French) or e (English) [8]. An example of an XML file is the following:

https://dd.weather.gc.ca/citypage_weather/xml/SK/s0000797_e.xml

which corresponds to the city of Saskatoon, SK.

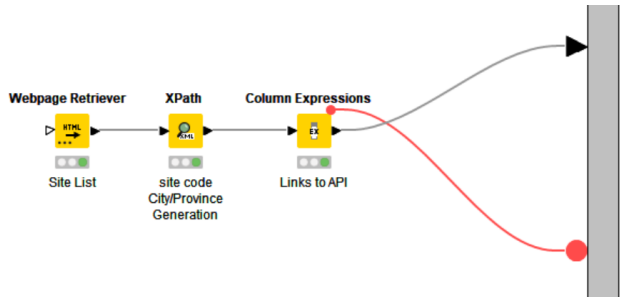

Figure 2. API Links metanode overview.

The API Links metanode output table generates 855 links that are available to access the ECCC API. Figure 3 shows a sample of the first rows of the output table.

Row ID	site_code	province	city	API_links
Row0_1	s0000001	AB	Athabasca	https://dd.weather.gc.ca/citypage_weather/xml/AB/s0000001_e.xml
Row0_2	s0000002	BC	Clearwater	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000002_e.xml
Row0_3	s0000003	BC	Valemount	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000003_e.xml
Row0_4	s0000004	BC	Grand Forks	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000004_e.xml
Row0_5	s0000005	BC	McBride	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000005_e.xml
Row0_6	s0000006	BC	Merritt	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000006_e.xml
Row0_7	s0000007	BC	Masset	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000007_e.xml
Row0_8	s0000008	BC	Invermere	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000008_e.xml
Row0_9	s0000009	BC	Dome Creek	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000009_e.xml
Row0_10	s0000010	MB	Little Grand ...	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000010_e.xml
Row0_11	s0000011	MB	Oxford House	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000011_e.xml
Row0_12	s0000012	MB	Bissett	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000012_e.xml
Row0_13	s0000013	MB	Shamattawa	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000013_e.xml
Row0_14	s0000014	MB	York Factory	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000014_e.xml
Row0_15	s0000015	MB	Flin Flon	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000015_e.xml

Figure 3. API Links metanode output table.

Execution Date&Time (Metanode)

Figure 4 shows the nodes that make up the Execution Date&Time metanode. The Create Date&Time Range node generates the execution date and time of the local server with the time zone. If the workflow is executed locally, the date and time generated corresponds to the city where the workflow is being executed, however, if the workflow is executed through the KNIME server, the date and time corresponds to the time zone of the KNIME server. The Modify Time Zone node is used to avoid ambiguity in the execution date and time, which ensures that the time zone is always CST and corresponds to the University of Saskatchewan time zone where this research is

carried out. The Date&Time to String node and the String to Date&Time node together have the objective of removing the time zone label from the table column and also the seconds parameter from the time, which is irrelevant for the research. The Column Appender node combines the API Links metanode output with the String to Date&Time node output in one table.

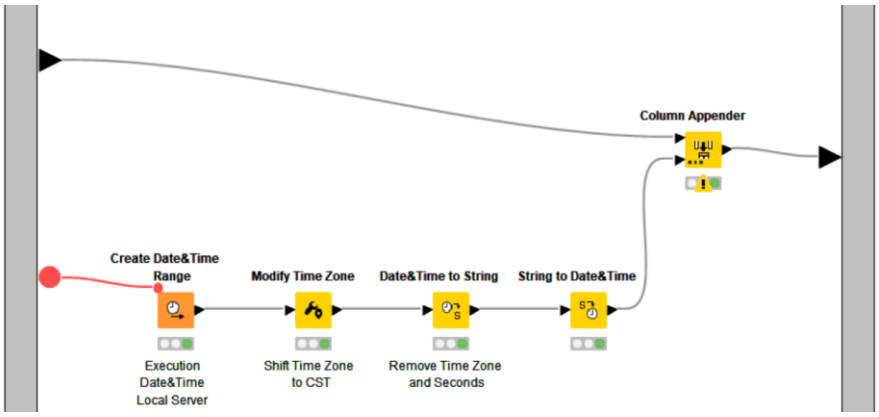

Figure 4. Execution Date&Time metanode overview.

Figure 5 shows a sample of the first rows of the Execution Date&Time metanode output table.

site_code	province	city	API_links	date_time_execution_CST
s0000001	AB	Athabasca	https://dd.weather.gc.ca/citypage_weather/xml/AB/s0000001_e.xml	2024-11-06T10:01
s0000002	BC	Clearwater	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000002_e.xml	2024-11-06T10:01
s0000003	BC	Valemount	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000003_e.xml	2024-11-06T10:01
s0000004	BC	Grand Forks	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000004_e.xml	2024-11-06T10:01
s0000005	BC	McBride	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000005_e.xml	2024-11-06T10:01
s0000006	BC	Merritt	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000006_e.xml	2024-11-06T10:01
s0000007	BC	Masset	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000007_e.xml	2024-11-06T10:01
s0000008	BC	Invermere	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000008_e.xml	2024-11-06T10:01
s0000009	BC	Dome Creek	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000009_e.xml	2024-11-06T10:01
s0000010	MB	Little Grand ...	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000010_e.xml	2024-11-06T10:01
s0000011	MB	Oxford House	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000011_e.xml	2024-11-06T10:01
s0000012	MB	Bissett	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000012_e.xml	2024-11-06T10:01
s0000013	MB	Shamattawa	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000013_e.xml	2024-11-06T10:01
s0000014	MB	York Factory	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000014_e.xml	2024-11-06T10:01
s0000015	MB	Flin Flon	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000015_e.xml	2024-11-06T10:01

Figure 5. Execution Date&Time metanode output table.

Filtering (Metanode)

Figure 6 shows the nodes that make up the Filtering metanode, which filters 22 cities available in the ECCC API: Calgary, Charlottetown, Edmonton, Fredericton, Halifax, Iqaluit, Montréal, Ottawa, Prince George, Québec, Regina, Saskatoon, St. John's, Thunder Bay, Toronto, Vancouver, Victoria, Whitehorse, Winnipeg, Yellowknife, Waskesiu Lake, and Nipawin.

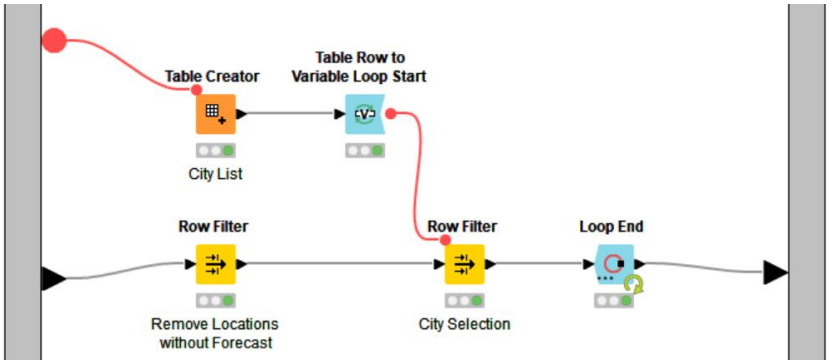

Figure 6. Filtering metanode overview.

Figure 7 shows the Filtering metanode output table.

Row ID	site_code	province	city	API_links	date_time_...
Row0_45	s0000045	AB	Edmonton	https://dd.weather.gc.ca/citypage_weather/xml/AB/s0000045_e.xml	2024-11-06T10...
Row0_47	s0000047	AB	Calgary	https://dd.weather.gc.ca/citypage_weather/xml/AB/s0000047_e.xml	2024-11-06T10...
Row0_141	s0000141	BC	Vancouver	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000141_e.xml	2024-11-06T10...
Row0_146	s0000146	BC	Prince George	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000146_e.xml	2024-11-06T10...
Row0_192	s0000193	MB	Winnipeg	https://dd.weather.gc.ca/citypage_weather/xml/MB/s0000193_e.xml	2024-11-06T10...
Row0_248	s0000250	NB	Fredericton	https://dd.weather.gc.ca/citypage_weather/xml/NB/s0000250_e.xml	2024-11-06T10...
Row0_277	s0000280	NL	St. John's	https://dd.weather.gc.ca/citypage_weather/xml/NL/s0000280_e.xml	2024-11-06T10...
Row0_315	s0000318	NS	Halifax	https://dd.weather.gc.ca/citypage_weather/xml/NS/s0000318_e.xml	2024-11-06T10...
Row0_363	s0000366	NT	Yellowknife	https://dd.weather.gc.ca/citypage_weather/xml/NT/s0000366_e.xml	2024-11-06T10...
Row0_391	s0000394	NU	Iqaluit	https://dd.weather.gc.ca/citypage_weather/xml/NU/s0000394_e.xml	2024-11-06T10...
Row0_408	s0000411	ON	Thunder Bay	https://dd.weather.gc.ca/citypage_weather/xml/ON/s0000411_e.xml	2024-11-06T10...
Row0_427	s0000430	ON	Ottawa (Ka...	https://dd.weather.gc.ca/citypage_weather/xml/ON/s0000430_e.xml	2024-11-06T10...
Row0_455	s0000458	ON	Toronto	https://dd.weather.gc.ca/citypage_weather/xml/ON/s0000458_e.xml	2024-11-06T10...
Row0_580	s0000583	PE	Charlottetown	https://dd.weather.gc.ca/citypage_weather/xml/PE/s0000583_e.xml	2024-11-06T10...
Row0_617	s0000620	QC	Québec	https://dd.weather.gc.ca/citypage_weather/xml/QC/s0000620_e.xml	2024-11-06T10...
Row0_632	s0000635	QC	Montréal	https://dd.weather.gc.ca/citypage_weather/xml/QC/s0000635_e.xml	2024-11-06T10...
Row0_772	s0000775	BC	Victoria	https://dd.weather.gc.ca/citypage_weather/xml/BC/s0000775_e.xml	2024-11-06T10...
Row0_785	s0000788	SK	Regina	https://dd.weather.gc.ca/citypage_weather/xml/SK/s0000788_e.xml	2024-11-06T10...
Row0_794	s0000797	SK	Saskatoon	https://dd.weather.gc.ca/citypage_weather/xml/SK/s0000797_e.xml	2024-11-06T10...
Row0_822	s0000825	YT	Whitehorse	https://dd.weather.gc.ca/citypage_weather/xml/YT/s0000825_e.xml	2024-11-06T10...

Figure 7. Filtering metanode output table.

API Access (Metanode)

The ECCC API offers 24-hour weather forecasts in XML files accessed through the links generated by the API Links metanode with a data structure, as shown in Figure 8. Figure 9 shows the nodes that make up the API Access metanode, which additionally consists of two UTC Shift and Null Values Control metanodes (Figure 10 and Figure 11, respectively).

The Get Request node accesses the ECCC API XML files and generates a column with the retrieved XML body. The Python Script node uses Python code that introduces the value 99999 every time an empty cell in the table is retrieved from the API. Later, this value of 99999 is removed with a Column Expression node. This procedure was necessary since preliminary tests showed that when an empty cell appeared in the data, the workflow stacked all the non-empty cells on top of the table, and the time series lost its structure. The code was generated with Cody, an AI coding assistant [3] available for Visual Studio Code [4], and the code was tested by the author several times to validate that the output was correct.

Figure 8. ECCC API XML data glance.

Subsequently, the XPath node accesses the XML file using an XPath value query adjusted to the hierarchical structure of the XML tree. For example, for the weather condition parameter, the XPath value query was as follows:

/siteData/hourlyForecastGroup/hourlyForecast/condition

In the same way, all the parameters available in the XML were accessed using a similar hierarchical structure. In addition, some necessary transformations were made, such as replacing the word Calm with the value 0 in the wind speed parameter column because it was the only string value in the column, and the workflow generated an error when transformations were carried out in that column because it had two data types, string and numerical. Additionally, all dates within the workflow were adjusted to have the following ISO 8601 structure [5]:

yyyy-MM-dd'T'HH:mm

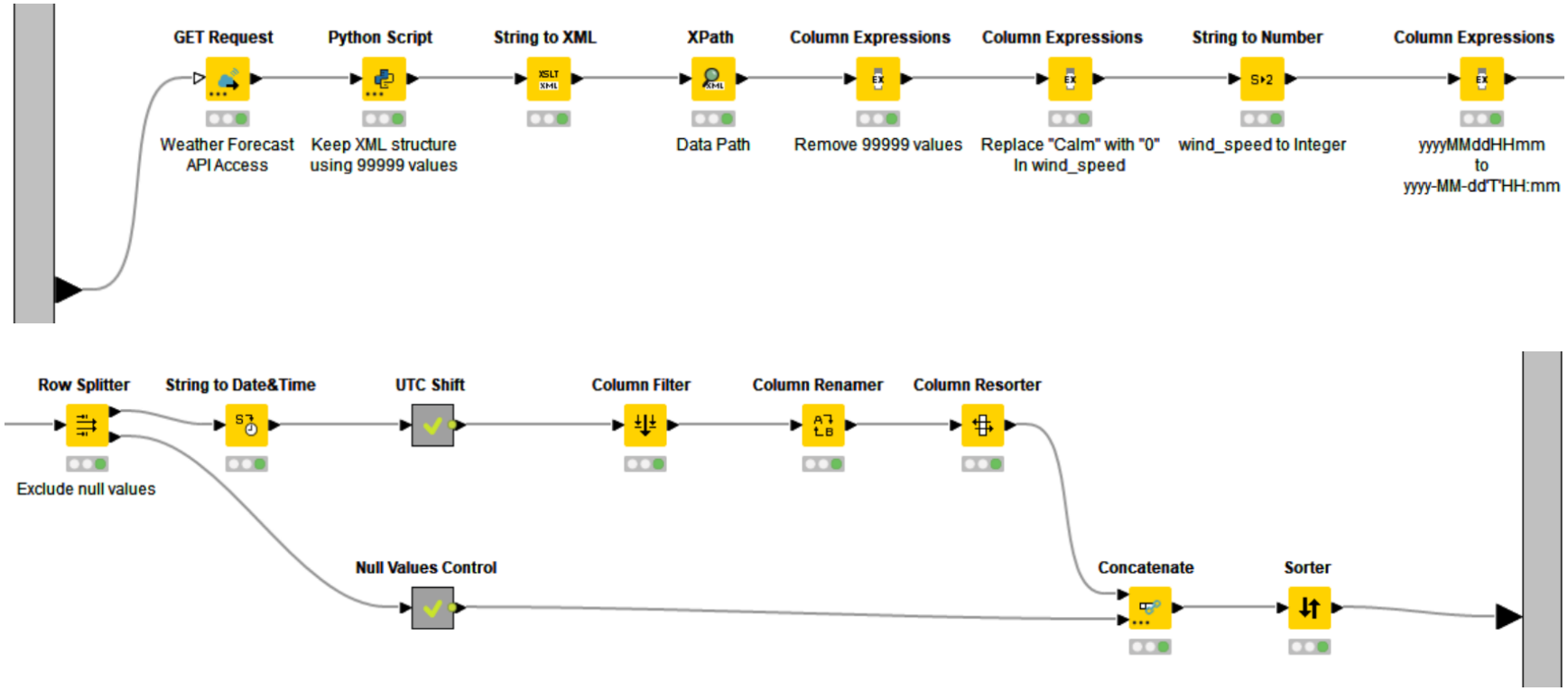

Figure 9. API Access metanode overview.

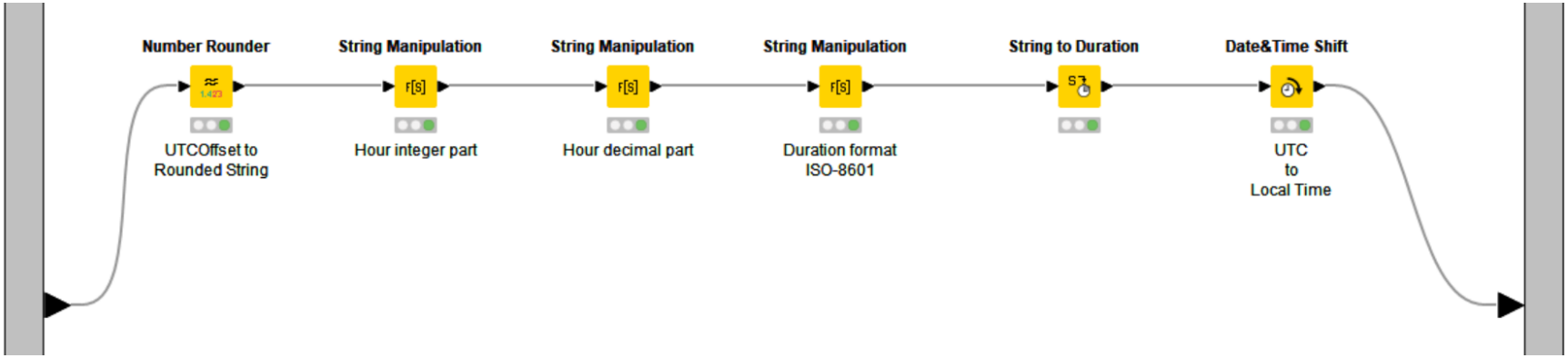

Figure 10. UTC Shift metanode overview.

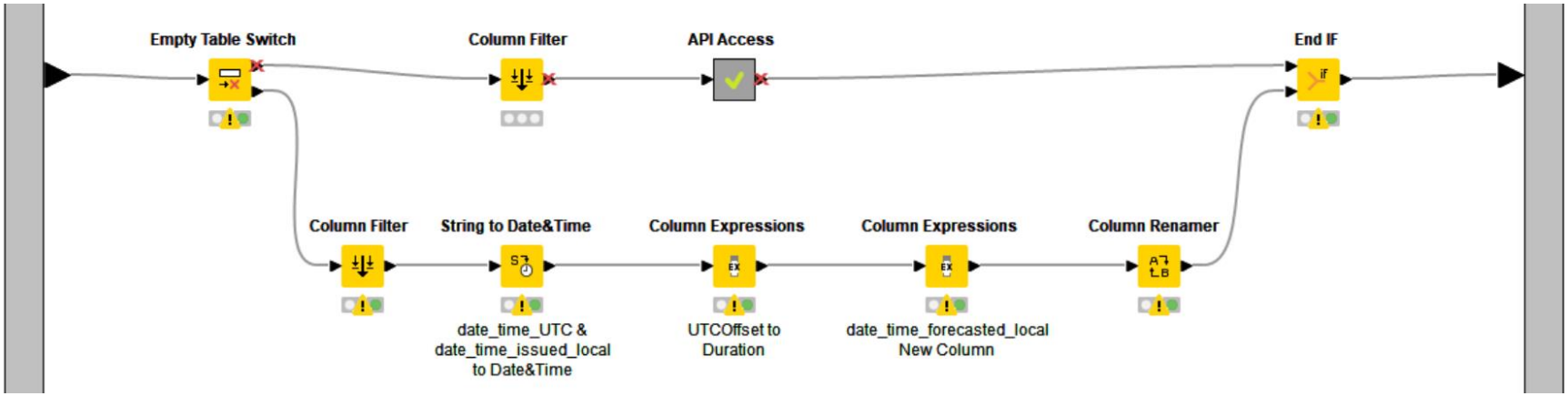

Figure 11. Null Values Control metanode overview.

The branch that contains the Null Values Control metanode in Figure 9 was necessary to introduce because, in preliminary tests, the workflow recovered some null values from the API; this occurred mainly when the workflow executed the complete list of locations available in the API, so possibly the large amount of processed data generated this error. To deal with this, the author created a conditional branch that manages these errors and calls the API again to obtain data without errors (see Figure 11).

Figure 10 shows the UTC Shift metanode that is responsible for transforming the UTC Offset value retrieved from the API and changing it to the ISO 8601 duration format and then using this value to convert the workflow date and time expressed in UTC to the local date and time. Finally, the main branch of the workflow is filtered, resorted, and concatenated with the null values control branch, generating the output of Figure 12. Because the table is large and has several columns, only a glance at the columns for the weather parameters is shown.

weather_conditions	temp_(°C)	likelihood_of_precip _(%)	wind_speed	wind_direction	wind_gust	wind_chill	uv_index	humidex
A mix of sun and cloud	-1	0	5	VR	?	-3	?	?
A mix of sun and cloud	1	0	5	VR	?	?	?	?
A mix of sun and cloud	3	0	15	W	?	?	1	?
A mix of sun and cloud	4	0	15	W	?	?	1	?
A mix of sun and cloud	5	0	15	W	?	?	?	?
Mainly cloudy	6	0	15	W	?	?	?	?
Mainly cloudy	6	0	15	W	?	?	?	?
Mainly cloudy	5	0	15	W	?	?	?	?
Mainly cloudy	5	0	15	W	?	?	?	?
Mainly cloudy	3	0	15	W	?	?	?	?
Mainly cloudy	2	0	15	W	?	?	?	?
Mainly cloudy	0	0	15	W	?	?	?	?
Mainly cloudy	-1	0	15	W	?	-6	?	?
Mainly cloudy	-2	0	15	W	?	-7	?	?
Partly cloudy	-3	0	5	VR	?	-5	?	?
A few clouds	-4	0	5	VR	?	-6	?	?
A few clouds	-5	0	5	VR	?	-7	?	?
A few clouds	-6	0	5	VR	?	-8	?	?
A few clouds	-6	0	5	VR	?	-9	?	?
A few clouds	-7	0	5	VR	?	-9	?	?
A few clouds	-7	0	5	VR	?	-10	?	?
A few clouds	-7	0	5	VR	?	-10	?	?
A few clouds	-7	0	5	VR	?	-10	?	?

Figure 12. API Access metanode output table.

CSV Files Writer (Metanode)

Figure 13 shows the nodes that make up the CSV Files Writer metanode. The GroupBy node generates a list of the total site codes at the output of the API Access metanode. The Table Row to Variable Loop Start node uses this list of site codes to start a loop with the site codes as a variable. This variable is used by the Create File/Folder Variable node to generate the paths where the .csv files will be saved. These paths are generated as a variable that will be read by the CSV Writer node, which will be responsible for saving the .csv files in the desired location. The Row Filter node gives the CSV Writer node the filtered table with the site codes as a variable and the Variable Loop End node closes the loop.

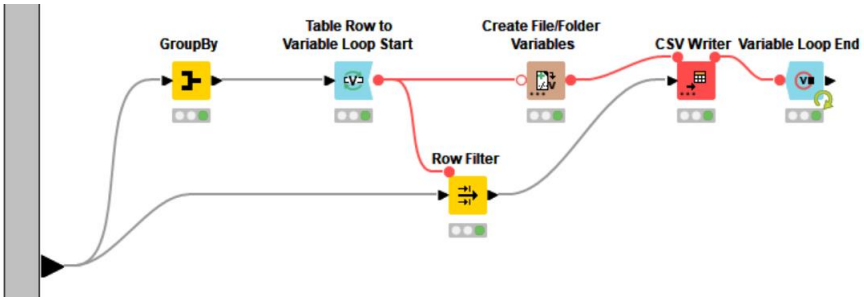

Figure 13. CSV Files Writer metanode overview.

Figure 14 shows a few .csv files saved on KNIME server.

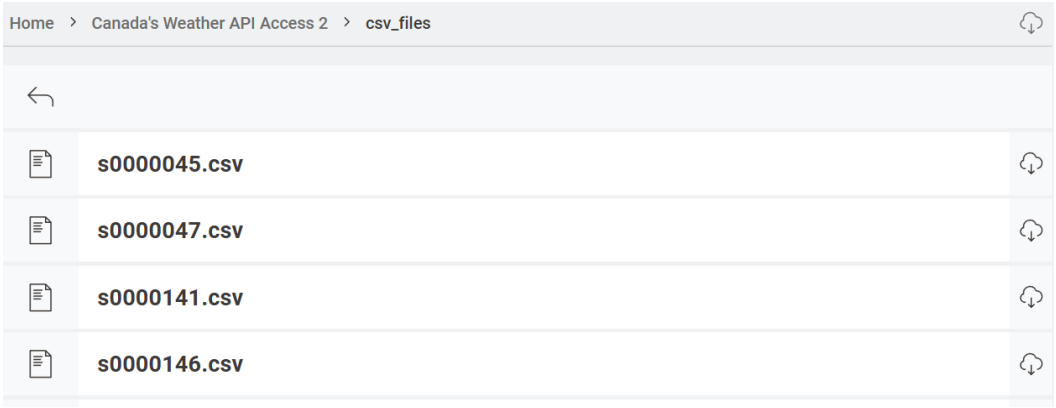

Figure 14. CSV files generated by the workflow.

CSV Updating (Metanode)

After the .csv files have been created, the second sub-workflow comes into action, which is responsible for updating the .csv files every time the workflow accesses the API. The second sub-workflow also consists of five sections, where sections 1 to 4 (Figure 1) are identical to those of the first sub-workflow. However, as shown in Figure 15, the last section corresponds to the CSV Updating metanode, which is similar but more complex than the CSV Files Writer metanode that was described previously.

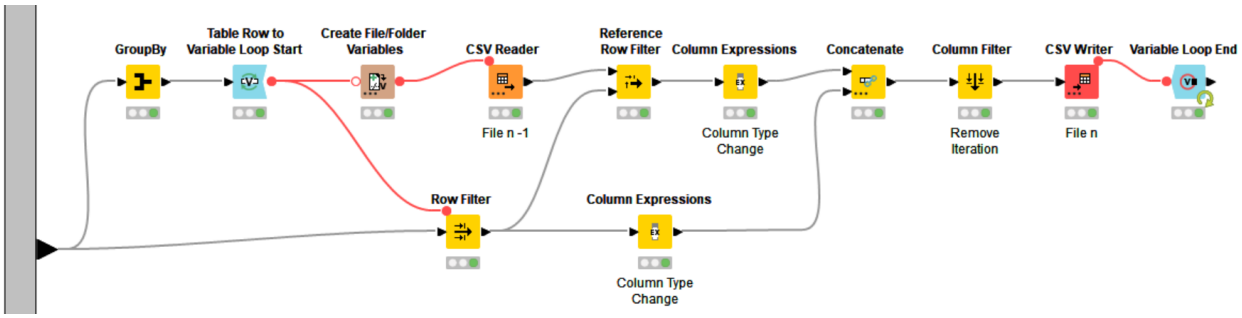

Figure 15. CSV Updating metanode overview.

One of the differences in the CSV Updating metanode is that a CSV Reader node is used, which reads the previously saved .csv files. A Reference Row Filter node is also utilized, which allows

rows to be filtered from the first table using the second table as a reference. Because .csv files store information in string or numeric data and do not use all KNIME data formats, some transformations are necessary to maintain data consistency. The Concatenate node is then used to join the historical data with the new data retrieved from the API, and finally, the new file is saved with the CSV Writer node. This entire process is also done in a loop so that the process is repeated for each of the cities queried.

REFERENCES

[1] C. Diaz-Urribarri, “Canada’s Weather API Access - Version 4.0,” Nov. 21, 2024. Accessed: Nov. 21, 2024. [Online]. Available: https://hub.knime.com/s/Hmb-WHa5xMKeWlB7/9

[2] KNIME AG, *KNIME Components Guide - Version 5.3*. Zurich, 2024.

[3] Sourcegraph, “Cody,” 2024, 1.0.0. [Online]. Available: https://sourcegraph.com/

[4] Microsoft, “Visual Studio Code,” 2024, 1.95.3. [Online]. Available: https://code.visualstudio.com/

[5] D. Adams, “JSON Developer’s Guide, 21c,” 2022, *Oracle Database*. Accessed: Nov. 16, 2024. [Online]. Available: https://docs.oracle.com/en/database/oracle/oracle-database/21/adjsn/iso-8601-date-time-and-duration-support.html