

Content Validator – HTML & XML package scanner

# Overview

The INK project deals with a of content sets day in and day out which makes tracking and validating the various content sets being used by various teams, practically impossible and humanly non feasible. The two most important formats of packages are .XML and .HTML format packages. These packages are used by the various consumption systems of INK. Ensuring that the underlying content within each package is in right place, is of utmost importance to all the stakeholders involved in the process of using these packages for their Application purposes. With the huge amounts of content sets which are being piled up daily, there is a clear gap when it comes to the amount of validation involved in checking these packages and the turn around time involved in delivering these to the application related teams.

# Problem statement

Due to a heavy inflow of content packages, there is a need to quickly validate and ensure that the most important checklist we go after while checking these packages is met in a very efficient and timely manner. The amount of time involved is high and the manual QA approach only leaves scope for a chance of error. We need a solution which is script based and which cuts down the time drastically and is very reliable.

# solution & expected result

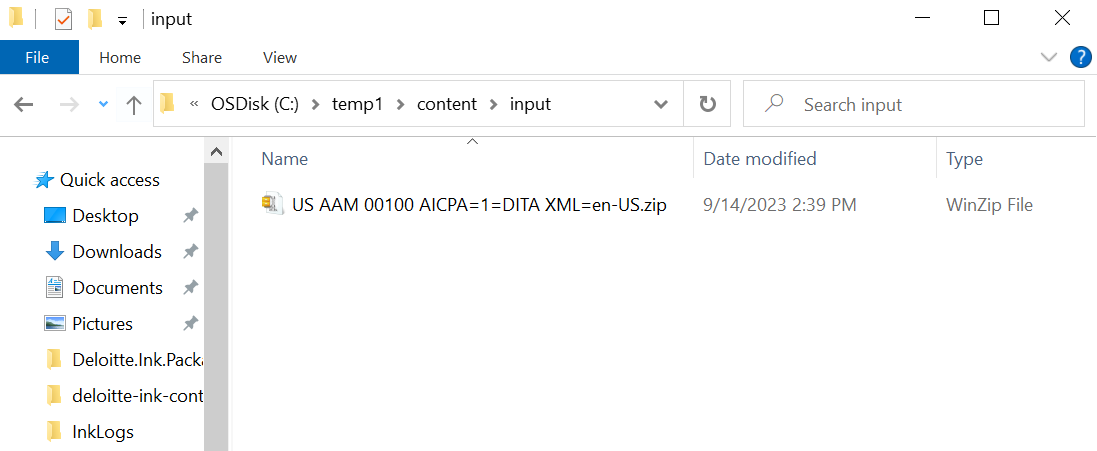
|  |  |
| --- | --- |
|  | The most effective solution for the problem is designing a simple yet effective console-based application which with just a few clicks, does the process of quickly scanning the packages and triggering an error when something is not in-line or as expected per the checklist. The solution is ‘Content Validator’ |

# implementation of the solution

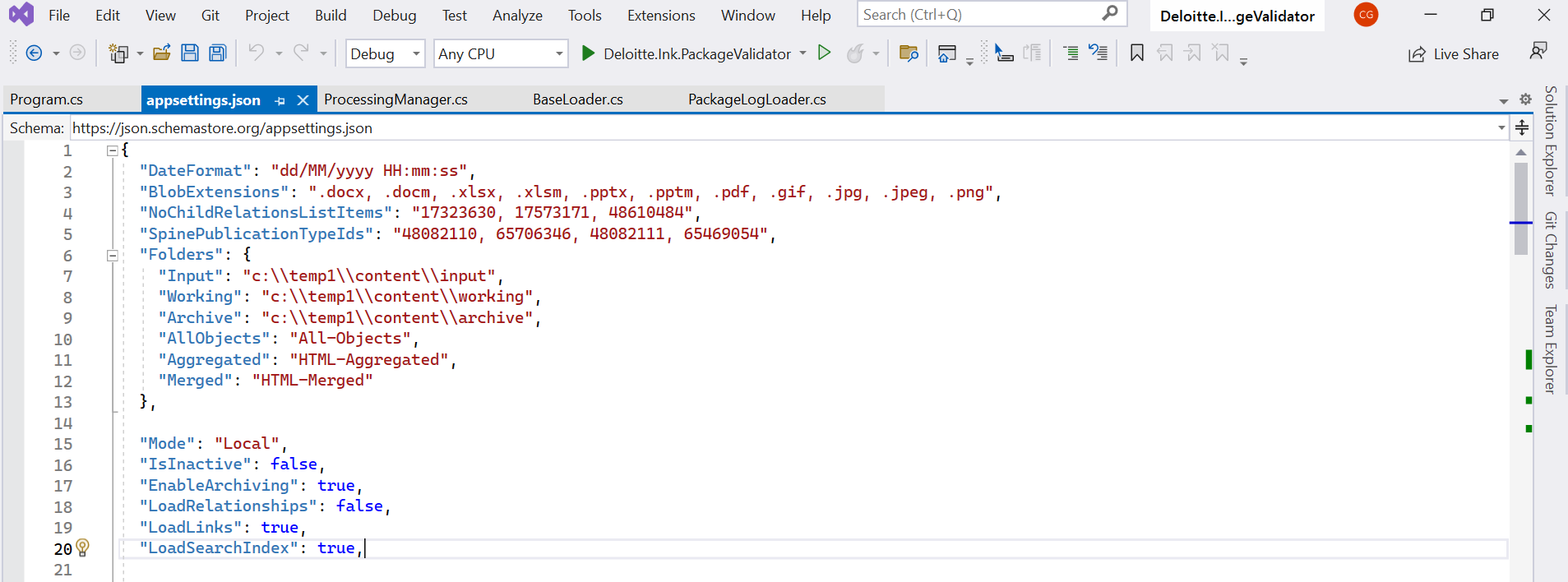
This application is presently configured in the local environment and is written using TargetFramework Dotnet 6.0. For the Application, INK Content Explorer, we have been supporting Reporting services for the latest Spring 2024 updates. We had to scan the huge database of content packages as part of Reporting, and we were able to identify content issues using this Content Validator script. We could cut down the time we spent on package validation and also were accurately identifying content issues. Going forward, we will be using this Content Validator at the INK Level and look out for any opportunities where bulk content processing and scanning is involved.

Steps for using the Application in local machine.

Step 1: Import or shift the packages which need to be validated into the ‘Input’ folder.

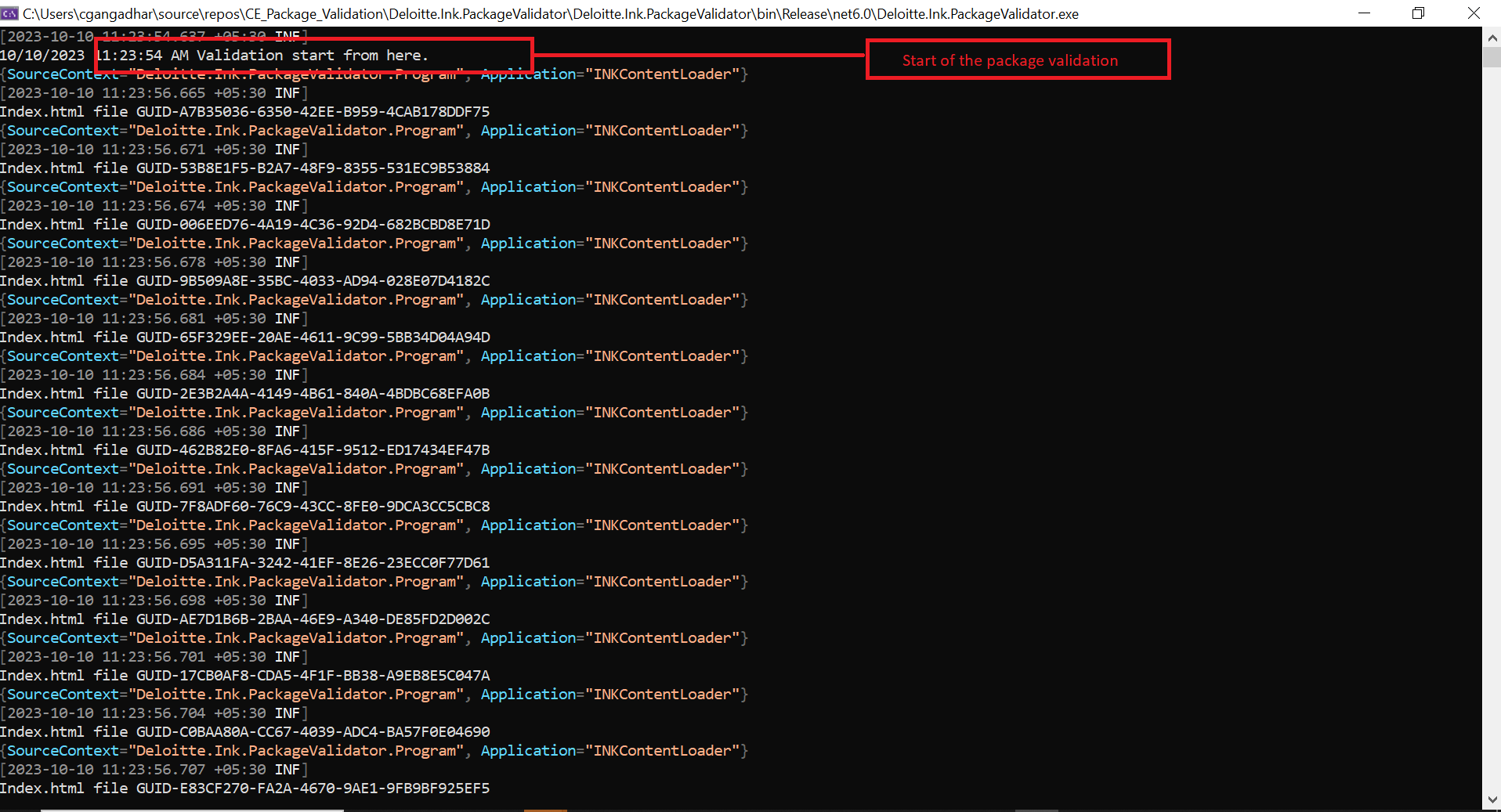


Step 2: Open the Packagevalidator and click on ‘Run.’

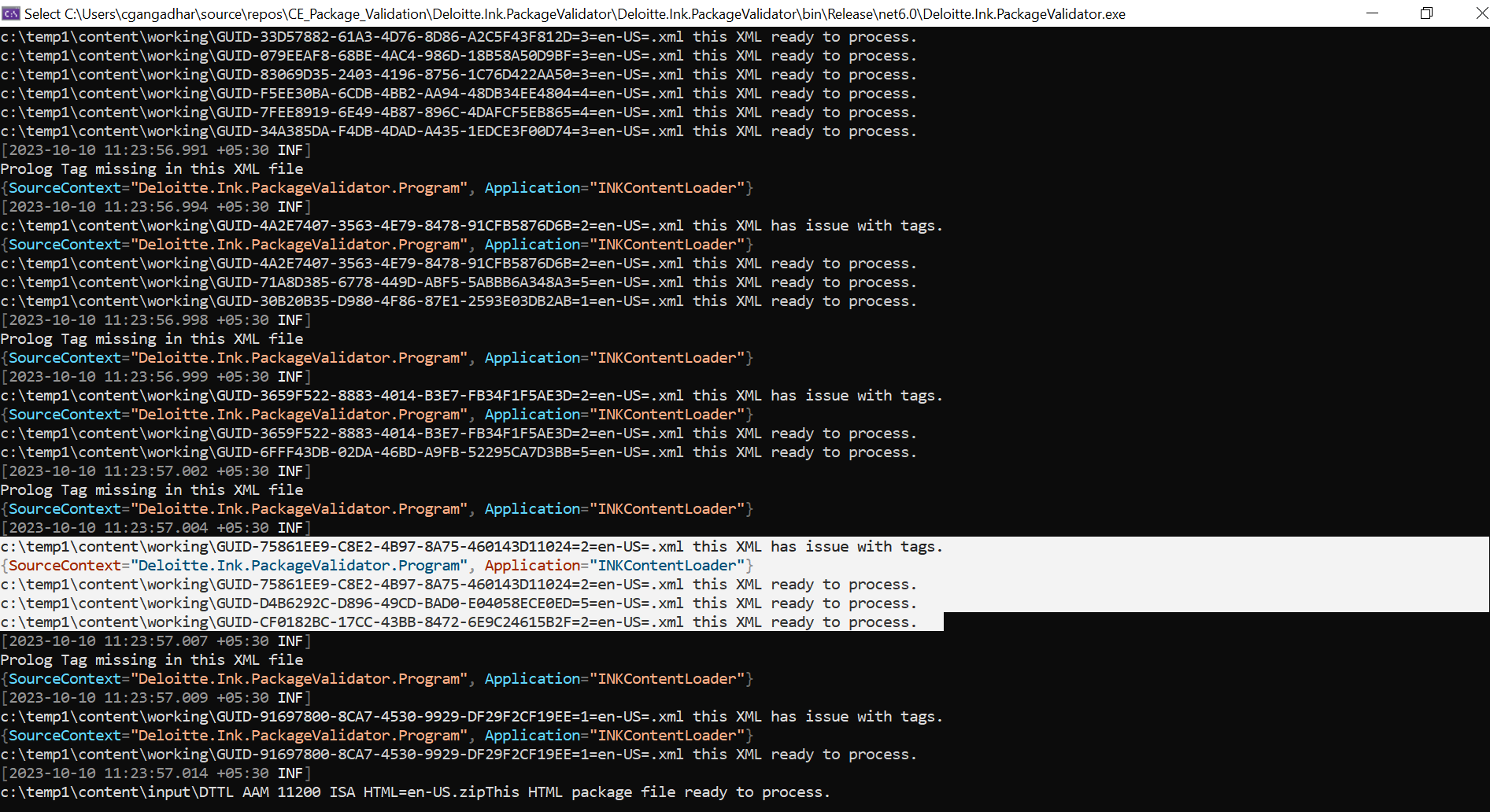


Step 3: Upon starting the process, the build shows as started.

Step 4: The console is loaded separately which indicates the start of the package validation process.

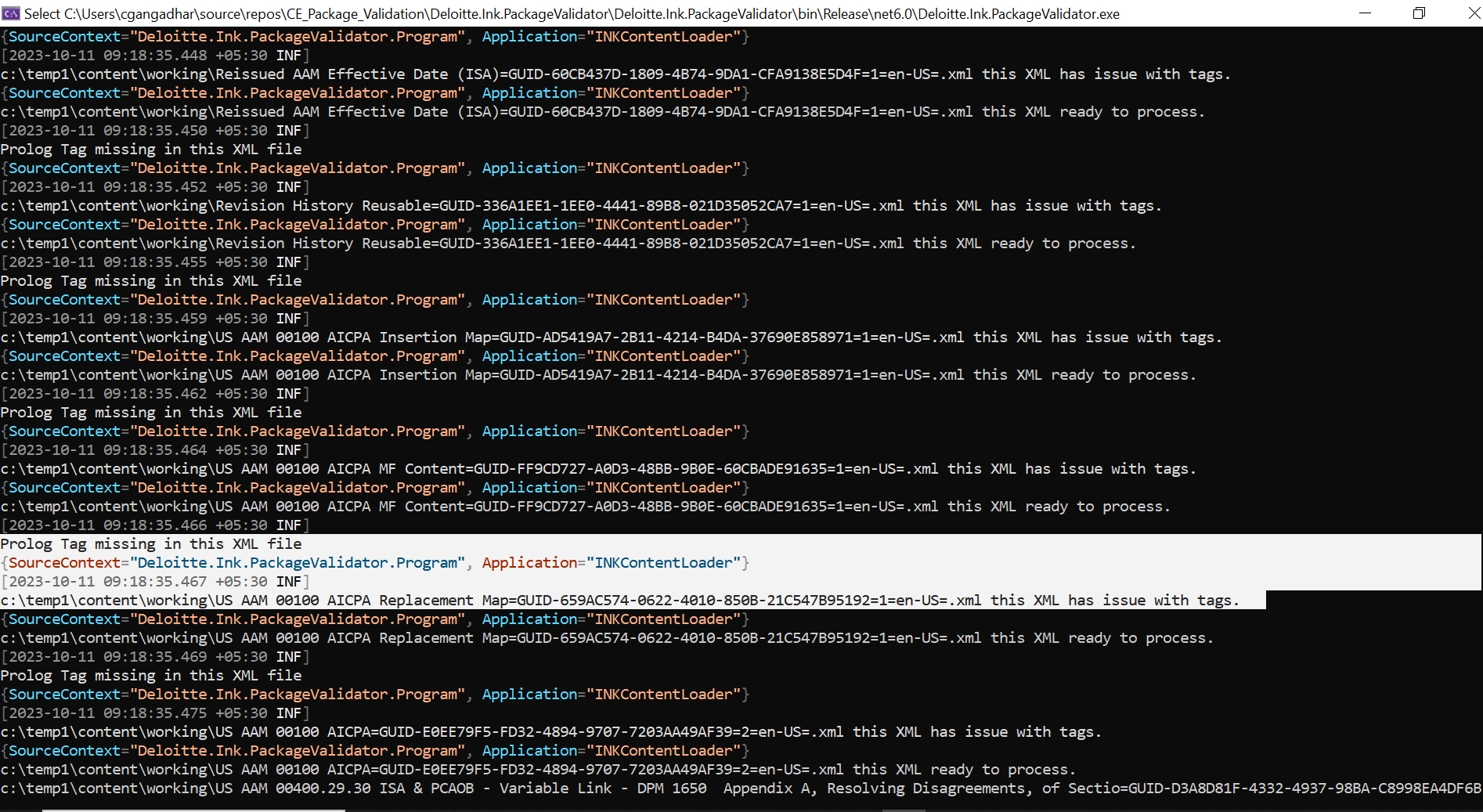


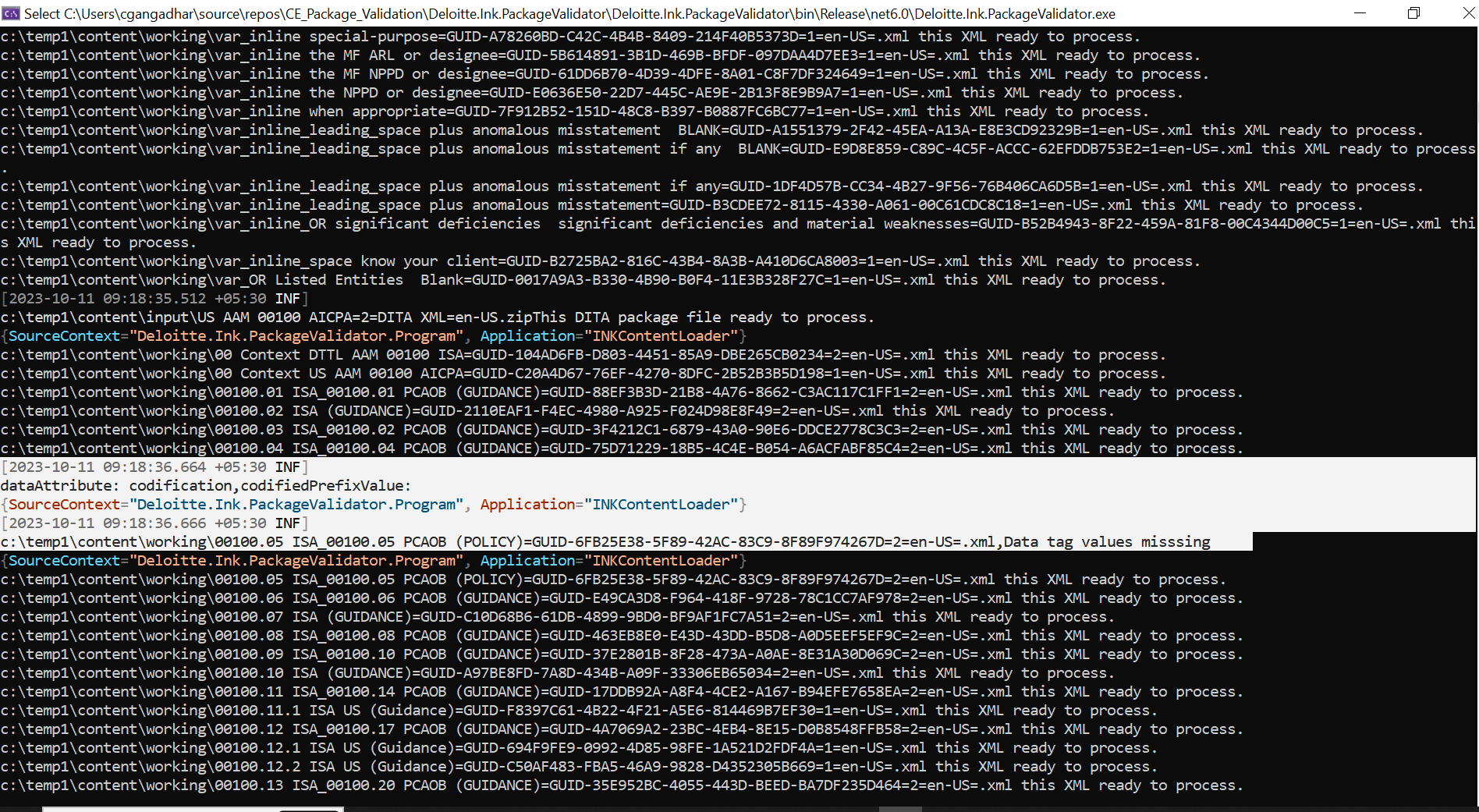
Step 5: Each and every XML & HTML file within each and every individual package set is scanned and then the result of the scan is shown on the respective line.



Step 6: If there is no issue detected with the package the status shows as ‘this XML file ready to process’.

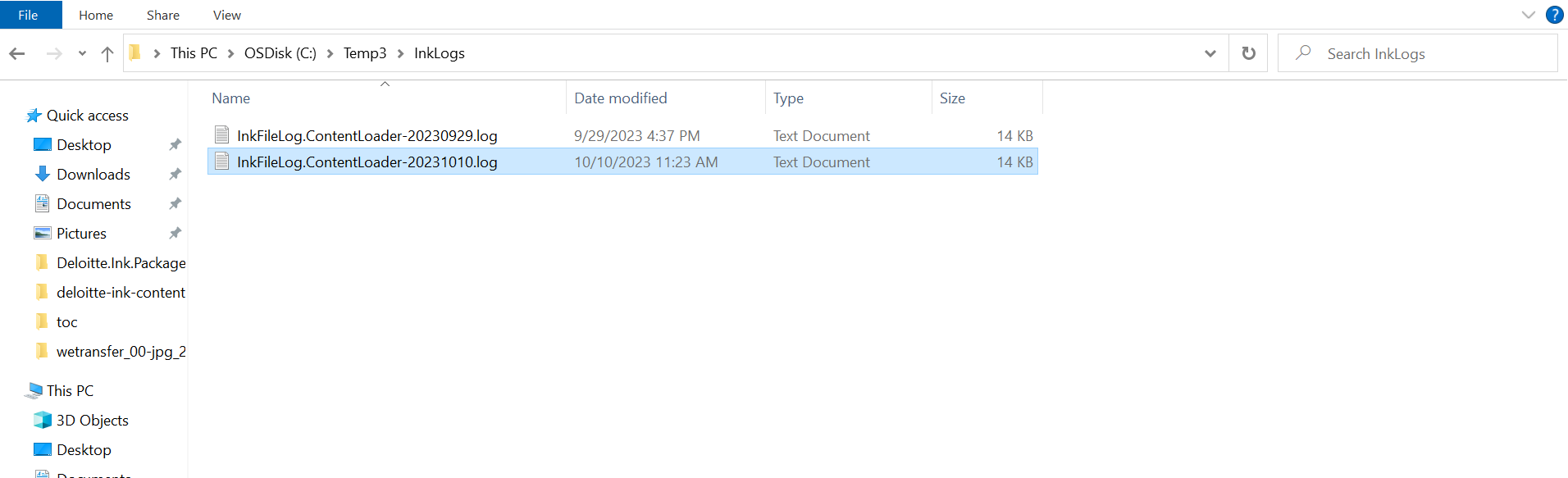
Step 7: This is the error log captured when there is an issue with the Data name Codification value (or) Codifedprefix value.





Step 8: After the end of scanning for all the XML & HTML files within a specific Manual Section package, the end of scanning and the overall result for a specific Manual Section package is indicated as below.

Step 9: The results are then automatically written to a text file and then saved in the local folder as below.



Step 10 : All the scanned results / logs can be seen in a text format in this file when opened as below.



.

# Business Value addition

* Simple to use and easy to share Application.
* Time saving (~60% savings on time spent for package validations)
* Highly reliable and very efficient
* Log based output, easy to share and track with the stakeholders.

# Conclusion

The Content Validator application is the right fit for the requirement of InK where we deal with the huge amounts of data packages and the turnaround time in validating these before they are passed on to the various consumption systems is of very high importance.