Contents

[Documentation for Automating the Extraction of Attribute Values from Workspace for Automation Testing 3](#_Toc179550254)

[Overview 3](#_Toc179550255)

[Purpose 3](#_Toc179550256)

[Project Structure 3](#_Toc179550257)

[1. DataExtractionNamespace 3](#_Toc179550258)

[Main Workflow 4](#_Toc179550259)

[Key Files Generated 5](#_Toc179550260)

[Instructions for Use 5](#_Toc179550261)

[Prerequisites 5](#_Toc179550262)

[Steps 5](#_Toc179550263)

[Further Enhancements 6](#_Toc179550264)

Revision History

|  |  |  |
| --- | --- | --- |
| Date | Version | Author |
| October 2024 | 1.0 | Juan Manuel Gutiérrez García |

# Documentation for Automating the Extraction of Attribute Values from Workspace for Automation Testing

## Overview

This project provides a tool that automates the extraction of equipment attributes from a workspace, stores the data in CSV format, and compares it with a predefined baseline Excel file. The tool outputs the comparison results, along with a test summary in both CSV and text formats, helping identify discrepancies in equipment attributes.

## Purpose

The tool helps ensure the accuracy of equipment attributes by comparing extracted data from a workspace against predefined baseline values.

## Project Structure

DataExtractionNamespace  
The core logic for data extraction and file comparison is implemented in this namespace. It contains the following classes:

* + **TestCaseInfo**  
    Stores metadata about each test case, including:
    - Sheet name in the baseline Excel file
    - Object ID (Oid)
    - Equipment class name
  + **TestCases**  
    A dictionary that maps test case IDs to equipment object IDs (Oid) and corresponding class names. This acts as a reference for all equipment being tested.
  + **Utilis**  
    A utility class with helper methods for workspace interaction and data processing:
    - GetFilteredTestCases: Filters and returns test cases based on Environment.titleToTestId.
    - OpenWorkspace: Establishes a connection to the workspace.
    - ReadCsv & ReadExcel: Methods to read extracted and baseline data from CSV and Excel files, respectively.
    - CompareFiles: Compares the extracted data against the baseline Excel data and logs discrepancies.
    - CreateGeneralTestSummary: Generates a summary report of the comparison results.
  + **Environment**  
    A static class storing global constants such as:
    - WorkspaceName and BusinessServer: Connection details for the workspace.
    - pathToStoreFiles: Directory where extracted data and comparison results are saved.
    - TitleToTestId: A dictionary mapping equipment test titles to their respective test IDs.

## Main Workflow

1. **Open Workspace**:  
   The Utilis.OpenWorkspace method connects to the workspace using details stored in the Environment class and the AZClientTools.dll.
2. **Extract Data**:  
   For each test case, the tool retrieves equipment attributes using the ReadViewEx method from the AZCMappingSvcs.dll. The extracted data is stored in a CSV file (extracted\_data.csv).
3. **Compare with Baseline**:
   * The extracted CSV file is compared with the baseline Excel file.
   * Discrepancies between attribute values are logged in separate CSV files for each test case.
   * A test summary (TestSummary.txt) records the total number of failed comparisons.

## Key Files Generated

* **Extracted\_data.csv**: Contains equipment attributes extracted from the workspace in the following format: | Oid | SheetName | ClassName | AttributeName | Unit | Value in Workspace |
* **TestResults/<SheetName>-attribute\_test.csv**: Contains comparison results for each equipment test in this format: | Value in Workspace | Baseline Value | IsSame (✔ if match, ✖ if not) |
* **TestSummary.txt**: A summary report listing the total number of failed comparisons for each test.

## Instructions for Use

### Prerequisites

1. Ensure that the workspace is populated with attribute values as detailed in the Excel file [Simulation Mapper - A+\_Updated.xlsx](https://aspentech-alm.visualstudio.com/AspenTech/_git/AZyqad_Product?path=/V10%20New%20feature%20test%20case/Simulation%20Mapper/Simulation%20Mapper%20-%20A%2B&version=GBmaster) (for A+) or [Simulation Mapper - Hysys - Repos](https://aspentech-alm.visualstudio.com/AspenTech/_git/AZyqad_Product?path=/V10%20New%20feature%20test%20case/Simulation%20Mapper/Simulation%20Mapper%20-%20Hysys&version=GBmaster) (for HYSYS).
2. Check the [OIDs Template.xlsx - Repos](https://aspentech-alm.visualstudio.com/AspenTech/_git/AZyqad_Product?path=/V10%20New%20feature%20test%20case/Simulation%20Mapper/data_transfer/OIDs%20Template.xlsx&version=GBmaster) file to see how the OIDs for each Test should look like
3. Omit any manual comparison, as it will be handled automatically during project execution.

### Steps

1. **Setup Workspace**:
   * Verify that the workspace and the variables in the Environment class are correctly configured for your environment.
   * Modify pathToStoreFiles to point to a valid directory where the baseline Excel file and extracted data will be stored.
2. **Run the Application**:
   * Ensure that the baseline Excel file is available in the specified path.
   * Build and run the DataExtraction1 project to initiate the extraction process.
3. **Review Results**:
   * Each test case will have a corresponding CSV file in the TestResults folder, indicating pass or fail for each attribute comparison.
   * The final summary, TestSummary.txt, provides an overview of the failed test cases and the total number of discrepancies.

## Further Enhancements

* Improve error handling for workspace connection failures.
* Enable dynamic configuration of test cases to avoid hardcoded values in the TestCases class.