**Coding Best Practices**

**For**

**Elanco CRDB – Test Automation**

eE

**Important Notice:** This document and any attachments submitted with it are confidential and intended solely for the use of the individual or entity to which it is addressed. If you are not the intended recipient, you are hereby notified that any use, dissemination, forwarding, printing, copying or other use of this document, any attached documents, and the information these documents contain, is strictly prohibited. If you have received this by mistake, please advise the sender immediately and destroy / invalidate the document.

**Prepared for**



**Submitted By**



**WinWire Technologies Inc.**

**2350 Mission College Blvd.**

**Suite 925, Mission City Center**

**Santa Clara, CA, 95054**

[**Info@WinWire.com**](mailto:Info@WinWire.com)

**Document Information**

|  |  |
| --- | --- |
| Document Title | Coding Best Practices |
| **File Name** | Elanco\_CRDB – Automation Coding Best Practices.docx |
| **Author(s)** |  |

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changed By | Items Changed Since Previous Version |
| 0.1 | 8/19/30 | Sujith Reddy | Draft Version |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1. Naming Conventions 4](#_Toc49789164)

[2. Layout Conventions 4](#_Toc49789165)

[3. Commenting Conventions 4](#_Toc49789166)

[4. Language Guidelines 4](#_Toc49789167)

[5. Test Data 5](#_Toc49789168)

[6. Others 5](#_Toc49789169)

## 1. Naming Conventions

1. Choose English-readable identifier names.
2. Use PascalCasing for class and method names.
3. Always prefix interface names with I followed by text in PascalCasing.
4. Use camelCasing for method parameters and local variables.
5. Use singular names for enums.
6. Do not suffix enum name with word Enum.
7. Avoid using abbreviations or acronyms.
8. Do not use hyphens, underscores or any nonalphanumeric characters (except static variables, which can have underscore as prefix followed by camelCasing).
9. Do not use all caps.
10. Do not use Hungarian notation (e.g. nCount).

## 2. Layout Conventions

1. Write only one statement per line.
2. Write only one declaration per line.
3. Keep the code properly indented, so that it is easily readable.
4. Add one blank line between property definitions and method definitions.
5. Always use parantheses to make clauses in expression apparent (e.g. if statements).
6. Declare all member variables at the top of the class, with static variables at the very top.

## 3. Commenting Conventions

1. Place the comment on a separate line, not at the end of a line of code.
2. Begin comment text with an uppercase letter.
3. End comment text with a period.
4. Insert one space between the comment delimiter (//) and the comment text.
5. Do not create formatted blocks of asterisks around comments.
6. Split longer comments into multiple lines so that reader does not need to scroll.
7. Avoid obsolete comments.
8. Remove unused/dead code.

## 4. Language Guidelines

1. Use predefined type names (int, string, etc.) instead of system type names (Int16, String, etc.).
2. Use **+** operator to concatenate short strings, use **StringBuilder** to concatenate larger text in loops.
3. Use implicit typing for local variables when the type of the variable is obvious from the right side of the assignment, or when the precise type is not important.
4. Use implicit typing to determine the type of the loop variable in **for** and **foreach** loops.
5. Do not use **var** when the type is not apparent from the right side of the assignment.
6. Use a **try-catch** statement for most exception handling.
7. Use **using** statements to avoid explicitly disposing objects, e.g. **SqlConnection**.
8. Use object initializers to simplify object creation.
9. Keep methods short, not longer than 40 lines of code, refactor if required.
10. Avoid too many parameters in methods, not more than 5 or 6.
11. Avoid complex expressions.
12. Minimize the number of **return** statements in a method, for better readability.

## 5. Test Data

1. Always try to place the test data sheet inside Test Data Folder.
2. Make Sure Test data sheet is aligned properly and cells inside the sheet is not missed.
3. Always save the data to Excel workbook and then to XML before running the script
4. Use Standard Wait time across the Sheet for every step. Ex: 3000.
5. Environment Set up / Test Run set up should be done and saved before running the Test

## 6. Others

1. A module should never be altered other than the one where it is defined
2. Use a common external file
3. All DOM manipulation should be done inside directives. Directives should be kept small and use composition.
4. Rework the default directory structure to group your project by logical sections.
5. Always use Relative Xpath for locators if possible, try to add 3rd party tools as extensions for finding xpath
6. If we want to add new user actions, always create a new method