WEB FLOWS DEVELOPMENT

MUST HAVE CHECKLIST AND GUIDELINE

Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Date | Author | Details |
| 1.0 | 10/11/2017 | Valentin | Initial draft |
| 1.1 | 10/11/2017 | Team | After review |
| 1.2 |  |  |  |
| 1.3 |  |  |  |
| 1.4 |  |  |  |

Contents

[Preface 4](#_Toc498347022)

[References 4](#_Toc498347023)

[Must have list 5](#_Toc498347024)

[Coding 5](#_Toc498347025)

[Tracing 5](#_Toc498347026)

[Input Interface definition and Validation 5](#_Toc498347027)

[“Delta collection” realization 5](#_Toc498347028)

[QA, PLUS etc. 5](#_Toc498347029)

[Performance 5](#_Toc498347030)

[Nice to have list 7](#_Toc498347031)

# Preface

This document should be considered by WEBFLOW-s developers as a kind of cheat-sheet defining must-be-done list of things that should be done/verified for every WF.

# References

Please, consider the following documents as having a lot of valuable information related to the scope of this document.

[WEB FLOWS DEVELOPMENT GUIDELINE.docx](https://gc/cis/The%20Webint%20site/MS/RnD/Procedures%20and%20documents/Development/WEB%20FLOWS%20DEVELOPMENT%20GUIDELINE.docx?Web=1)

[JS Patterns and practices.docx](https://gc/cis/The%20Webint%20site/_layouts/15/WopiFrame.aspx?sourcedoc=/cis/The%20Webint%20site/MS/RnD/Procedures%20and%20documents/Development/Patterns%20and%20practices%20-%20custom%20actions%20and%20libraries.docx&action=default)

Such aspects as naming conventions, versioning, and deployment related questions and many others are defined in that documents.

# Must have list

## Coding

* Naming convention (ADI – TBD)

* CACommon library usage
  + Input validation module should always be used
  + Needed fixes (Dido TBD)
    - Debug Level (should be on the WF level)
* XPATH library usage
  + XPATH-s are never inlined
  + All data extraction happens via CACommon library, so that not found XPATH-e will be reported in the same way
* CA is never reused between WF
* Library is always reused
  + Library per API
  + Library per DOM
  + Common libraries
* Blocked VA right notification
* Separation of “could be collected without VA” and “could be collected only with VA” parts
* Not too much sophisticated code
* CA should as much as possible follow “Use latest” (library version) approach

## Tracing

* Clear separation between “Production” and “Debug” entry points.
  + Not more than 50-100 production points.
  + All critical situations should be “Error-ed”
* In production mode WF should not have ANY of:
  + Defected rows
  + Error Events
* In production mode WF should have reporting :
  + Main decision points (*Do not have pictures/ max number of comments reached etc.*)
  + Input parameters validation
  + Post-collection statistics (*123 posts had been collected*) TBD

## Input Interface definition and Validation

* Clear and unambiguous definition of all parameters:
  + Input parameters (coming from end-user) should have min/max possible values as far as default value (if this may not be supplied as input)
* Other configuration parameters should be defined in a separate library

All FED related attributes should be specified in IE as far as in WFMS

## “Delta collection” realization

* Every WF should be tested for Delta realization.

## “IsMonitored element

* Always should be registered in case of target like WF

## Performance

* CR should be executed in a reasonable time that will be coordinated with PM.
* Every WF should declare “rough formula” of execution time calculation.

(*E.g. could collect 500 records in 20 min, 100 in 40 min etc.*)

## Algorithms and architecture

* SWF with N instances of the second level thread that will get bulks of M elements to proceed
* VMF – for atomic stateless operations
* As less API as possible
* Controlling number and size of media that should be downloaded

## QA, PLUS etc.

* All data element types should be collected and checked on the daily basis (many small CR-s?) TBD

So that it will be clear that “*Likes are not collected while comments are*”

* Min/Max modes of test.
* Any WF should be always tested E2E

## Documentation

* WFMS based
* CRD
  + One per WP
  + WFMS compliant file name *WP\_CRD.docx*
* VA creating manual
  + if needed
  + one per WP

Quality analysis per existed WF

# Instagram

**What is good?**

* Input validation
* Delta realization
* Scrolling down concept
* Integration with VMF (F/F)

**What is bad?**

* Too complicated code
* Not clear comments and naming convention
* XPATH-s are “inline”-ed
* Not clear concept of “MAX number of the posts”. It is hardcoded now (1000)
* Performance
  + All posts are processed in one thread
* Not found XPATH-s are not reported properly
* Likes (count per post) is not collected
* Locations not seen on the map
* Comments K/V for videos not collected

**Recommended**

* Check that there are no:
  + Defected rows at all
  + Too many events
  + Errors (in event log)
* Kind of “special event-code” for statistics.

Could be used later for PLUS purposes

* Collection of the posrts could be done
  + Either in VMF
  + Or in SWF, where two parameters will be controlled for the fine tuning:
    - Number of parallel threads
    - Length of the collection that will be passed to each rthread (for WHILE-ing)