Handle Validation Failures and Continue Scenario Execution

Problem

As a user, I want to report on validations which fail, but I'd also like to use workaround steps if possible to move past the error when it occurs.

Resolution

Create a validation which also includes the steps to overcome the issue and move forward.

Verification Snippet

```
var visibilityResult = isVisible(small(["Enter valid Date of Birth"]));
if (visibilityResult){    
  setValue(textbox('DateofBirthMask'),'01011955');
};
assertFalse(visibilityResult);
```

Explanation

1. Check if something is visible
2. If it is visible, then proceed with workaround step(s)
3. Then return a verification failure if the item checking for was visible using assertFalse(visibilityResult);

Many other asserts types exist. For a complete list, see Assert and Verify Functions.

Why is the assertion last?

- As a user, you would like to check for the existence of an issue...
- But before sending the result back (where the script or AISG will continue running)
- You would like to recover from the issue and work around it if possible.
- This means you
  - First check for the existence of the issue,
  - And if it exists, handle the issue,
  - And only then after it's handled, return the failure status back to the script.
- If the assertion were first,
  - Then the subsequent steps would be skipped, as a failure result would occur,
  - And the remaining steps would be skipped.

Steps

Option A: Example Using SmartTag Validations

- SmartTag validations are triggered either pre- or post- interacting with associated SmartTag(s)
- In this example, after an action against the **svBtnAddToList** SmartTag, the **failIfValidEffectiveDateMsgExistsAndHandle** validation triggers

For additional information about SmartTag Validations see Validations in SmartTag Workbench.

SmartTag Validation Assignment



SmartTag Validation Post-Action Trigger



SmartTag Validation Usage

This SmartTag automatically triggers when a Test Designer IDE script, JS Edit & Play script or AISG uses the svBtnAddToList SmartTag.

- JavaScript
  `click(svBtnAddToList('Add To List'));`
- Test Designer IDE

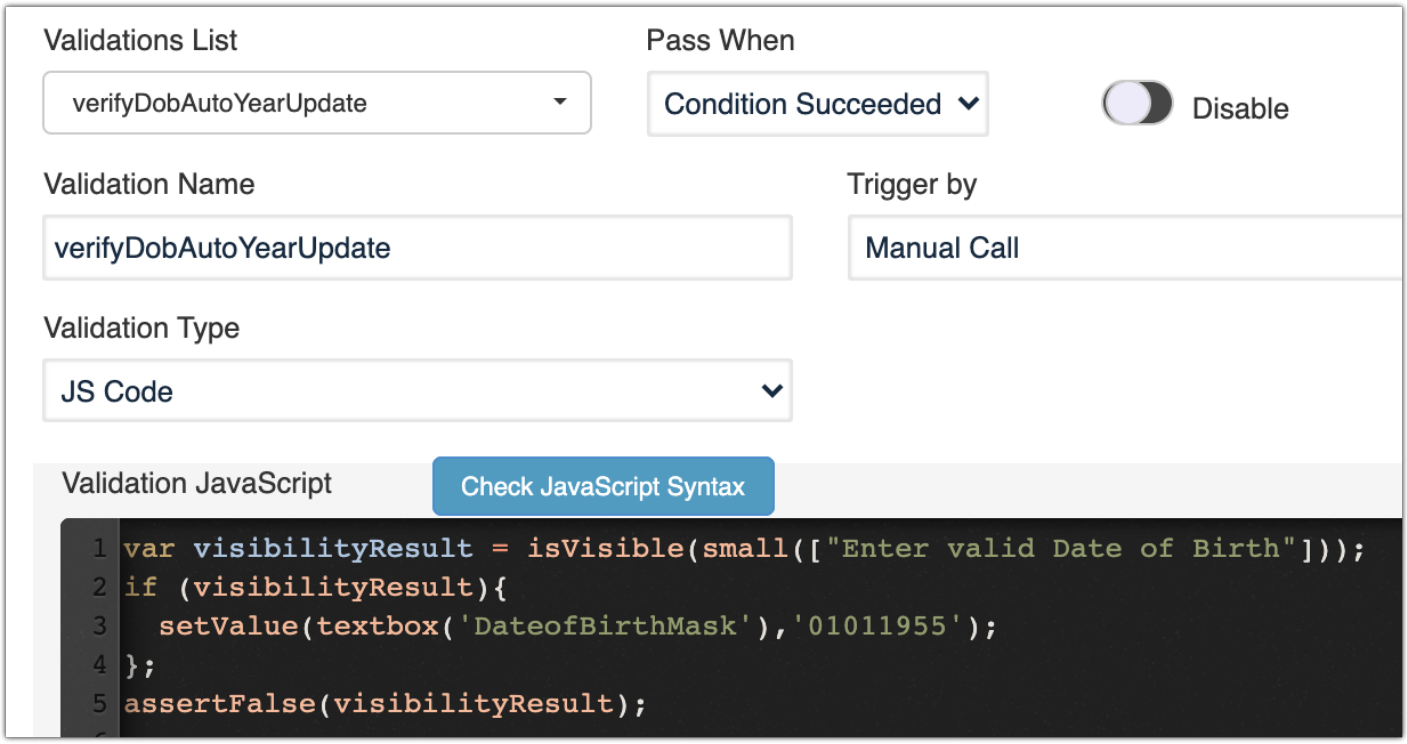
Action	Accessor
Click	svBtnAddToList('Add To List')

- AISG

Type	Name	Variable
Button	105) svBtnAddToList('Add To List')	Click

Option B: Example using Validation Workbench

Validation Configuration



Validation Usage

The **Trigger by** type in this example is a **Manual Call**, allowing you to call the validation from a script or AISG simply using **Validate** as follows:

- JavaScript
  `validate("verifyDobAutoYearUpdate");`
- Test Designer IDE

Action	Accessor
Validate	"verifyDobAutoYearUpdate"

- AISG

Type	Name
Custom	validate("verifyDobAutoYearUpdate");

For additional information about triggers see "Trigger By:" in Validation Workbench

Last updated: August 25, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Revision #8
Created 2026-07-31 15:40:23 UTC by Admin
Updated 2026-11-04 18:26:59 UTC by AI_Mananger