AS - Failure to get dump

On iOS 9 and up (full control mode), in some scenarios, SeeTest may fail to perform actions(i.e open object-spy) on a page that contains a very large number of elements,

In order to resolve this issue you can use one of the following methods:

- Node-based dump (recommended).
- Set dump parameters.
- Dump by UIElements only.
- Dump by grid sampling.

ⓘ Since iOS 12, the use of UIWebView for web elements is deprecated - https://developer.apple.com/documentation/uikit/uiwebview. Since iOS 14, it is no longer supported.

For a suitable alternative to handle your application's web content, please see - https://developer.apple.com/documentation/webkit/replacing_uiwebview_in_your_app.

ⓘ **Instructions for building apps to latest OS:**

When building an application using Xcode, the best practice is to make sure that the Xcode version includes the relevant iOS SDK. The following page contains the relevant information - https://en.wikipedia.org/wiki/Xcode.

If not, Xcode should be upgraded to a later version that does support the relevant iOS SDK and the application re-built, otherwise, there could be **dump misalignment.**

Node-based dump

This ability should be configured per application using its bundle identifier.

In order to configure that ability:

1. Open the **"ios_dump.properties"** found here.
2. Add the line:

   ⓘ <bundle identifier>=ios.non-instrumented.dump.node-based.bundle

   ---

   For example, application with bundle identifier 'com.experitest.eribank' will add the line:

   ⓘ com.experitest.eribank=ios.non-instrumented.dump.node-based.bundle

   ---

3. Restart SeeTest.
4. Deleting the line from the **"ios_dump.properties"** file will use the default dump on the application.
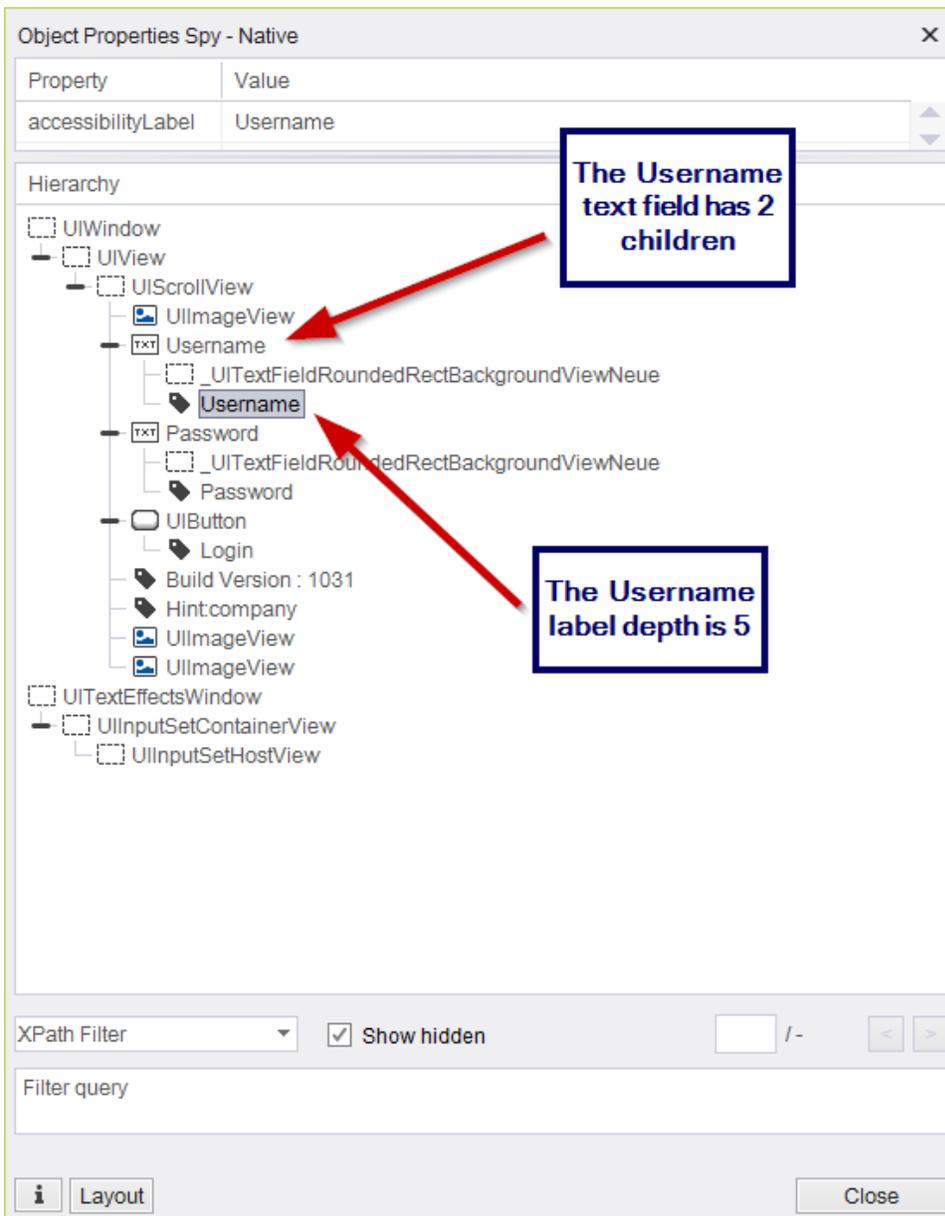

ⓘ By default, the Safari application bundle identifier is found in the "ios_dump.properties" file.

Set Dump Parameters

If the Node-based dump method did not work you can use this method to define specific dump parameters to limit the element's numbers.

there are 3 dump parameters:

- **Depth** - Define the depth of the element hierarchy tree.
- **Total** - Define the total amount of elements in the dump.
- **Children** - For each element (node) what are the max node's successors (children).

To use this method you will need to add to the app.property file the property: **ios.non-instrumented.dump.parameters= <depth>,<total>,<children>** for example:

> (i) ios.non-instrumented.dump.parameters=20,1000,50

You can use this property in run-time (for specific tests) by using the command: setProperty() with the property **ios.non-instrumented.dump.parameters** for example:

> (i) client.setProperty("ios.non-instrumented.dump.parameters", "20,1000,50");

Improvements in the non-instrumented dump

From STA 10.3 the 'Class' property has changed and now is more informative.

To go back to the older dump - add to the app.property file the property: 'xcelement.type.include=false'

Dump by UIElements only :

Create a dump by collecting native UI elements. This can help to avoid crashes in the case of a large web dump.

to use this method, add to app.properties file the following property:

ios.dump.uielements.only_<bundleId>=<fetch.web.anchor=true/false> - for a specific application.

ios.dump.uielements.only.params=<fetch.web.anchor=true/false> - for all applications.

fetch.web.anchor: when set to true, we collect few extra non UIElements to be able to identify web elements.

Dump by grid sampling :

Create dump by sampling grid of points, SamplesX refers to how many samples on X-Axis when a device is in portrait orientation when in Landscape the samples on each axis will switch.

We recommend try first **default** value which is 5,12,5,true,false and if it does not create the expected dump only then try other values.

to use this method add to app.properties the following property:

ios.dump.grid.sampling_<bundleId>=<SamplesX>,<SamplesY>,<Depth>,<Hybrid(optional)>,<FrontMostAppFocused(optional)>

optional parameters:
Hybrid: combines the grid sampling and the traditional dump method, which can improve performance.
FrontMostAppFocused: ignore results from elements that belong to other pids, such widgets, _UIRemoteView elements, and other apps.