How To Use Regex to Define the Domain(s) Which Inject WalkMe

General Guidelines

Regex should be as specific as possible, but also general enough to work on different pages. To achieve this balance, use customer-specific prefixes.

- **Example**:
  - **URL:** walkme.my.salesforce.com
  - **Regex:** .*walkme.my.salesforce.com.* instead of .*salesforce.com.*

The Regex for each environment **must** be different. Prefixes also come in handy here.

- **Example**:
  - **Production URL:** na3.salesforce.com
  - **Sandbox URL:** cs23.salesforce.com
  - **Production Regex:** .*na.*.salesforce.com.*
  - **Sandbox Regex:** .*cs.*.salesforce.com.*

Begin and end every string of regex with .*

- **Example**: .*salesforce.com.*

Instead of using digits, use .*. Replace any numbers in a URL with .*

- **Example**:
  - **URL:** cs23.salesforce.com
  - **Regex:** .*cs.*.salesforce.com.*

Make sure to put a \ before every period.

- **Example**: .*salesforce\.com.*

No need to escape hyphens.

- **Example**:
  - **Don't use**: .*panasonic\-\-full\.salesforce\.com.*
  - **Do use**: .*panasonic–full\.salesforce\.com.*)

To include a URL with two or more different prefixes use (first|second|third|etc.) where the prefixes

occur.

- **Example**:
  - **URLs:** walkme.salesforce.com and sandbox.salesforce.com
  - **Regex:** .*(walkme|sandbox).salesforce.com.*)

For regex to exclude a word, refer to the below example, where you want to create an extension with one editor account on Salesforce Classic and one on Salesforce Console.

- **Example**:
  - **Salesforce classic regex:** .*cs.*.salesforce.com(?!console).*
    - **Note:** This (?!console) excludes the word console from the regex.
  - **Salesforce console regex:** .*cs.*.salesforce.com/console.*

To include more than one regex in a package, place an | in between each regex with no spaces.

- **Example**: .*cs.*.salesforce.com.*|.*cs.*.visual.force.com.*

Make sure to never leave two consecutive || in a package, as this will cause a allow list and match to all URLs and introduce errors on certain sites.

Regex is case sensitive. Be sure to match the correct case of the URL within the regex.

Stricter Domain Regex

There may be a scenario where you need to "tighten" the regex to exclude a specific frame. For instance, you may only want WalkMe to inject in a specific iFrame, but that iFrame's domain is included in many other iFrames as a reference.

Example of iFrame domain you want WalkMe to appear in:

- https://visual.force.com/1000

Example of iFrame domain that we are trying to prevent WalkMe from injecting into:

- salesforce.com/referrer?=https://visual.force.com/1000

To specify that regex that must come at the beginning of the URL, we use:

- ^https?://[^/]*(visual.force).com

Using, this WalkMe will only inject into the https://visual.force.com iFrame if its the first part of the URL.

Platform-Specific Regex Tips

There are a number of extensions we are often asked to configure that are standard.

Salesforce

It is extremely important to build Salesforce extensions carefully and to test them in a number of test scenarios due to the variance in the platform across pages and the presence of many analytics iFrames that take up 0px.

Creating the extension

- Use negative look-ahead regex to avoid the session server and other unnecessary injections: (?!/(login/sessionserver|s.gif)
- Using “Or” logic within the package is more efficient than creating two separate extensions
- “na” can be sometimes “ap,” “eu,” or “cs” depending on the Salesforce server geographic location, but it generally doesn’t change, so stick with the URL you have
    - nameansNorthAmerica,apmeansAsia-Pacific,andeumeansEurope, Middle East or Africa.csrepresents a sandbox environment
    - The number next to the 2-letter code indicates the server instance in the assigned country. This can change when uploading a new environment or refreshing an existing one
    - You can see the Salesforce servers’ health status at http://trust.salesforce.com/trust/instances

Testing the extension

- Ask the CSM to provide a “heavy” page with lots of content or alternatively, an important page in the implementation and make sure it works well there
- Verify the extension isn’t injecting into too many iFrames using the Master Extension Diagnostics Tool

Note

When building/testing in Salesforce Lightning, use a new extension without the “walkme_” component; for example, walkmecs will work and not walkme_cs will not.

Test

Website	Example URLs	Suggested Regex

Salesforce Classic	https://panasonic-fullcopy.cs24.my.salesforce.com/ https://panasonic-fullcopy-field-trip.cs24.visual.force.com/	.*customername.*.cs.*.my.salesforce.com.*|.*cs.*.salesforce.com/(?!console).*|.*cs.*.visual.force.com.*
Salesforce Lightning	https://cs24.lightning.force.com/	.*cs.*.lightning.force.com.*
Salesforce Console	https://cs6.salesforce.com/console?tsid=02u50000000e56v	.*cs.*.salesforce.com/console.*

Production

Website	Example URLs	Suggested Regex
Salesforce Classic	https://panasonic.my.salesforce.com/home/home.jsp https://panasonic-c.na34.visual.force.com/	.*customername.my.salesforce.com.*|.*na.*.salesforce.com/(?!console).*|.*na.*.visual.force.com.*
Salesforce Lightning	https://na34.lightning.force.com/	.*na.*.lightning.force.com.*
Salesforce Console	https://na2.salesforce.com/console?tsid=02u50000000e56v	.*na.*.salesforce.com/console.*

In the above examples, you'll notice we added (?!console) to the Salesforce Classic regex. Since these packages are within the same extension and the Classic and Console URLs are fairly similar, we don't want WalkMe to get confused and load WalkMe where it shouldn't load (or not load at all!). The only difference between one of the Classic and Console URLs is that the Console URL has "/console" at the end. So, we add the (?!console) to the Salesforce Classic regex to ensure it does not load any content meant for Salesforce Console.

Workday®

Test

Example URL	Suggested Regex
impl.workday.com/tripadvisor/d/home.htmld	.*impl.workday.com/customername.*

Production

Example URL	Suggested Regex
https://www.myworkday.com/sunbeltrentals/login.flex	.*myworkday.com/customername.*

SuccessFactors

Test

Example URL	Suggested Regex

https://hcm4preview.sapsf.com/	.*hcm.*preview.sapsf.com.*

Production

Example URL	Suggested Regex
https://performancemanager5.successfactors.com/	.*performancemanager.*.successfactors.com.*

Note

SuccessFactors URLs don't have a constant pattern. Production and test can be practically identical, for example: https://performancemanager5.successfactors.eu.
If that's the case, use different extensions with the same regex.

Distinguishing between similar but critically differentiated URLs

In configuring an extension, you will often need multiple packages for different platforms, for example, one for normal Salesforce (SFDC) and another for Service Cloud. In such cases, your browser extension will utilize regex to differentiate between platforms and inject different snippets into these different platforms as appropriate. In the SFDC Service Cloud hypothetical, we could use regex to differentiate between these platforms as follows:

Normal SFDC URL:

- https://cs2.salesforce.com/home/home.jsp?tsid=02u0000000000hV

Normal Service Cloud URL:

- https://cs2.salesforce.com/console?tsid=02u50000000e56v

Regex statement for the extension to capture any Service Cloud URL:

- **.*cs2.salesforce.com/console.***

This hones in on the appearance of the term "console" immediately after the salesforce.com domain.

Because a clear difference between the Service Cloud URL and the SFDC URL is the appearance of the word "console" immediately after "salesforce.com", we can use this to our advantage.

Using the regex symbols "?!", we can single out all instances of "salesforce.com" that are not

followed by the word “console”, and, given our two base URLs, we know these will be SFDC URLs.

SFDC regex:

- **.*cs2.salesforce.com/(?!console).***

Here, “?!console” means any URL possessing the items in this regex, where “console” does NOT appear in this position in the expression.

This will not match https://cs2.salesforce.com/console?tsid=02u50000000e56v, but it will match https://cs2.salesforce.com/home/home.jsp?tsid=02u0000000000hV.se