Quick Guide: Understanding the Syndicated Conditional Chat Link widget (SCCL)

This document is specifically for those deploying the Syndicated Conditional Chat Link directly on their external web pages.

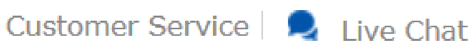
This document provides helpful insight into the Syndicated Conditional Chat Link widget and answers the following questions:

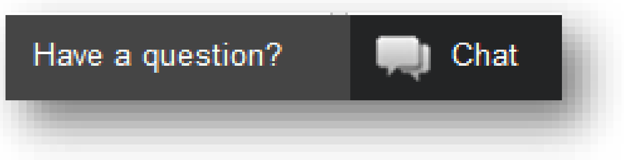
- *What is a Syndicated Conditional Chat Link widget?*
- *Where would I place the Syndicated Conditional Chat Link on my website?*
- *Are there any requirements for deploying the Syndicated Conditional Chat Link?*
- *Do you have any hints for configuring the available Syndicated Conditional Chat Link options?*
- *Where can I go for more information?*

What is a Syndicated Conditional Chat Link widget?

The Syndicated Conditional Chat Link widget (SCCL) is a link to chat that is deployed on an external web page when defined conditions are met.

Common ways to display the SCCL include displaying a chat button, a chat link, or a combination of both:





Most often, the conditions associated with the SCCL, (these conditions must be true in order for the chat button or link to show), are:

- The operating hours of the contact center
- Agent availability
- Wait time

Where would I place the SCCL on my website?

Companies tend to use chat as an alternative communication channel to a phone call, so for that reason the option to chat is often placed next to a phone number on a website (or altogether replaces the phone number). Placement of the

SCCL tends to be in a universal header or footer of the site, also common on the *Contact Us* page, or where a phone number and email link are provided.

Common use cases for the SCCL are:

- Chat button or link on the banner of the web site
- Chat button or link across the bottom of the web site
- Chat button or link on the *Product* or *Contact Us* pages
- Replace phone numbers with chat buttons
- Alongside the option to send an email

Are there any requirements for deploying the SCCL?

- **Chat:** Oracle Service Cloud Chat must be enabled for your CX site
- **External Web Pages:** The SCCL is only available for use on non-customer portal pages, so it can be deployed on your company's website. For example:
  - The SCCL **can** be deployed on *//www.oracle.com*
  - The SCCL *cannot* be deployed on *//custhelp.oracle.com*

Do you have any hints for configuring the available SCCL options?

There are many attributes that can be configured within the SCCL widget. A brief explanation of each attribute is provided, as well as helpful tips.

- c – The ID of the CX category to use. Only a single category may be used with chat. Category is used in conjunction with Product for chat advanced routing and reporting. Even if the chat pre-routing is not used for this widget (i.e. visitor input of Product/Category from the chat login page is used for routing), define the Product/Category for the SCCL widget so they can be used for the chat reporting.

- chat_login_page – When an end-user opts to chat by clicking the chat button or link, the chat login page is displayed to the end-user. If this field is left blank, the default chat login page (*/app/chat/chat_launch)* is used. If you prefer to bypass the chat login page, meaning you do not want the end-user to fill out the form and you are fine with taking anonymous chats, set this attribute to */app/chat/chat_landing*. If the chat login page is not used, after the end-user opts to chat they are immediately placed into the chat queue. The product/category defined for the widget is passed to the chat login page and used for routing.

- chat_login_page_height – The height, in pixels, of the window to display. If open_in_new_window is set to false, setting this value has no effect. The default is 686 pixels.

- chat_login_page_width – The width, in pixels, of the window to display. If open_in_new_window is set to false, setting this value has no effect. The default is 700 pixels.

- container_element_id – The ID of the HTML element on the page that represents the container for the entire widget. This is not a required attribute. If present, a CSS class rn_ConditionalChatLink will be added to it. Customers may customize the css for this element to use their own look and feel. If a value is not entered, this field sends through a default value of myChatLinkContainer.

- custom_fields – This attribute should be used if you want a default value for a custom field sent in with the chat request. It is a JSON parsed array of custom fields. The array should be in the form of a key-value pair where the key is the custom field ID and the value is the value of the custom field. Ex: {"ID1":"VALUE1","ID2":"VALUE2"}.

  custom_fields: "{\"26\":\"1234\",\"20\":\"Test\",\"18\":\"2\"}", where "26", "20", and "18" are the IDs of the custom fields.

  Custom fields are defined in the CX console and then visible for chat. Chat visibility needs to be checked to become active.

  Real World Implementation Example: You may wish to present the agent with the value of the consumers shopping cart or other information within the incident or chat pane.

- enable_availability_check – Determines whether to check for agent session availability to display the SCCL. If set to *Yes*, the SCCL will display only after the agent session availability is evaluated. The min_sessions_avail option is described below and used in this evaluation. If a value is not selected the default value of *Yes* is used. For example:

  - If set to *Yes*, the available agent sessions are evaluated and the chat link displays if the defined number of min_sessions_avail is met

  - If set to *No*, the option to chat is available at all times regardless of agent availability or hours of operation. If there are no agents available to handle the session, the chat waits in the queue until an agent becomes available. You can inform the end-user of expected wait time by using label_available_with_wait_template. The option for wait_threshold can also be used to control whether chat is offered. The label_default Label is used when this option is set to No

- enable_polling – If *Yes*, the widget will continually check for agent session availability, after the initial page has loaded. The default value of *Yes* will be used if a value is not selected. Customers should have polling enabled if they want to have the SCCL widget be available to customers as much as possible when agents are available or based on the widget attributes for their particular implementation. If the customer does not want the widget changing from showing chat is available and then not available (or vice versa) while the end user is viewing a page, then polling should not be enabled.

  With the August 2015 version of CX, optimization has been done to limit the amount of polling by the widget. Now the amount of time the visitor spends on the page and the wait times are considered before the SCCL starts to throttle. The polling evaluation will occur every twelve seconds for the first 5 minutes, then once per minute after that. Expected wait time is not used in the polling evaluation here.

When using the enable_availability_check, the SCCL might not be available to a visitor with the initial page load. Enabling polling would repeat the agent session availability evaluation at intervals and then display the SCCL if the minimum agent sessions were to become available during the visitor's time on the page and providing access to chat. If polling is not enabled in this use case, the SCCL will be only evaluated for agent session availability at page load and not present even if during the visitor's time on the page the minimum agent sessions became available.

If not using enable_availability_check option, the polling option is not used and should be disabled and *Minimum Sessions Available* set to 0

- ignore_preroute – This attribute is used to specify whether or not to ignore the chat session pre-routing for this widget. The default value of *False* is used if a value is not selected.

    - When this attribute is set to ***False***, it means the widget will determine the appropriate chat queue based upon agent availability for that queue and chat routing rules. Product/Category definitions for the widget will be used for advanced routing. Any information supplied by the visitor on the *chat_launch* page will not be used to determine routing of the chat.

    - When this attribute is set to ***True***, it means the widget will determine the appropriate chat queue and agent availability for that queue based on the information available to the widget and the chat routing rules. However, Product/Category information supplied by the visitor on the *chat_launch* page will then be used to determine the chat queue based on the chat routing rules.

    - For Example: Your chat routing rule is defined to send chats to Queue 1 if Product = Lawnmower, send chats to Queue 2 if Product = Outdoor Furniture (else, send to Queue 3).

        - If Ignore Pre-Route is set to ***False*** and the product is set on the widget t*o Lawnmowers*, even if the shopper changes the product field on the chat_launch page t*o Outdoor Furniture*, the chat will still route to th*e Lawnmower* queue (Queue 1).

        - If Ignore Pre-Route is set to ***True*** and the product is set on the widget to *Lawnmowers*, if the shopper changes the product field on the chat_launch page to *Outdoor Furniture*, the chat will route to the *Outdoor Furniture* queue (Queue 2).The important thing to note is that if there are available agents servicing Queue 1, the SCCL will display that there are agents available without any wait. However, Queue 2 may have a wait time so the end user could end up having to wait in queue unexpectedly.

- info_element_id – The ID of the HTML element on the page that represents the information that should be displayed for the widget. The information for this element may be served from one of the *label_xx* attributes. This is not a required attribute and information will be displayed only when the element exists on the page. The default value is *myChatLinkInfo* which will be used if a value is not specified. If the service is available but there is a wait, then it would display something like:

> Chat Now
> An agent should be available in 0:23 (0 Minutes and 23 Seconds)

The element can be separated from the link element so that it does not assume the styling of its parent and display like a hyperlink. Example:

> Chat Now
> An agent should be available in 0:23 (0 Minutes and 23 seconds)

- instance_id – Must be a unique id for each widget on the page. This identifies the widget so that server calls related to polling will know which widget to return the response to. The default setting for the widget instance_id is "sccl_0" but you can customize this. A suggested standard is to use "sccl_" as a prefix for your Syndicated Conditional Chat Links, and increment the widget number thereafter.

- label_available_immediately_template – Specifies the label to display when chat is immediately available, with no wait. The default value is "*Agents are available with no wait time"* and is used if no value is specified.

- label_available_with_wait_template – Specifies the label to display when chat is available, but there is a wait. The default value is "*An agent should be available in {TIME} {NUM_MINUTES} {MINUTES} {NUM_SECONDS} {SECONDS}"* and is used if no value is specified.
  - Usable macros include:
    - {TIME}: Displays minutes:seconds
    - {NUM_MINUTES}: Displays number of minutes
    - {MINUTES}: Displays minute or minutes label
    - {NUM_SECONDS}: Displays number of seconds
    - {SECONDS}: Displays second or seconds label

  Note: The default would display as "*An Agent should be available in 02:00 2 minutes 0 seconds*", this value should be modified to choose either time, or X minutes and X seconds.

Real World Implementation Example:
An agent should be available in {TIME} ({NUM_MINUTES} {MINUTES} and {NUM_SECONDS} {SECONDS}).

This would display to the end user as:
An agent should be available in 1:40 (1 Minutes and 40 Seconds).

- label_default – Specifies the label to display when enable_availability_check is set to *No*, or if an error is encountered while fetching the chat availability information. The default value is “*Chat directly with a member of our support team”* and is used if no value is specified.

- label_unavailable_busy_template – Specifies the label to display when chat is unavailable due to agent session or wait time conditions. There are macros that can be used to include the estimated wait time. Please reference “label_available_with_wait_template” for more information.

- label_unavailable_hours – Specifies the label to display when chat is outside of operating hours. The default value is “*We're sorry, chat is currently unavailable”* and is used if no value is specified. Chat hours are set within the CX interface.

- link_element_id – The Link Element ID is a required attribute. This attribute is used in the tag for the widget shown below the image. This tag needs to be placed on the page where you would like the widget to be displayed. If you have multiple SCCL widgets being displayed on a page, then the link_element_id must be unique for each widget. The recommended value is *myChatLink*. Example of the recommended DIV’s for the widget:

```
<div id="myChatLinkContainer">
    <div id="myChatLink">
        <div id="myChatLinkInfo">
        </div>
    </div>
</div>
```

- min_sessions_avail – This option is used in conjunction with the enable_availability_check and specifies the minimum number of open agent sessions that must be available in order for the link to be actionable. If this option is to be used, it must be set to a value of 1 or greater. If set to 0, the trigger is disabled which also disables the agent availability check and polling options. The *Minimum Available Sessions* will always override *Wait Threshold*. Meaning that if a value is set for the minimum number of sessions that need to be available, even if the current wait threshold is met, the chat will not be displayed until the minimum available sessions is also met.

- open_in_new_window – If set to *Yes*, the page defined by *chat_login_page* will be displayed in a new window. If *No*, the current window will navigate to that page. Chat window height/width pixels specified will not be used if set to *No*. The default value is *Yes* and is used if a value is not specified.

- p – The ID of the CX *Product* to use. Only a single product may be used with chat. Product is used in conjunction with Category for chat advanced routing and reporting. Even if the chat pre-routing is not used (i.e. visitor input of Product/Category from the chat login page is used for routing), define the Product/Category for the SCCL widget as it will be used for the chat reporting.

- wait_threshold – Specifies the maximum wait time threshold where a SCCL widget will be displayed. If the current wait time exceeds this value, the SCCL widget will not be displayed. This trigger is disabled when set to 0. The number of available sessions will always override wait threshold; the available session criteria must also be met. The default value is 40 seconds and used if no value is specified. For example, if the wait threshold is set to 40 seconds and the current wait time is 45 seconds, the SCCL will not be displayed until the current wait time is below 40 seconds and the minimum number of agent sessions is also met.

Where can I go for more information?

- ✓ Oracle Service Cloud User Guide - Chatting on the Customer Portal - Creating a chat link on an external page
- ✓ Oracle Service Cloud Community Forum - Syndication Widget
- ✓ Oracle Service Cloud Knowledge Base