



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Digital Channels Developer's Guide

Persistent Chat across Different Domains

2/9/2026

---

## Contents

- 1 Prerequisites
  - 1.1 Third-party Cookie Notification
- 2 Overview of the solution
- 3 Step 1: Create and host an Iframe HTML file
- 4 Step 2: Include the Cookie Provider JavaScript file
- 5 Step 3: Create an Instance of Cookie Provider or Add Cookie Provider Extension in Widget
  - 5.1 Create an Instance of Cookie Provider
  - 5.2 Third-party Cookie Notification
  - 5.3 Add Cookie Provider Extension in Widget
  - 5.4 Complete Chat Initialization Sample
- 6 Limitations

---

Learn how to make chat conversations persistent across different domains.

### Related documentation:

- 

#### Important

- Mozilla Firefox browsers do not support this functionality since it requires Cookie Partitioning.
- Starting from 2024, Chromium has announced its end of support for Cookie Partitioning for both Chrome and Edge browsers. If you're using this feature with the aforementioned browsers, Genesys recommends you update your integration to avoid usage of third-party cookies.

Digital channels supports chat persistence across different domains, allowing your customers to continue chat conversations with agents across the organization's sites they visit. It also allows your organization to have a consistent group of agents to support customers visiting different domains.

The following sections describe how to implement a cross domain persistent chat.

### Prerequisites

Before setting up persistent chat across different domains, ensure you configure the WebChatService.

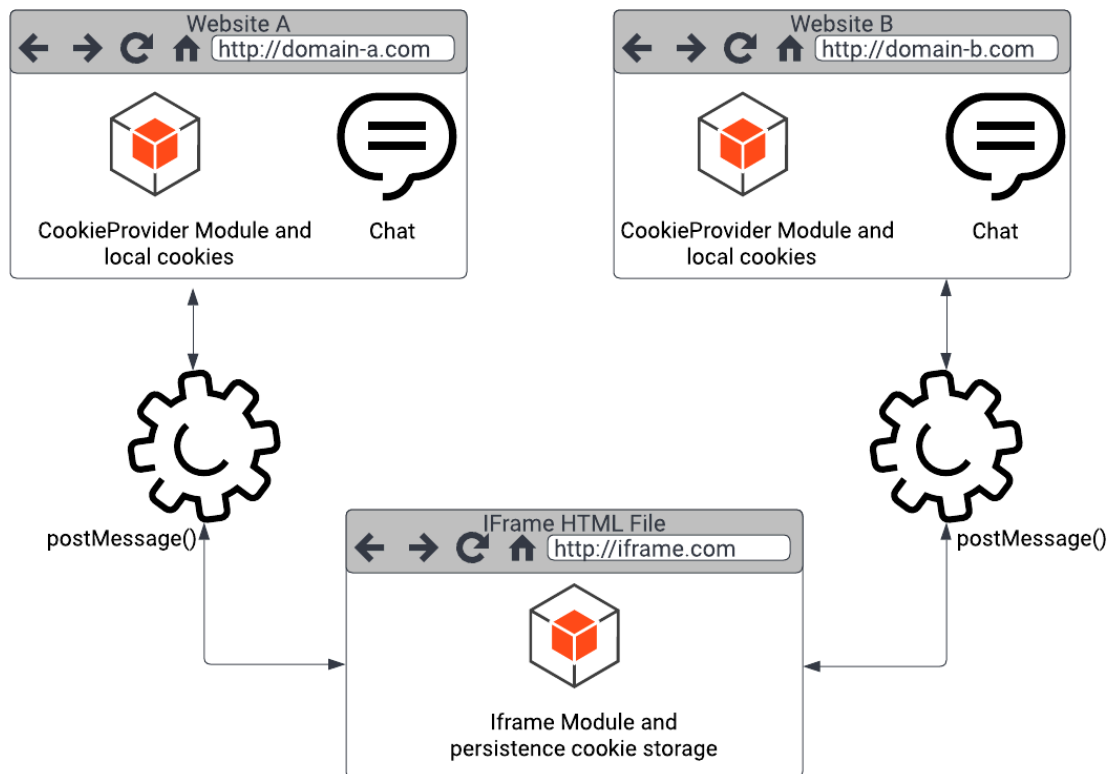
### Third-party Cookie Notification

Ensure that the third-party cookies are enabled on your customer's browser and you can suggest them to enable cookies with your custom messages on your site or in the Widget.

## Overview of the solution

Suppose that a customer started a chat on domain-a.com and intends to continue on domain-b.com, Digital Channels persists the chat conversation, and the customer can continue the interaction with the agent across the two domains.

The solution combines a third-party JavaScript file that you include on the webpages with a hidden Iframe added to the webpages. The module uses a **window.postMessage()** function of HTML5 to establish connections among the webpages and the Iframe site. The function sets the cookies and saves them on the Iframe site.



To implement this solution, complete the configuration steps.

## Step 1: Create and host an Iframe HTML file

Create a plain HTML page with no server-side dependencies and host this file on a webpage you own. Include the attached **iframe.min.js** file and use the following sample **iframe.html** file. Modify the **TRUSTED\_DOMAINS** array with the list of websites you would like to implement the chat persistence with agent.

iframe.min.js

For example:

After hosting the HTML file, note down the direct URL of the site such as

---

<https://www.iframe.com/iframe.html>. This URL is used in Step 3.

## Step 2: Include the Cookie Provider JavaScript file

Include the attached **cookie-provider.min.js** across all the webpages (in our example, it was domain-a.com and domain-b.com) where your Genesys chat widget is started.

cookie-provider.min.js

## Step 3: Create an Instance of Cookie Provider or Add Cookie Provider Extension in Widget

To start chat persistence across different websites, you can either

Create an Instance of Cookie Provider or

Add Cookie Provider Extension in Widget

### Create an Instance of Cookie Provider

Create an instance of `CookieProvider` in the script where you configure your chat widget. While creating an instance, the `CookieProvider` adds `Iframe` to your webpages. The following table describes the parameters of the `CookieProvider` class.

Parameter Name	Mandatory	Description
SESSION_COOKIE_NAME	Yes	Add your own session cookie name.
IFRAME_URL	Yes	Add the Iframe URL obtained from Step 1.
showThirdPartyCookieNotification	No	Add this parameter if you want to show a message that third-party cookies are disabled in the customer's browser. For more information about third-party cookie notifications, see Third-party Cookie Notification.

```
// provide your own SESSION_COOKIE_NAME (example 'customer-defined-session-cookie') and
// IFRAME_URL (example 'https://iframe.com/iframe.html')
// showThirdPartyCookieNotification is an optional parameter to show message if third party
// cookies are disabled
let cookieProvider = new CookieProvider(SESSION_COOKIE_NAME, IFRAME_URL,
```

---

```
showThirdPartyCookieNotification);
//load cookie
cookieProvider.loadCookie();

// chat initializing
```

After creating the instance, you can call the **loadCookie** method to receive your newly created session cookie.

### Save Cookie

To use the cookie across various websites, you must save the created cookie into the Iframe. To do this, call the **saveCookie** method after your chat session cookie is set. In the following example, the registered plugin is subscribed to the **WebChat.started** event, to save the cookie inside the callback function.

```
var cookieProviderPlugin = window._genesys.widgets.bus.registerPlugin("CookieProvider");
cookieProviderPlugin.subscribe("WebChat.started", function (e) {
    // call saveCookie when you want to save chat session cookie
    cookieProvider.saveCookie();
});
```

### Third-party Cookie Notification

You can check if third-party cookies are unavailable on your customer's browser and suggest them to enable cookies with your custom messages.

#### Important

Mozilla Firefox browsers do not support this functionality.

```
// example of function to show if third party cookies are enabled
function showThirdPartyCookieNotification(isEnabled) {
    // you can show here your own notification to enable third-party cookies
    alert("Third-party cookies are " + isEnabled ? "enabled" : "disabled");
}

// you can call cookieProvider.cookieTest with callback
cookieProvider.cookieTest(showThirdPartyCookieNotification);
// or without parameter if you provide callback when created instance of CookieProvider
cookieProvider.cookieTest();
```

### Add Cookie Provider Extension in Widget

The following option is an alternate solution to enable chat persistence across

---

---

different websites. In this approach, instead of creating an instance of `CookieProvider`, add the `CookieProviderExtension` in the `extensions` field of your Widget's configuration. As a part of an extension, you can call the callback function in `initCookieProviderExtension` to check whether third-party cookies are enabled in your customers' browser.

```
// add it in you init configuration
if (!window._genesys) window._genesys = {};

// EXAMPLE of options value
const IFRAME_URL = "http://domain-a/iframe.html";
const SESSION_COOKIE_NAME = "customer-defined-session-cookie";

// EXAMPLE of function to show notification for third-party cookies
function showThirdPartyCookieNotification(isEnable) {
  if (!isEnable) {
    alert("please enable your third-party cookies");
  }
}

window._genesys = {
  widgets: {
    extensions: {
      CookieProviderExtension: initCookieProviderExtension(SESSION_COOKIE_NAME,
IFRAME_URL, showThirdPartyCookieNotification)
    }
  }
}

// or add it separately after initialization
if(!window._genesys.widgets.extensions){
  window._genesys.widgets.extensions = {};
}

window._genesys.widgets.extensions["CookieProviderExtension"] =
initCookieProviderExtension(SESSION_COOKIE_NAME, IFRAME_URL,
showThirdPartyCookieNotification);
```

## Complete Chat Initialization Sample

The following code sample shows the chat configuration with chat persistence across different websites.

```
const IFRAME_URL = "http://127.0.0.1:5500/iframe.html";
const SESSION_COOKIE_NAME = "customer-defined-session-cookie";

if (!window._genesys) window._genesys = {};

window._genesys = {
  widgets: {
    webchat: {
      transport: {
        type: "",
        dataURL: "", // Provided by Genesys
        endpoint: "", // Provided by Genesys
        headers: {
          "x-api-key": "", // Provided by Genesys
        },
        async: {
```

---

```

        enabled: true,
        getSessionData: function (sessionData, Cookie, CookieOptions) {
            // Note: You don't have to use Cookies. You can, instead, store in a secured
location like a database.
            Cookie.set(
                SESSION_COOKIE_NAME,
                JSON.stringify(sessionData),
                CookieOptions
            );
        },
        setSessionData: function (Open, Cookie, CookieOptions) {
            // Retrieve from your secured location.
            return Cookie.get(SESSION_COOKIE_NAME);
        },
    },
    },
    },
    // you need add this option only if you use alternative connection
    extensions: {
        CookieProviderExtension
    }
},
};

const widgetScriptElement = document.createElement("script");
const widgetBaseUrl = "https://apps.mypurecloud.de/widgets/9.0/";

// provide your own SESSION_COOKIE_NAME (example 'customer-defined-session-cookie') and
IFRAME_URL (example 'https://iframe.com/iframe.html')
let cookieProvider = new CookieProvider(SESSION_COOKIE_NAME, IFRAME_URL);
//load cookie
cookieProvider.loadCookie();

// chat initializing
widgetScriptElement.setAttribute("src", widgetBaseUrl + "cxbus.min.js");
widgetScriptElement.addEventListener("load", async function () {
    await CXBus.configure({
        debug: true,
        pluginsPath: widgetBaseUrl + "plugins/",
    });
    await CXBus.loadPlugin("widgets-core");
    await CXBus.command("WebChat.open");

    // set cookie to iframe when chat started
    var cookieProviderPlugin = window._genesys.widgets.bus.registerPlugin("CookieProvider");
    cookieProviderPlugin.subscribe("WebChat.started", function (e) {
        //save cookie
        cookieProvider.saveCookie();
    });
});

document.head.appendChild(widgetScriptElement);

```

---



---

## Limitations

Chat persistence does not work when a browser is configured to disable third-party cookies. Each browser manages and defines third-party cookies and data differently.

Mozilla Firefox and Apple Safari browsers block the third-party cookies and data by default, but they allow you to set cookies manually. Google Chrome will stop supporting third-party cookies during 2024.

The following documents provide guidelines and settings for third-party cookies across different browsers.

Browser	URL
Google Chrome	<a href="https://support.google.com/chrome/answer/95647">https://support.google.com/chrome/answer/95647</a>
Mozilla Firefox	<a href="https://support.mozilla.org/en-US/kb/third-party-cookies-firefox-tracking-protection">https://support.mozilla.org/en-US/kb/third-party-cookies-firefox-tracking-protection</a>
Apple Safari	<a href="https://support.apple.com/en-gb/guide/safari/sfri11471/mac">https://support.apple.com/en-gb/guide/safari/sfri11471/mac</a>
Opera	<a href="https://blogs.opera.com/news/2015/08/how-to-manage-cookies-in-opera/">https://blogs.opera.com/news/2015/08/how-to-manage-cookies-in-opera/</a>