

Extended Tracking API

- CardioLog Client-Side API
 - The SendEvent JavaScript Function
 - The SendGoal JavaScript Function
 - Monitoring Internal E-Mail Campaigns Using the `__referrer` Parameter
- CardioLog Server-Side API
 - The SendEvent Method
 - Event Types
 - Creating Custom Event Types
- The REST API

CardioLog Client-Side API

The CardioLog Tracking Agent allows a client-side API to send events, and easily monitor button clicks, banner clicks, or interactions with other UI components. You can call the following JavaScript functions on any monitored page:

- The SendEvent JavaScript Function
- The SendGoal JavaScript Function

The SendEvent JavaScript Function

The *SendEvent* function allows you to send events via the client browser. Using this function you can monitor clicks on UI elements such as buttons, banners, navigation menu items, links, videos and more.

The Tracking Agent uses AJAX (Asynchronous JavaScript And XML) to send the event to the CardioLog server, and does not affect user interaction with the page in any way.

```
document.__Page.SendEvent('Event Type', 'Event Title', 'Param 1', 'Param 2');
```

The *SendEvent* function accepts the following parameters:

- **Event Type** - Required. The event type title (eg., 'Visit' or [custom event types](#)).
- **Event Title** - Required. This can be any text that represents the event or action. We recommend defining a unique prefix label for each logical group of objects for streamlined reports. This string is limited to 1000 characters.
- **Param 1** - Optional. Used to store additional information about the event, typically for custom event types. This string is limited to 1000 characters. Built-in events reserve this string for the system.
- **Param 2** - Optional. Used to store additional information about the event, typically for custom event types. This string is limited to 50 characters. Built-in events reserve this string for the system.

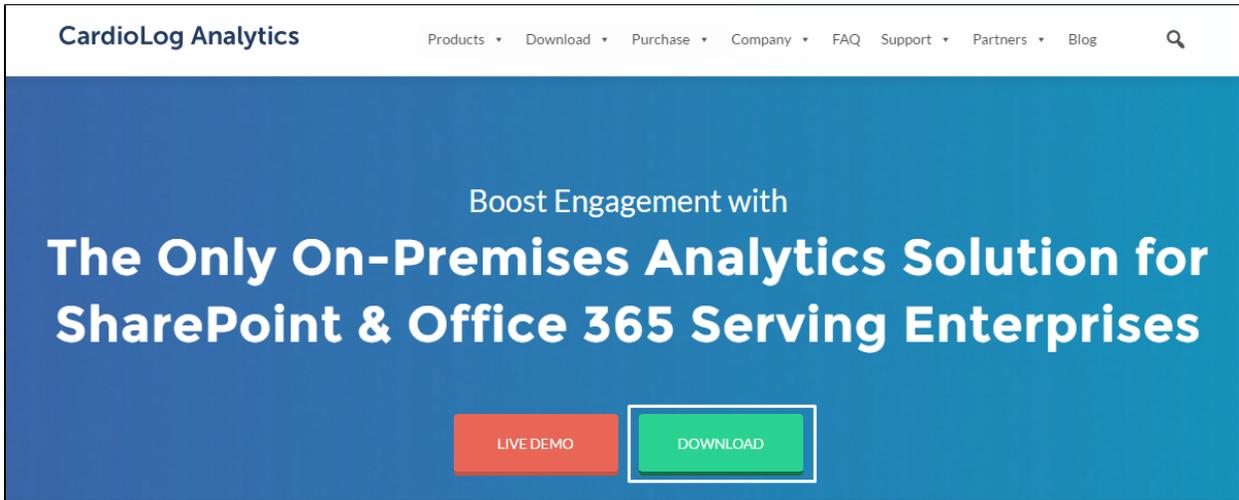
Notes:

- When calling the *SendEvent* function within a different frame from where it is defined, make sure to reference the frame object containing it. For example, by using `parent.document.__Page.SendEvent()`.
- In order to capture a click on a banner or image that links to an external site, make a call to the CardioLog *SendEvent* function before redirecting the user to the external destination page.

Code Samples

1. Send a click event on a *DOWNLOAD* button:

```
<input type="button" value="DOWNLOAD" onclick="document.__Page.SendEvent('Visit', 'Main Page Buttons > DOWNLOAD');" />
```



www.intlock.com Main Page "Download" Button

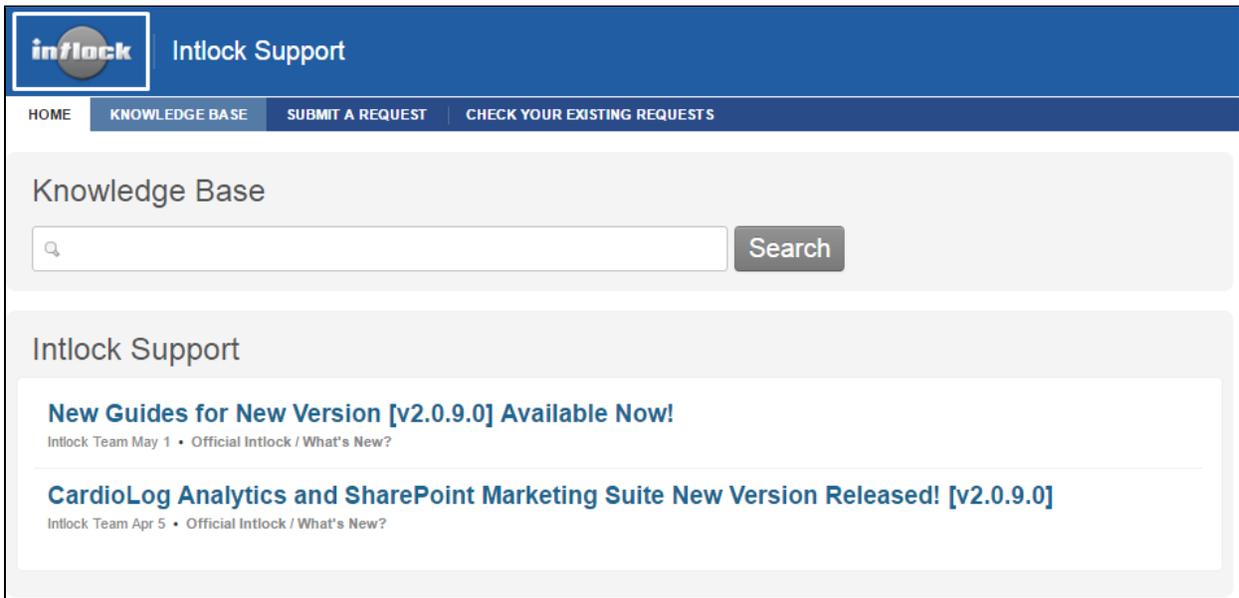
In this example, when a user clicks on the *DOWNLOAD* button CardioLog will send an event of type 'Visit' to the server, specifying which button has been clicked by the user (ie., 'Main Page Buttons > *DOWNLOAD*'). We recommend defining a unique label for each group of UI components that will be displayed in reports (in this case, 'Main Page Buttons').

To see how many times the button was clicked, use the *Events* report (chart or table), and type 'Main Page Buttons' in the Website Item field.

To see who clicked on the button, use the *Visitors by Events* report (chart or table), and type 'Main Page Buttons' in the Website Item field.

2. Send a click event on a banner that links to an external site:

```
<a href="http://www.intlock.com" onclick="document.__Page.SendEvent('Visit', 'Banners > Intlock Website');"></a>
```



support.intlock.com Main Page Intlock Logo

In this example, when a user clicks on the banner CardioLog will send an event of type "Visit" to the server, specifying which banner has been clicked by the user before redirecting to "*www.intlock.com*" (ie., "*Banners > Intlock website*"). We recommend defining a unique label for each group of UI components that will be displayed in reports (in this case, "Banners").

To see how many times the banner was clicked, use the *Events* report (chart or table), and type '*Banners*' in the Website Item field. To see who clicked on the banner, use the *Visitors by Events* report (chart or table), and type '*Banners*' in the Website Item field.

Note: CardioLog automatically tracks outbound links from SharePoint sites to external destinations. To see how many times your links were clicked, use the [External Destinations](#) report.

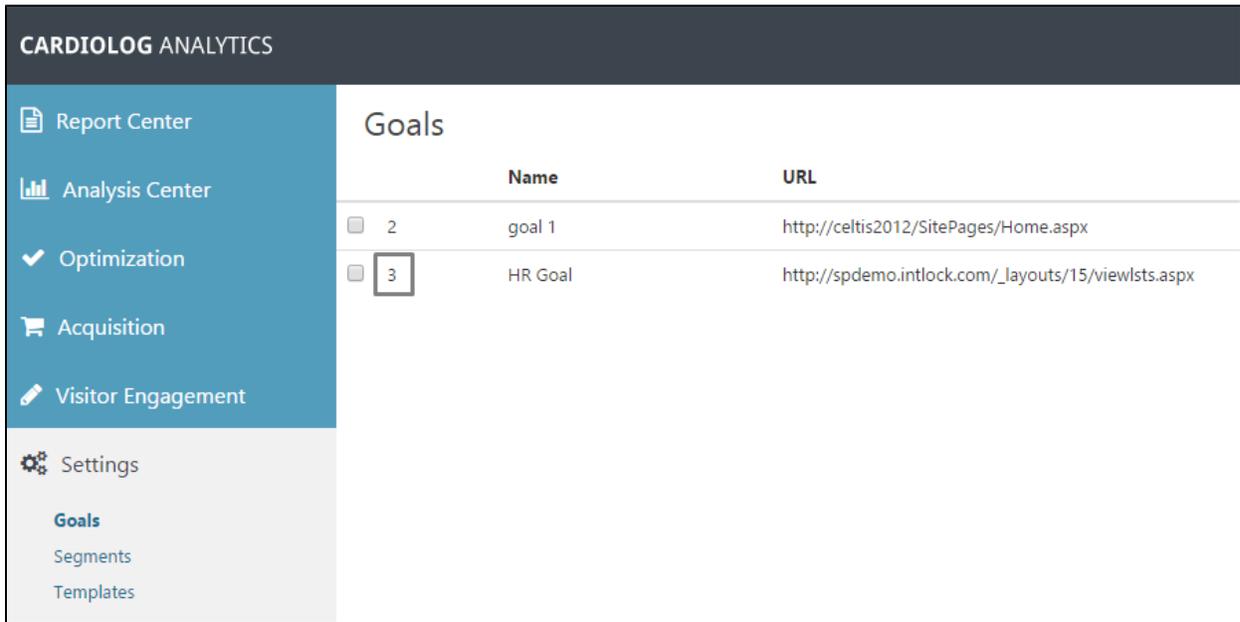
The SendGoal JavaScript Function

The *SendGoal* function allows you to send JavaScript based goals via the client browser. The Tracking Agent uses AJAX (Asynchronous JavaScript and XML) to send the goal event to the server, and does not affect the user's experience with the page in any way.

```
document.__Page.SendGoal(Goal ID);
```

The *SendGoal* function accepts the following parameter:

- **Goal ID** - Required. Goals are created in CardioLog under **Settings > Goals**. After creating a goal, the Goal ID can be found in the ID column.



CARDIOLOG ANALYTICS		
Goals		
	Name	URL
<input type="checkbox"/> 2	goal 1	http://celtis2012/SitePages/Home.aspx
<input type="checkbox"/> 3	HR Goal	http://spdemo.intlock.com/_layouts/15/viewlists.aspx

Settings - Goals

Code Sample

1. Send a goal event on a *Register to a Webinar* button:

Create the [goal](#), and make a call to the CardioLog *SendGoal* function with the proper Goal ID.

```
<input type="button" value="REGISTER NOW " onclick="document.__Page.SendGoal(1);" />
```

CardioLog Analytics

Products ▾ Download ▾ Purchase ▾ Company ▾ FAQ Support ▾ Partners ▾ Blog

Ensure Governance with

The Only On-Premises Analytics Solution for SharePoint & Cloud Serving Enterprise

LIVE DEMO DOWNLOAD

Upcoming webinar - Reserve your spot!

May 19th, 2016

How to Enhance a SharePoint Migration with Analytics

First Name: *

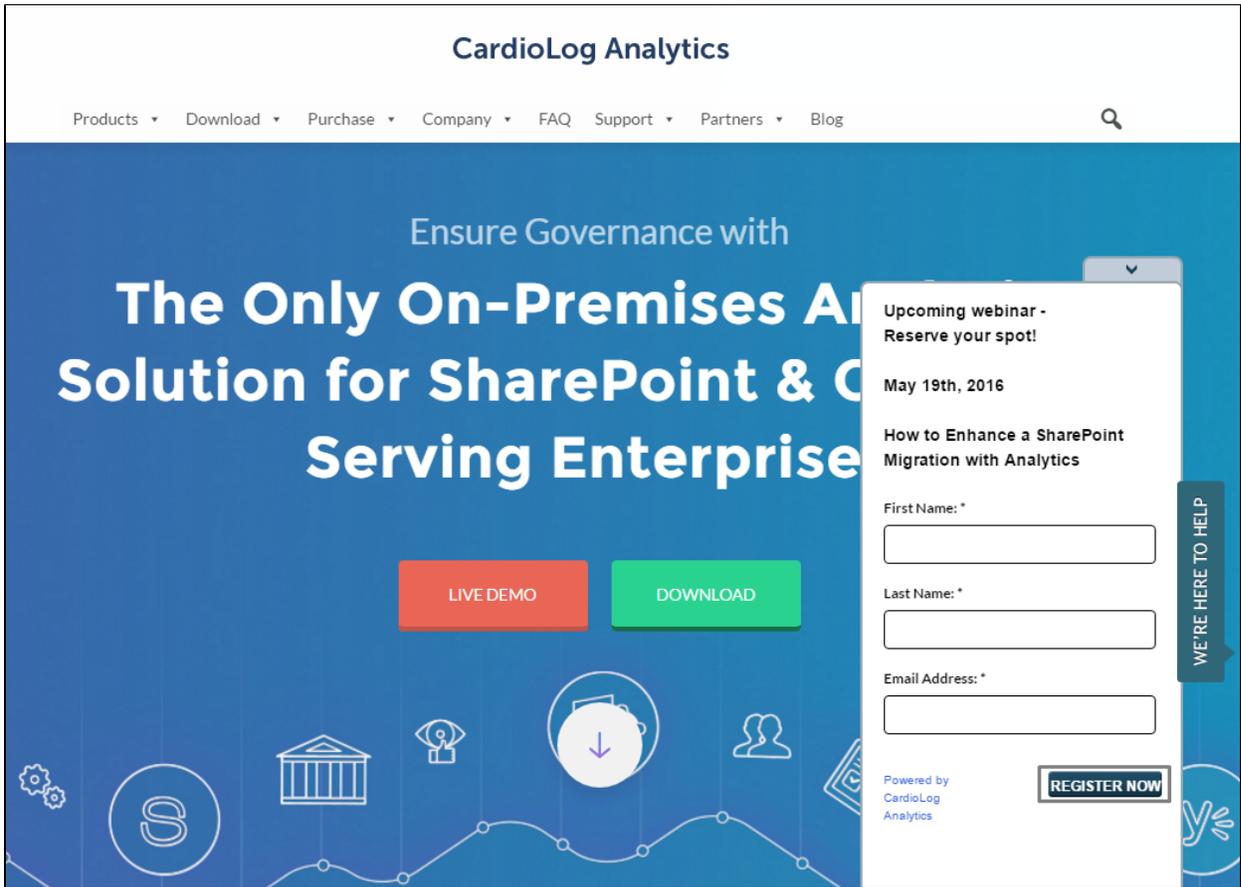
Last Name: *

Email Address: *

Powered by CardioLog Analytics

REGISTER NOW

WE'RE HERE TO HELP



www.intlock.com Main Page REGISTER NOW Survey Button

To monitor your goal progress, use the [Goal reports](#), and select the relevant goal in the *Visitor Segments > Goal* filter of the widget preferences.

Select Visitor Segments ✕

Medium: ? [Choose] ▼

Term: ? [Choose] ▼

Campaign: ? [Choose] ▼

Content: ? [Choose] ▼

User Behavior:

Goal: ? All ▼

Details and demographics:

Country: ? All ▼

Active User Categories:

Region: ? All ▼

Department: ? All ▼

Visitor Segments > Goal

Monitoring Internal E-Mail Campaigns Using the `__referrer` Parameter

The CardioLog Client-Side API supports identifying and monitoring external locations. You can use the `__referrer` parameter whenever you publish a link to your monitored environment and you wish to track the source of the referring page.

This is most commonly used in email or social media. For example, if you have included an internal link in a promotional email add the `__referrer` parameter to track the number of users who visited your site via this link. Define a unique referrer name that will be displayed in reports, such as `__referrer=email_promotion_Dec_2016` in order to display the link accordingly:

`http://myportal/sites/rnd/default.aspx?__referrer=email_promotion_Dec_2016`

To see how many visitors arrived on your page through your email campaign, use the [Internal Traffic Sources](#) report, and select the relevant page in the Website Item URL field.

CardioLog Server-Side API

By using the CardioLog Server-Side API, `/CardioLogAPI/Events.aspx` web service, and the `SendEvent` method, you can send `view`, `duration` and `search` events, custom events, and track the URL of the referrer page.

The SendEvent Method

The `SendEvent` method in the `/CardioLogAPI/Events.aspx` web service accepts the following parameters:

- **UserName** - Optional. The user associated with the event. If not passed, the method will use the current user credentials.

- **SessionID** - Required. A unique browser session ID associated with the event.
- **Event Type** - Required. The event type ID number
- **URL** - Required. The URL of the page where the event occurred.
- **UserAgent** - Optional. The user details including browser type and OS version.
- **ClientIP** - Optional. The user IP address.
- **Param1** - Optional. Used to store additional information about the event, typically for custom event types. This string is limited to 1000 characters. Built-in events reserve this string for the system. For example, in a "Visit" event it stores the referrer information; in a "Search" event it stores the search term.
- **Param2** - Optional. Used to store additional information about the event, typically for custom event types. This string is limited to 50 characters. Built-in events reserve this string for the system. For example, in a "Search" event it is used to store the number of results.

After the event is sent, the web service XML response will include the event identifier for future use, which you can later use to send events related to this event accordingly:

```
<agent>
<log>
<status id='0'>OK</status>
<event id='211022' />
</log>
</agent>
```

In case of an error, the XML response will include this error description:

```
<agent>
<log>
<status id='-1'>Invalid Path</status>
</log>
</agent>
```

Event Types

The SendEvent method supports four event types. The ID number is used in the Server-Side API. The Event Type Title is used in the Client-Side API.

View - Individual page request, or element clicked

- Event Type ID - 0
- Event Type Title - "Visit"

To set the page referrer, use the following optional parameters to track the source of the page:

- Param1 -The related view event ID or a string representing referrer types such as navigation toolbars, campaigns, banners, etc.
- Param2 -The referrer page URL

Referrer pages can be retrieved in CardioLog with the "Navigation" visual controls.

Duration - Time spent on a page or element (in seconds).

1. Event Type ID - 1
2. Event Type Title - "Leave"
3. URL - The related view event URL.
4. Param1 - Duration in seconds.
5. Param2 - The related view event ID.

Search - Onsite (internal) search

- Event Type ID - 2
- Event Type Title - "Search"
- URL - The related view event URL.
- Param1 - Search keyword/phrase.
- Param2 - Number of search results.

Search events can be viewed in CardioLog in the Onsite Searches report, popular search terms are displayed in the Onsite Search Phrases report.

Search Result Click -

- Event Type ID - 5
- Event Type Title - "SearchResultClick"
- URL - The search result URL.
- Param1 - The search result position within search results.

- Param2 - The related search event ID.

Creating Custom Event Types

You can create custom event types and build custom reports using the [CardioLog SDK](#). The custom event types can be used in the SendEvent function, for both Client-Side and Server-Side APIs. For example, if your website is a sales site and you wish to track each product sold, and then you can create a report on the number of items sold, and create a custom event type in CardioLog named "Items-Sold."

1. Add the custom event type to the CardioLog database by executing the following query against the CardioLog database:

```
INSERT INTO tab_event_type(id,description)
VALUES(100, 'Items-Sold')
GO
```

- **ID** - must be 100 or higher
- **Description** - event type title

2. Add the custom event type to the CardioLog tracking agent events array - edit the __eEvents array and add the custom event type in:

For MOSS 2007 only:

[CardioLog installation directory]\CardioLogAgent\MOSS2007_1.2.js

For SharePoint 2010 only:

[CardioLog installation directory]\CardioLogAgent\SP2010_1.2.js

For SharePoint 2013 only:

[CardioLog installation directory]\CardioLogAgent\SP2013_1.2.js

For SharePoint 2016 only:

[CardioLog installation directory]\CardioLogAgent\SP2016_1.2.js

Note: If you have implemented the Tracking Agent using the CardioLog Analytics SharePoint Feature on your WFE (Web Front End), edit the __eEvents array and add the custom event type accordingly:

For MOSS 2007 only:

C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\TEMPLATE\LAYOUTS\CardioLogAgent\MOSS2007_1.2.js

For SharePoint 2010 only:

C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\LAYOUTS\CardioLogAgent\SP2010_1.2.js

For SharePoint 2013 only:

C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\LAYOUTS\CardioLogAgent\SP2013_1.2.js

For SharePoint 2016 only:

C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\LAYOUTS\CardioLogAgent\SP2016_1.2.js

```
__eEvents = {
  Visit: { id: 0, name: "Visit" },
  Leave: { id: 1, name: "Leave" },
  Search: { id: 2, name: "Search" },
  SearchResultItem: { id: 4, name: "SearchResultItem" },
  SearchResultClick: { id: 5, name: "SearchResultClick" },
  External: { id: 10, name: "External" },
  Goal: { id: 20, name: "Goal" },
  "Goal-Test": { id: 21, name: "Goal-Test" },
  Test: { id: 22, name: "Test" },
  Score: { id: 23, name: "Score" },
  Ping: { id: 24, name: "Ping" },
  Item-Sold: { id: 100, name: "Item-Sold" },
  Unknown: { id: -1, name: "Unknown" }
};
```

3. Implement the call to the `SendEvent` function when an item is sold (using our [Client-Side API](#) or [Server-Side API](#)).
4. Create a custom report to display the number of items sold using the [CardioLog SDK](#).

The REST API

In order to track visitor actions for custom browsers and non-browser apps, developers must send usage event objects to the `CardioLogAgent` web application through HTTP POST requests in JSON format. The JSON string must be embedded in the request body. The POST response is provided in JSON format as well.

- The request must include a "Content-Type: application/json" header
- The request may optionally include an "Accept: application/json" header

The Event Object

CardioLog Analytics usage event objects have the following properties:

- **u** - Decoded URL of the page the event is related to. For example, the URL of a "View" event will be the current page; the URL of a search result will be the clicked link.
- **X** - Holds the following event dependent information:
 - Search event - The search term
 - View event:
 - If `__Referrer` parameter was passed to this page, `__Referrervalue`
 - If there is a Referrer cookie, sends the `ReferrerReason` cookie value
 - If this is the first page the user views in this session, sends "First In Session" and appends the data from `__Referreragent`
 - Duration event - Time on page in seconds
- **Y** - Holds the following information:
 - In Search event - Number of total results
 - In Visit event - If there is a Referrer cookie, sends its content
 - In Duration event - Last visit event ID
- **Z** - In Search event - First result on page (number of results)
- **RI** - Last event ID
- **et** - Event type code (eg., Goal event is 20, Leave event is 1, etc.)
- **events** - The event can have sub events sent with it. Typically, this is used when a search event is sent or with saved events.

In addition to event objects, the request body includes the following properties:

- **browserType** - user-agent information.
- **sessionid** - Required. A unique browser session ID associated with the event. The unique session ID associated with the events. Any string format accepted.
- **un** - username associated with the events. Any string format accepted. If not passed, the current user credentials will be used.

The JSON Request Format

The full JSON request format is as follows:

```
{
  "events": {
    "event": [
      {
        "u": "http://www.intlock.com/pages/home.aspx",
        "X": "|referral|blog.intlock.com|/about/|",
        "et": 0,
        "RI": 564940
      },
      {
        "X": "1",
        "Y": "7",
        "et": 23,
        "RI": 0
      }
    ]
  },
  "browserType": "Apple-iPhone5C2/1001.525",
  "sessionId": "861529144.325047542",
  "un": "INTLOCK\\tomj"
}
```

The API URL

The default API URL for a SharePoint site is:

[https://\[SharePoint WFE\]/_layouts/CardioLogAgent/tunnel.aspx?random=895.3324414324015](https://[SharePoint WFE]/_layouts/CardioLogAgent/tunnel.aspx?random=895.3324414324015)

- Replace [SharePoint WFE] with your SharePoint URL address
- The random param is used as a cache buster