

➤ Topic: Adding iframe for viewing a dashboard

Sources: <https://learn.microsoft.com/en-us/dynamics365/customerengagement/on-premises/developer/use-iframe-and-web-resource-controls-on-a-form?view=op-9-1>

<https://intercom.help/goavaSE/sv/articles/6262479-microsoft-dynamics-add-goava-s-company-dashboard-using-an-iframe-solution>

Explanation: The main purpose of having an IFRAME in dynamics is to embed content from another location in HTML pages. We use it here to add a new dashboard to an existing dashboard by viewing its contents in the iframe inserted, which will reference the dashboard located in another place.

If the iframe contains contents on a different domain, then we would need to enable communication.

We can use the Pass record object-type code and unique identifier as parameters option if we want to configure the iframe to process parameters passed to it which are placed in the URL defined in the control.

Parameter	Name	Description
<code>typename</code>	Entity Name	The name of the entity.
<code>type</code>	Entity Type Code	The integer that uniquely identifies the entity in a specific organization.
<code>id</code>	Object GUID	A GUID that represents a record.
<code>orgname</code>	Organization Name	The unique name of the organization.
<code>userlcid</code>	User Language Code	The language code identifier that is being used by the current user.

Passed parameters are typically read in the target .aspx page by using the `HttpRequest.QueryString` property. In an HTML page, the parameters can be accessed by using the `window.location.search` property in JavaScript.

In order to pass form data, we use the `getValue()` method on the attributes that contain data we want to pass to the other website and compose a string of the query string arguments the other page will be able to use.

Then use a Field OnChange Event, IFRAME OnReadyStateComplete Event, or Tab TabStateChange Event and the setSrc method to append your parameters to the src property of the IFRAME or web resource.

It is also mentioned to avoid using the OnLoad Event. IFRAMES and web resources load asynchronously and the frame may not have finished loading before the Onload event script finishes.

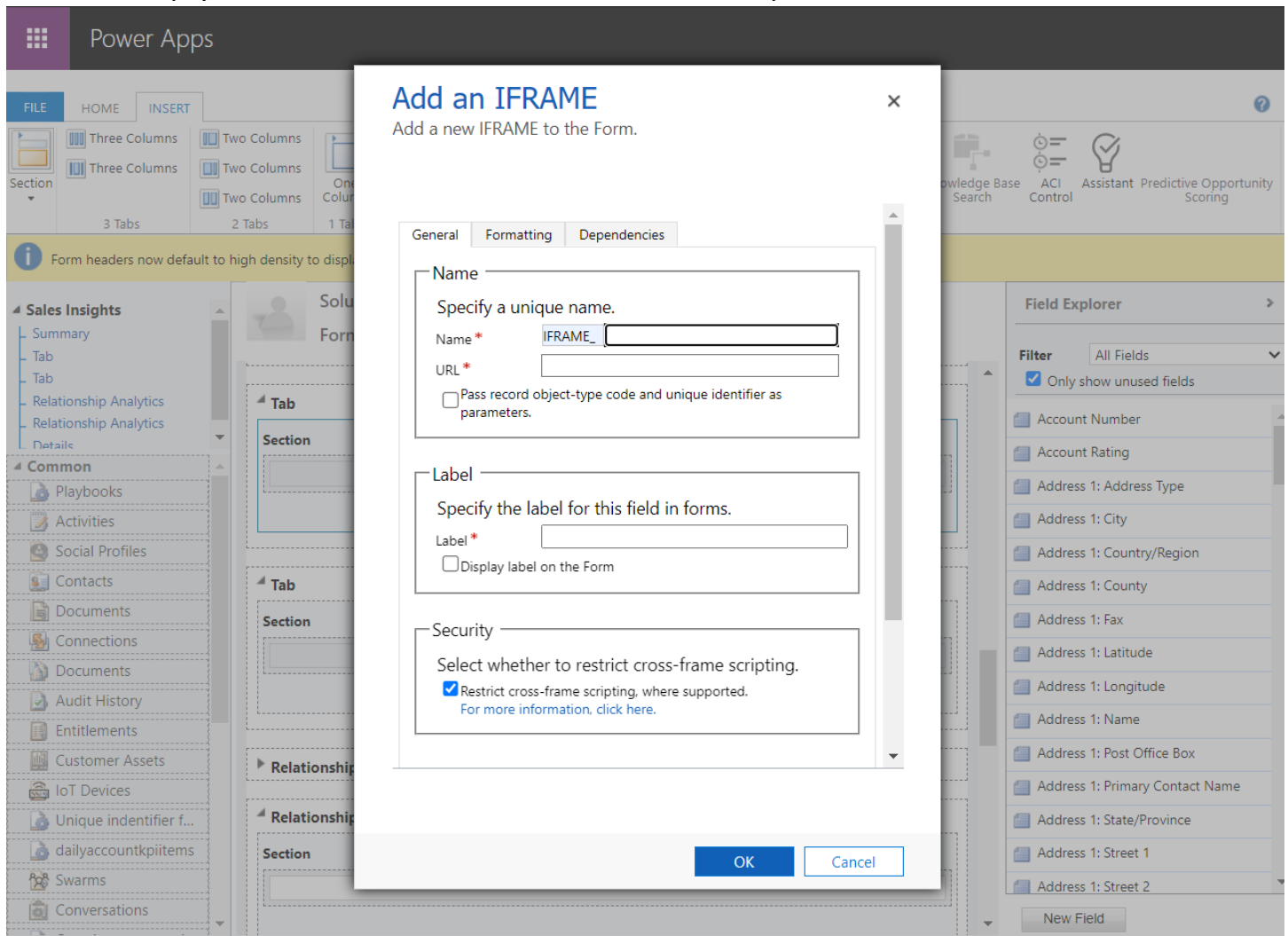
Code:

```
//Get the value of an option set attribute
var formContext = executionContext.getFormContext();
var value = formContext.getAttribute("new_pagechooser").getValue();
var newTarget = "";
//Set the target based on the value of the option set
switch (value) {
    case 100000001:
        newTarget = https://myServer/test/pageOne.aspx;
        break;
    default:
        newTarget = https://myServer/test/pageTwo.aspx;
        break;
}
//Get the default URL for the IFRAME, which includes the
// query string parameters
var IFrame = formContext.ui.controls.get("IFRAME_test");
var Url = IFrame.getSrc();
// Capture the parameters
var params = Url.substr(Url.indexOf("?"));
//Append the parameters to the new page URL
newTarget = newTarget + params;
// Use the setSrc method so that the IFRAME uses the
// new page with the existing parameters
IFrame.setSrc(newTarget);
```

Steps:

1. Start by going to Settings -> Customizations and select "Customize the System".
2. From the left navigation bar, select Components -> Entities -> Account ->Forms. Double click the Account on this list to add the element to the Accounts view or any other view you'd like to use. In this example, we'll use the Sales Insights view.

- When you are in the Account View you will be able to create a Column for a Dashboard. Select Insert -> One Column. The column should be visible in the center of the page. Click on the empty section inside the new Column and on top Insert IFRAME.



- Give the IFRAME property a name. Add a name to the label as well and add about:blank to the URL section. Make sure you don't have "Restrict cross-frame scripting" selected. Make a note of what you named the IFRAME, you'll need this name later.
- Add formatting settings as you prefer: 20-30 rows as height. Add Row layout to expand automatically. Enable ALWAYS scrolling for the IFRAME.
- Click on OK to save IFRAME. After you have saved IFRAME double click the Column again and you should see the events section. You should now add a Form Library. Click the green plus icon to add new. Click "New" again, then "Web resources" and a new window will open.
- For this part you need to know the API-name of the field where you store organization number/CVR-number/business ID for your accounts. Create a name for Web resource, add Display Name and Description. For Content you select Script and paste the script below to the Text Editor. Remember that the code is case-sensitive. Click on OK and Publish. Just like the code above we will do the same thing:

```
function changeURL(executionContext) {
    var formContext = executionContext.getFormContext();
    var business_id = formContext.getAttribute("new_organizationnumber").getValue();
    var IFrame = formContext.ui.controls.get("IFRAME_name");
    var url = "https://www.ejada.com/" + business_id + "?showAll";
    IFrame.setSrc(goava_url);
}
```

8. Add this function as an event handler for the IFRAME in the events tab
9. Save and publish changes. Done!

➤ Topic: **Importing external data from another portal using Tableau & view it in dynamics 365**

Sources: <https://www.progress.com/blogs/microsoft-dynamics-crm-to-tableau-via-odata>
<https://learn.microsoft.com/en-us/dynamics365/customerengagement/on-premises/basics/start-your-day-dashboard-chart?view=op-9-1>
<https://community.tableau.com/s/>

Explanation: In order to import external data outside of dynamics 365 to charts we have in our dashboards for example, we use a tool called Tableau.

Tableau was founded in 2003 as a result of a computer science project at Stanford that aimed to improve the flow of analysis and make data more accessible to people through visualization.

First, we need to go to the customization tab and select the developer resources.

The information we need is under Organizational Service. The Endpoint Address is what DataDirect Cloud will use to ping Dynamics. Keep this URL saved.

To set up the DataDirect Cloud, we will open the main page and select data sources in the left side panel. Many options will appear so we will choose to connect to Microsoft Dynamics CRM

Once the “Create Dynamics CRM Source” page is loaded, be sure to select the Cloud option. You should see a few changes to the form made between the On-Premise and Cloud options.

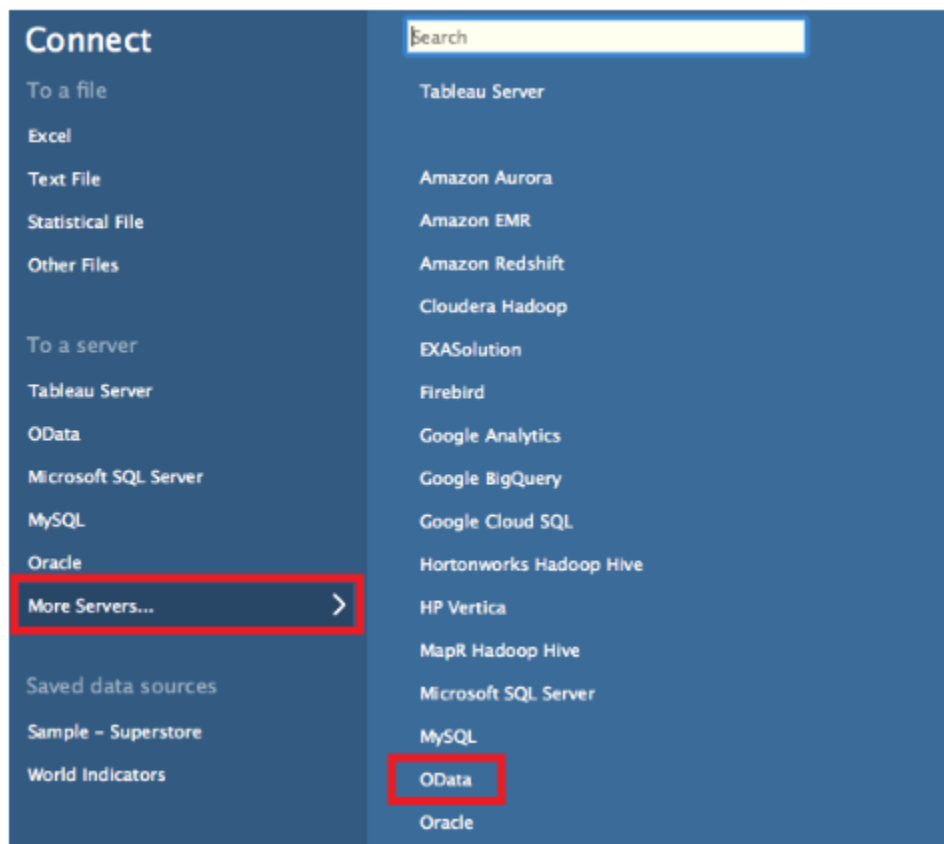
Then, fill out the form which contains fields such as Data source name, Description, User ID, Password and the organization service URL which we got from the Developer resources section. Test the connection and you should then see a success message that confirms all fields are set correctly.

From here we will navigate to the OData tab. This tab will need to be populated with some info to properly configure an OData connect. First, let's select "Configure Schema."

Select DYNAMICSCRM from the Select Schema pull down. From here you can decide to map whichever of the tables or objects available to you from Dynamics. As a note, this holds true for all of the DataDirect Cloud OData connections that establish.

Once an entity (Example: Account) is selected, the schema map field will be populated and at this point we are ready to configure Tableau to consume the Dynamics data via OData.

Open Tableau. As always, at the launch screen, you'll get several connection options. One of which is OData, which can be found under More Servers.

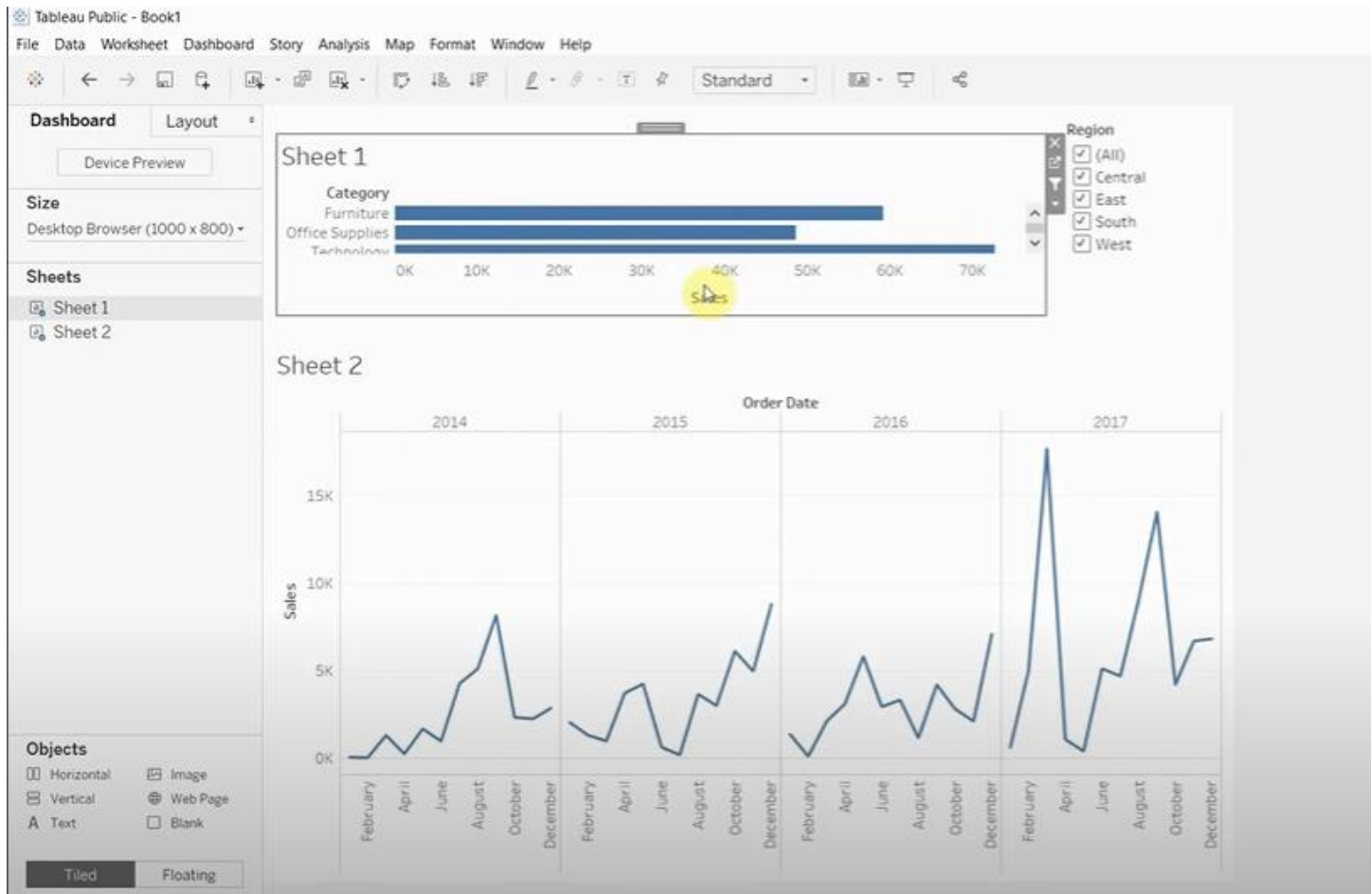


Once selected, the OData Connection prompt will appear. It will need a few bits of info. We will give it the ACCESS URI available in the DataDirect Cloud in the OData section with /ACCOUNT at the end. Select 'Use a specific username and password. This will be your D2C username and password.

After a few seconds Tableau will populate metadata about the table. More specifically it shows the available columns from that table. Select Sheet 1 towards the bottom and you are free to make great visualizations.

We are ready to use data from the Account entity to create charts and visualizations along with external data we would like to add from another portal, then import these back to dynamics 365 in order to view it.

We will then use a web resource in Dynamics 365 CRM that loads the URL of a Tableau dashboard by Creating a JavaScript with a parameterized URL.



Note: There is also a third-party tool to connect Dynamics 365 and Tableau — Skyvia. This no-coding solution provides integration through a database or cloud data warehouse with automatic schema creation. Then you can connect Tableau to this database or data warehouse.

Steps To embed a Tableau dashboard in Dynamics 365 using JavaScript, you can follow these general steps:

1. Get the Tableau Embed Code: In Tableau, publish your dashboard to Tableau Server or Tableau Public and obtain the embed code for the dashboard. This code will be used to embed the dashboard in Dynamics 365.
2. Open Dynamics 365 Form: Open the Dynamics 365 form where you want to embed the Tableau dashboard.

3. Add a Web Resource: In Dynamics 365, add a web resource that will host the JavaScript code to embed the Tableau dashboard. Go to the customization area of Dynamics 365 and create a new web resource. Choose the type as "Webpage (HTML)".
4. Add JavaScript Code: Edit the HTML of the created web resource and add JavaScript code to embed the Tableau dashboard. You can use the following code as a starting point:

```
<script>
function loadTableauDashboard() {
  var dashboardUrl = 'YOUR_TABLEAU_DASHBOARD_URL';
  var tableauOptions = {
    hideTabs: true,
    hideToolbar: true,
    width: '100%',
    height: '600px'
  };

  var vizElement = document.getElementById('tableauViz');
  var viz = new tableau.Viz(vizElement, dashboardUrl, tableauOptions);
}

document.onreadystatechange = function() {
  if (document.readyState === 'complete') {
    loadTableauDashboard();
  }
};
</script>
```

5. Replace 'YOUR_TABLEAU_DASHBOARD_URL' with the actual URL of your Tableau dashboard.
6. Save and Publish: Save the web resource and publish the changes in Dynamics 365.
7. Add Web Resource to Form: Go back to the Dynamics 365 form and add the web resource to the desired section of the form. Configure the web resource properties and set the height and width according to your requirements.

8. Test the Embedding: Save and publish the form changes in Dynamics 365. Open the form and verify if the Tableau dashboard is embedded correctly.

To import Tableau data into Dynamics 365, you can follow these general steps:

Export Tableau Data: In Tableau, export the data you want to import into Dynamics 365. You can export the data as a CSV, Excel, or any other supported format.

Prepare Data for Import: Review the exported data and make sure it is in a format compatible with Dynamics 365. You may need to clean up or transform the data to match the structure and requirements of your Dynamics 365 entities.

Import Data Using Data Import Wizard: In Dynamics 365, use the Data Import Wizard to import the data. The Data Import Wizard guides you through the process of mapping the data fields from the source file to the corresponding fields in Dynamics 365.

- Go to the Dynamics 365 app where you want to import the data.
- Navigate to Settings > Data Management > Data Import.
- Click on "Import Data" and follow the wizard steps.
- Select the file containing the exported Tableau data.
- Map the fields from the source file to the corresponding fields in Dynamics 365 entities.
- Configure import options, such as duplicate detection and error handling.
- Start the import process and monitor the progress.
- Validate and Review Imported Data: Once the import process completes, review the imported data in Dynamics 365 to ensure it has been imported correctly. Verify that the data is accurate and properly associated with the appropriate records.
- Perform Data Cleanup and Post-Import Tasks: After importing the data, you may need to perform additional cleanup tasks, such as updating or correcting records, associating imported data with related entities, or performing data validation checks.