**Using Context Navigation and User Settings in PCF Controls**

First, let’s create a new PCF control called SampleXrmControl, we will use the standard template (not the React template) for this example:

**pac pcf init -n SampleXrmControl -ns carl -t field**

Let’s add some simple code to display a button in the init function. This HTMLButtonElement button will call an onclick function when clicked. The function calls a standard JavaScript alert, and then it calls the context that we get with PCF controls. The context has navigation capabilities, and these have openAlertDialog, which we use when building Power apps and Dynamics apps. Let’s open a dialog this way:

***const button: HTMLButtonElement = document.createElement("button");***

***button.innerHTML = "Click me";***

***button.onclick = () => {***

***alert("Hello World");***

***var alertStrings = { confirmButtonLabel: "Yes", text: "This is an alert.", title: "Sample title" };***

***var alertOptions = { height: 120, width: 260 };***

***context.navigation.openAlertDialog(alertStrings, alertOptions);***

***;***

***}***

***container.appendChild(button);***

That’s all we need to do. We can publish this to our org, Once published, add a field to a form and go to Components to change it to our new PCF component. Here we will bind it to the Account Number field on the Account form, and we can select our SampleXrmControl:

A screenshot of a computer component

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Once added we see the control rendered with the Click Me button:

A screenshot of a computer

AI-generated content may be incorrect.

Let’s Save and Publish so we can see it on a record. We now see:

A screenshot of a computer

AI-generated content may be incorrect.

Clicking the button, the standard JS alert:

A white rectangular object with black text

AI-generated content may be incorrect.

And the Dynamics 365 alert dialog is displayed:

A screenshot of a computer

AI-generated content may be incorrect.

And clicking Yes closes the alert dialog and logs to the console:



Let’s look at some of the other functionality we get with context.navigation. We have openConfirmDialog, openErrorDialog, openFile, openForm, openUrl, and openWebResource:

A screen shot of a computer

AI-generated content may be incorrect.

Let’s add some code to do the openUrl:

var openUrlOptions = { height: 500, width: 500 };

context.navigation.openUrl("https://microsoft.com",openUrlOptions);

We now see a browser window open to the dimensions and URL specified:

A screenshot of a computer

AI-generated content may be incorrect.

Let’s now take a look at the **userSettings** context. We have the following:

* dateFormattingInfo
* getTimeZoneOffsetMinutes
* isRTL
* languageId
* numberFormattingInfo
* securityRoles
* userId
* userName

A screenshot of a computer program

AI-generated content may be incorrect.

Let’s try the userId and userName, printing it on the screen using the previously used ***openAlertDialog:***

***var DisplayText = `User Id: ${context.userSettings.userId}, UserName: ${context.userSettings.userName}`;***

***var alertStrings = { confirmButtonLabel: "Yes", text: DisplayText, title: "Sample title" };***

***var alertOptions = { height: 120, width: 260 };***

***context.navigation.openAlertDialog(alertStrings, alertOptions).then(***

***function (success) {***

***console.log("Alert dialog closed");***

***},***

***function (error) {***

***console.log(error.message);***

***});***

On running this, we see the User Id and User Name displayed:

A screenshot of a computer

AI-generated content may be incorrect.