Build your Web App Widgets using TypeScript

Mark Cederholm

UniSource Energy Services

Flagstaff, Arizona

What is TypeScript?

- Typed superset of JavaScript
- Driven by type definition files (.d.ts)
- Compiles to plain JavaScript
- Generated map files (.js.map) allow debugging TypeScript source directly in a browser
- Integrates with Visual Studio for IntelliSense and error checking

Why use TypeScript?

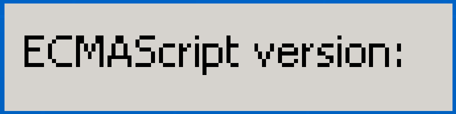
- Encourages more structured and legible code
- IntelliSense enhances productivity
- Checks your logic while coding
- Visual Studio development experience
- Available for other IDEs and editors:

Eclipse, WebStorm, Atom, and Sublime Text

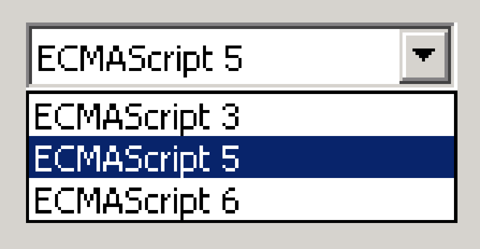
TypeScript Resources

- TypeScript home:
  www.typescriptlang.org
- DefinitelyTyped home:
  definitelytyped.org
- MSDN blog:
  blogs.msdn.com/b/typescript/
- Esri jsapi-resources:
  github.com/Esri/jsapi-resources

ES5 or ES6?



ECMAScript version:



ECMAScript 5

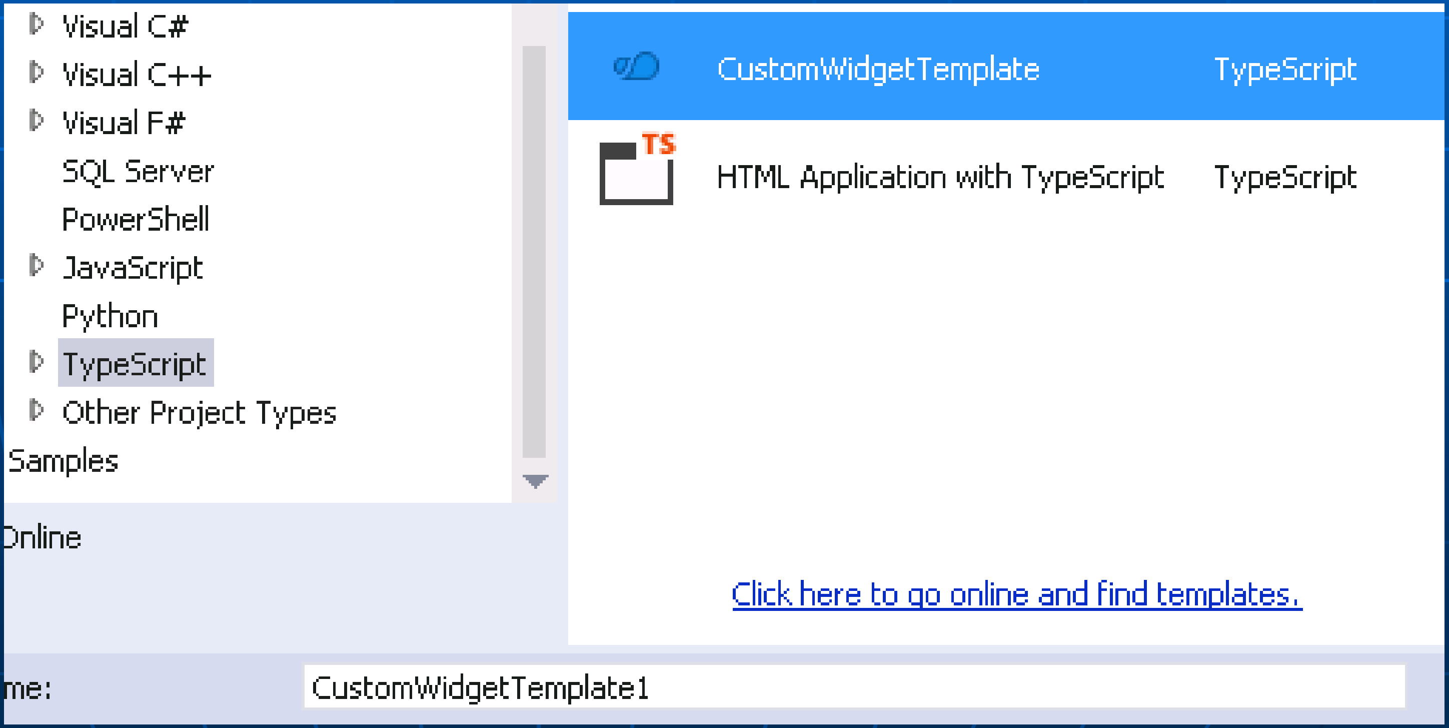
ECMAScript 3
ECMAScript 5
ECMAScript 6

- ES6 syntax is not fully supported by modern browsers (e.g. import)
- Requires a transpiler such as Babel
- ES5 type definition files must be edited to conform to ES6 syntax

Examples in this presentation use ES5 output

A Sample Project Template

- For creating a Web AppBuilder 1.3 custom widget
- Included in sample code download



CustomWidgetTemplate	TypeScript
HTML Application with TypeScript	TypeScript

Click here to go online and find templates.

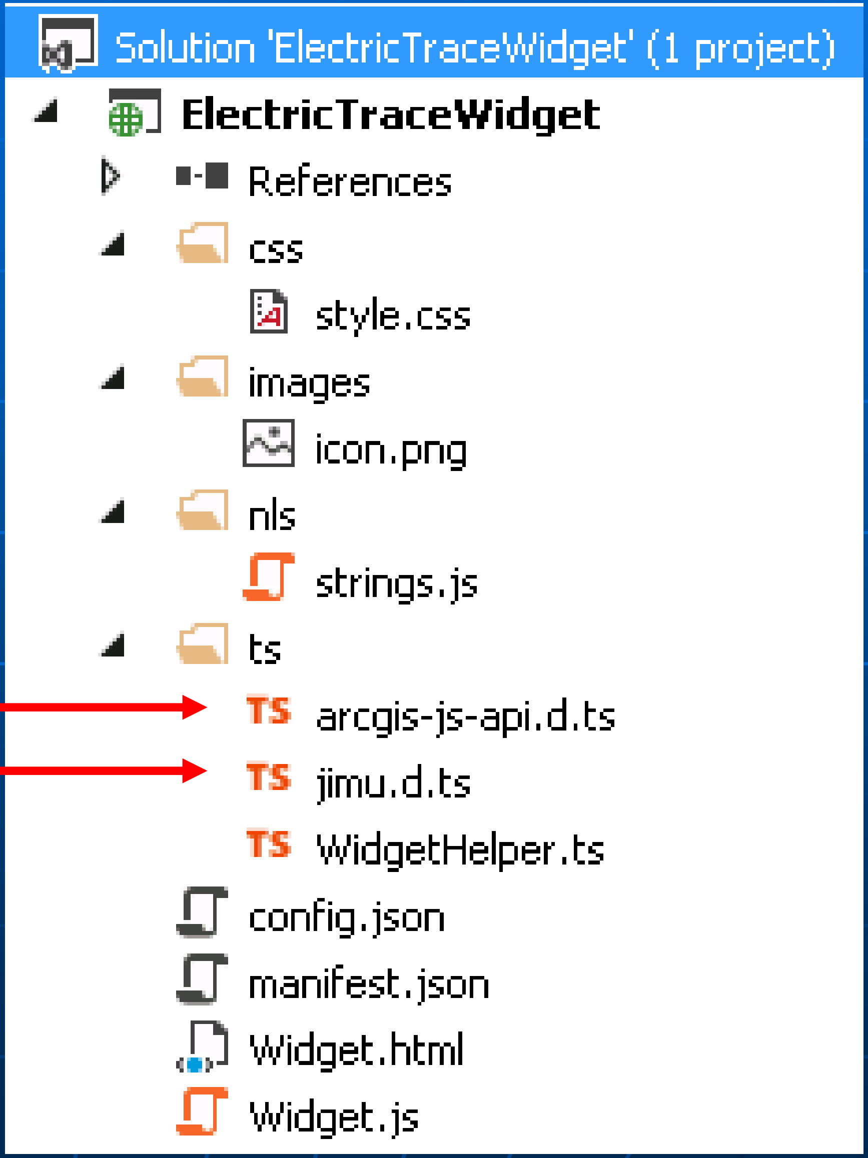
me: CustomWidgetTemplate1

4:25:14 PM

Project Structure

ArcGIS API for JavaScript (3.15)

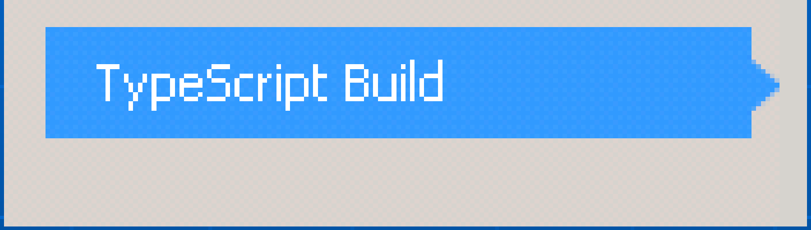
BaseWidget



```
Solution 'ElectricTraceWidget' (1 project)
◢ ElectricTraceWidget
    ▷ References
    ◢ css
          style.css
    ◢ images
          icon.png
    ◢ nls
          strings.js
    ◢ ts
          arcgis-js-api.d.ts
          jimu.d.ts
          WidgetHelper.ts
      config.json
      manifest.json
      Widget.html
      Widget.js
```

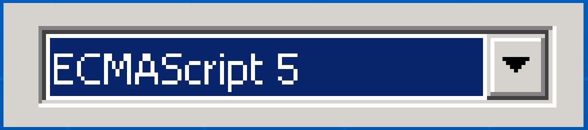
4:25:14 PM

Project Properties

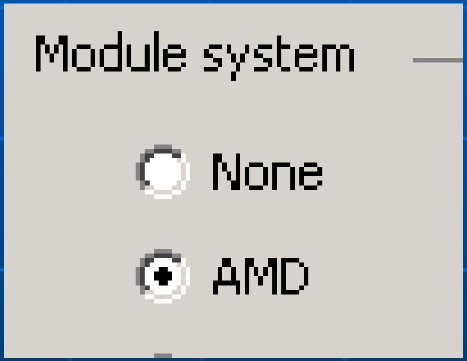


TypeScript Build

ECMAScript version:



ECMAScript 5



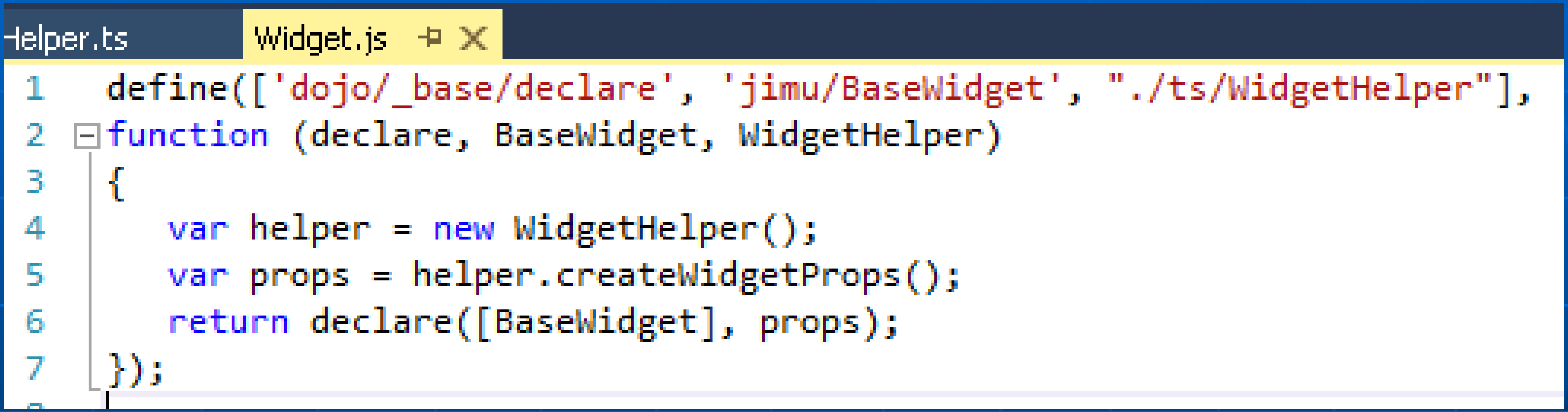
Module system

○ None

● AMD

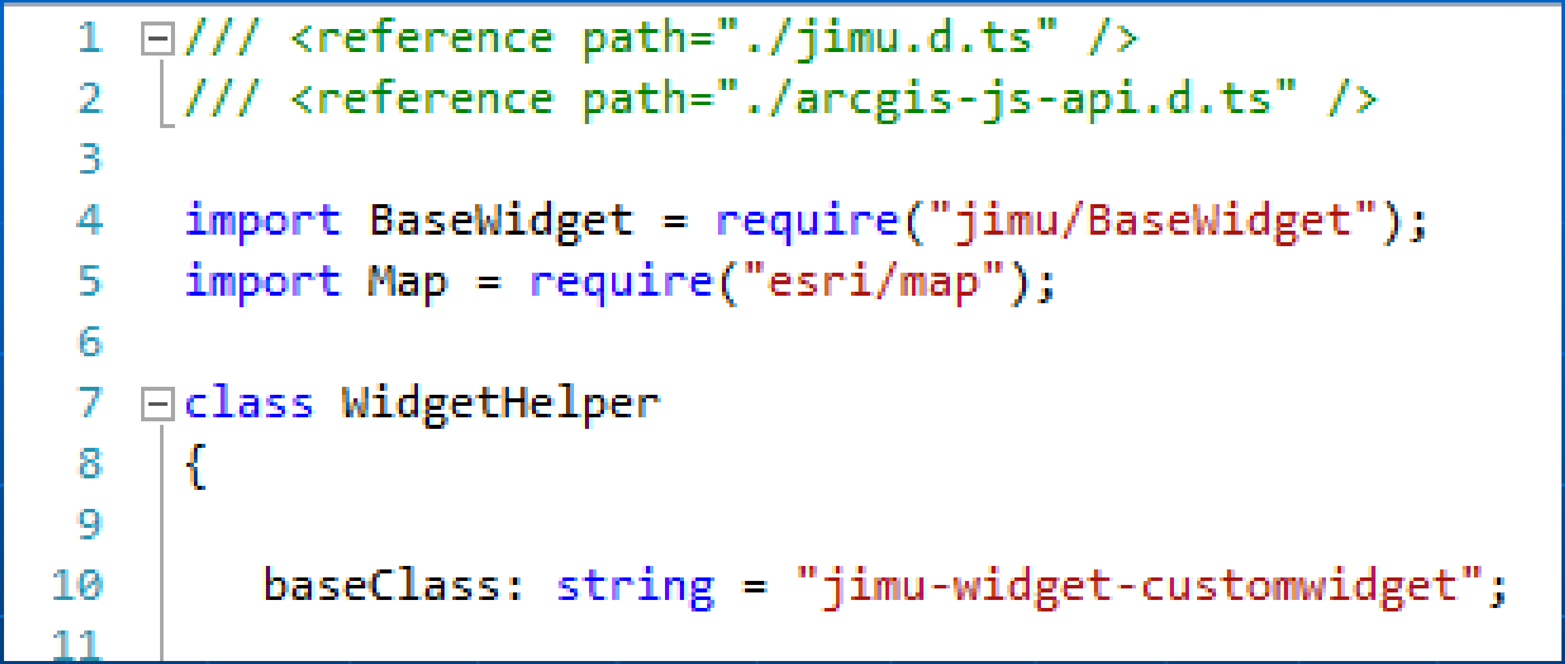
If ES6 output is selected, set Module system to None

Widget.js



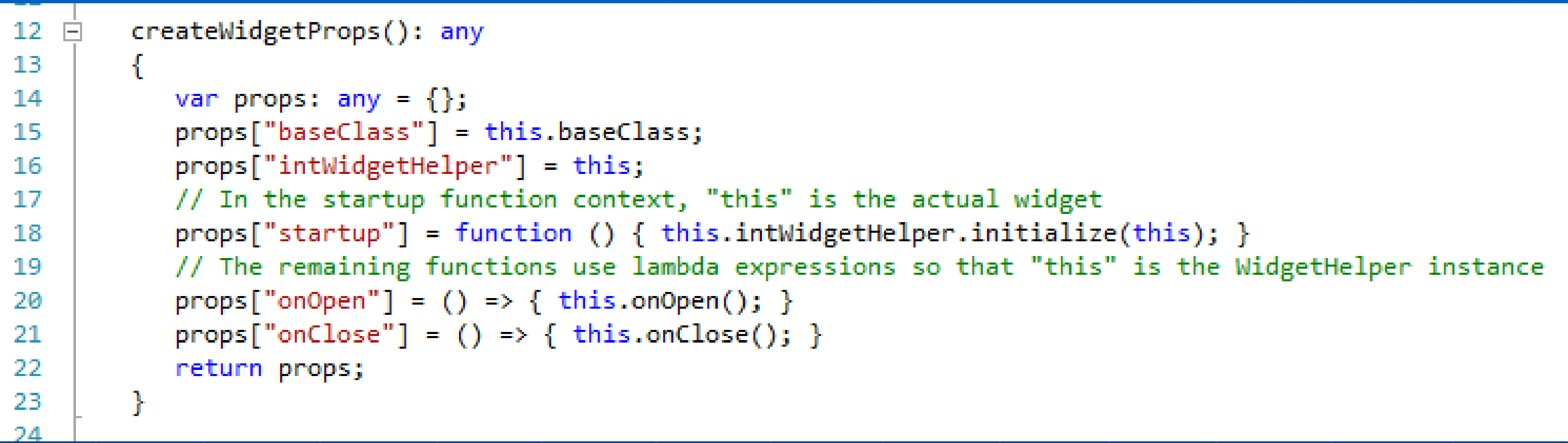
```
define(['dojo/_base/declare', 'jimu/BaseWidget', "./ts/WidgetHelper"],
function (declare, BaseWidget, WidgetHelper)
{
    var helper = new WidgetHelper();
    var props = helper.createWidgetProps();
    return declare([BaseWidget], props);
});
```

WidgetHelper.js



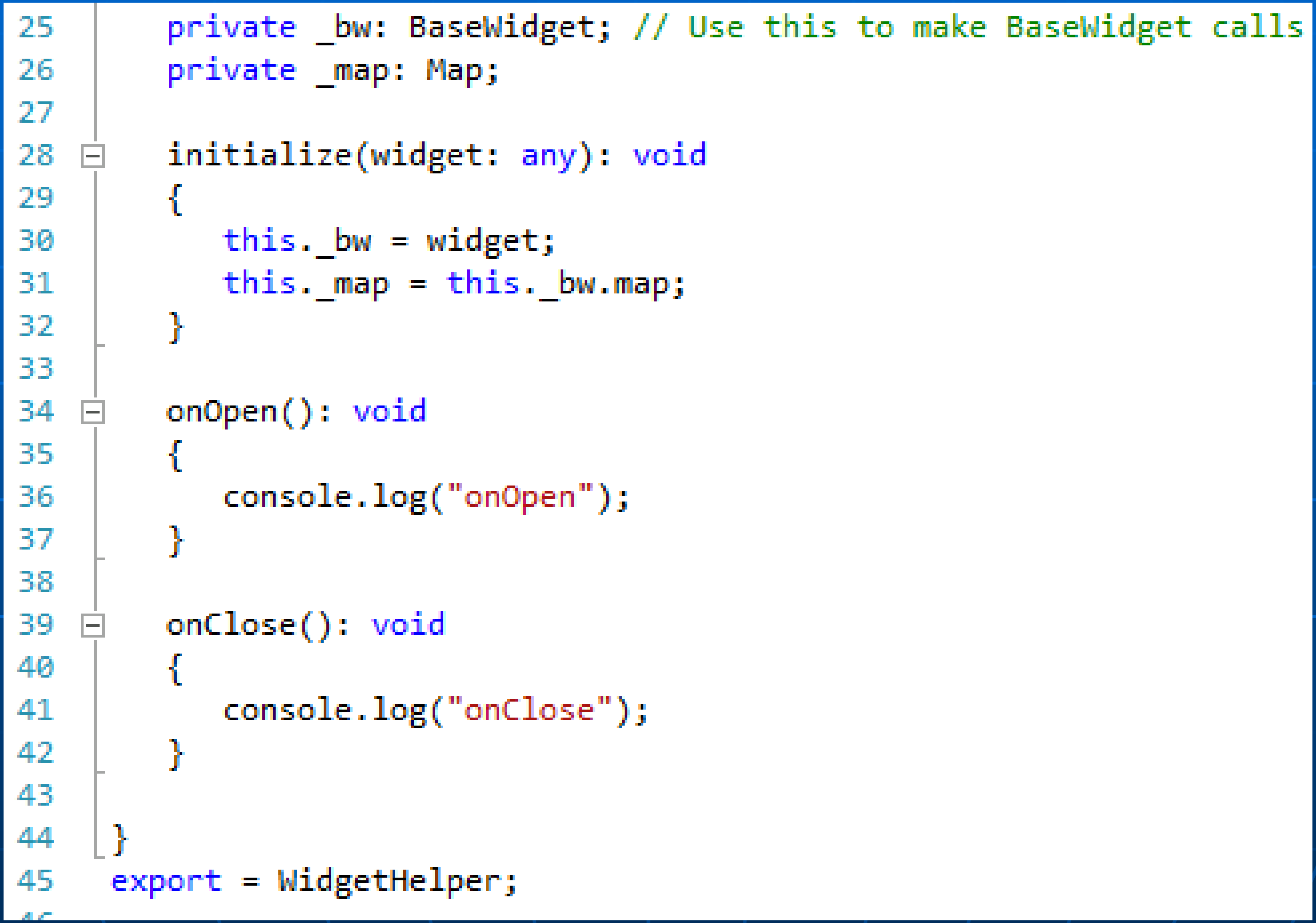
```
/// <reference path="./jimu.d.ts" />
/// <reference path="./arcgis-js-api.d.ts" />

import BaseWidget = require("jimu/BaseWidget");
import Map = require("esri/map");

class WidgetHelper
{

    baseClass: string = "jimu-widget-customwidget";

```

WidgetHelper.js (cont'd)



```
createWidgetProps(): any
{
    var props: any = {};
    props["baseClass"] = this.baseClass;
    props["intWidgetHelper"] = this;
    // In the startup function context, "this" is the actual widget
    props["startup"] = function () { this.intWidgetHelper.initialize(this); }
    // The remaining functions use lambda expressions so that "this" is the WidgetHelper instance
    props["onOpen"] = () => { this.onOpen(); }
    props["onClose"] = () => { this.onClose(); }
    return props;
}
```

WidgetHelper.js (cont'd)

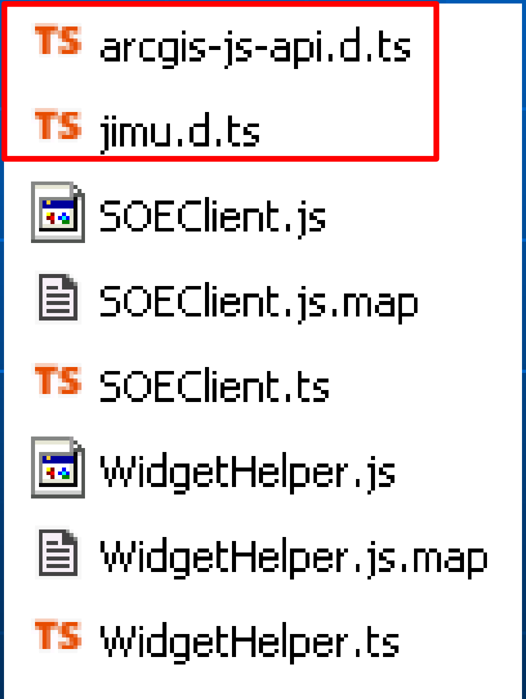


```
    private _bw: BaseWidget; // Use this to make BaseWidget calls
    private _map: Map;

    initialize(widget: any): void
    {
        this._bw = widget;
        this._map = this._bw.map;
    }

    onOpen(): void
    {
        console.log("onOpen");
    }

    onClose(): void
    {
        console.log("onClose");
    }

}
export = WidgetHelper;
```

Deployment

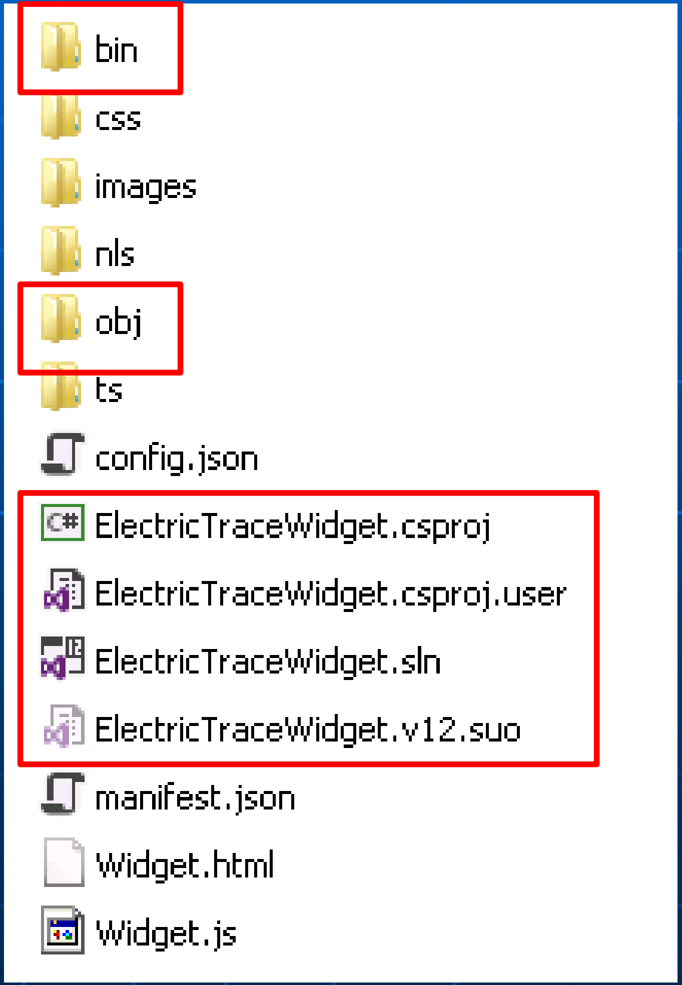
Copy the project folder to the deployment location and delete the following:

Delete any .d.ts files:



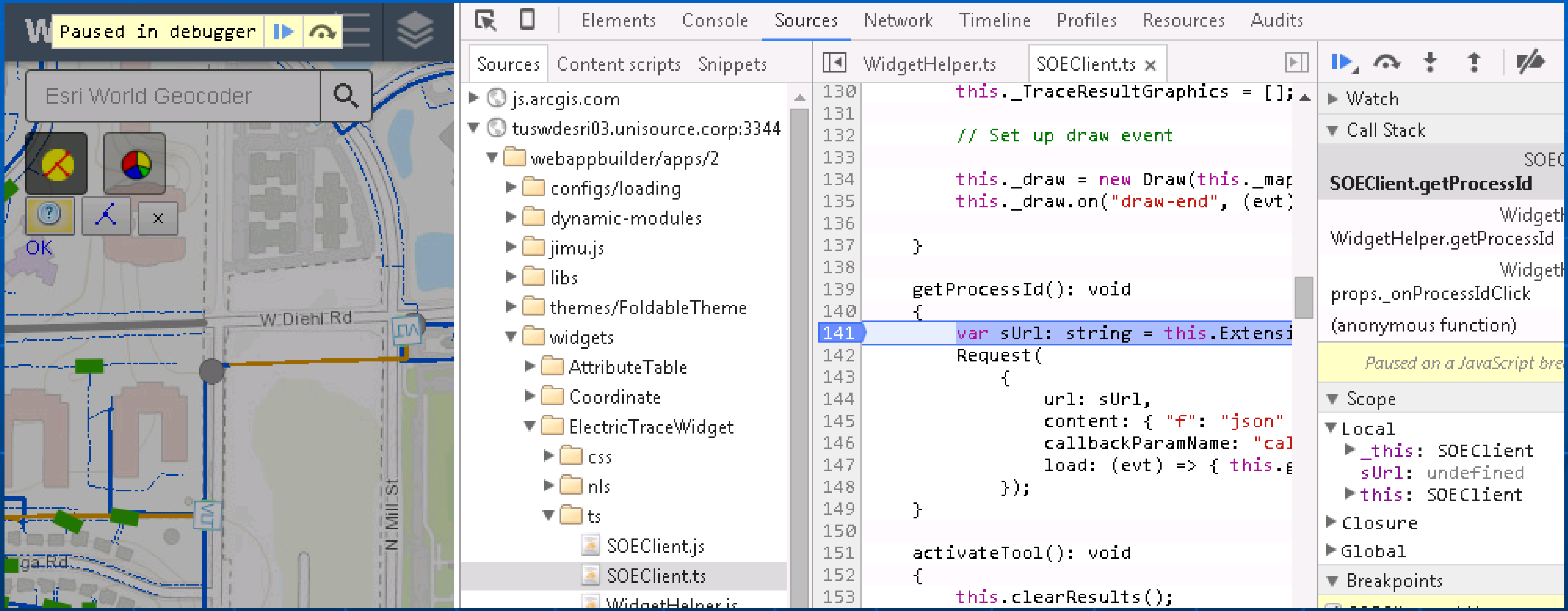
arcgis-js-api.d.ts
jimu.d.ts
SOEClient.js
SOEClient.js.map
SOEClient.ts
WidgetHelper.js
WidgetHelper.js.map
WidgetHelper.ts

Delete bin, obj, and project files:



bin
css
images
nls
obj
ts
config.json
ElectricTraceWidget.csproj
ElectricTraceWidget.csproj.user
ElectricTraceWidget.sln
ElectricTraceWidget.v12.suo
manifest.json
Widget.html
Widget.js

Debugging



```
        this._TraceResultGraphics = [];

        // Set up draw event

        this._draw = new Draw(this._map
        this._draw.on("draw-end", (evt)

    }

    getProcessId(): void
    {
        var sUrl: string = this.Extensi
        Request(
            {
                url: sUrl,
                content: { "f": "json"
                callbackParamName: "cal
                load: (evt) => { this.g
            });
    }

    activateTool(): void
    {
        this.clearResults();
```

Call Stack

SOEC
SOEClient.getProcessId
Widget
WidgetHelper.getProcessId
Widget
props._onProcessIdClick
(anonymous function)

Paused on a JavaScript bre

Scope

Local
_this: SOEClient
sUrl: undefined
this: SOEClient
Closure
Global

4:25:14 PM

DEMO 1: Converting an existing app to a widget

4:25:14 PM

What About Dojo?

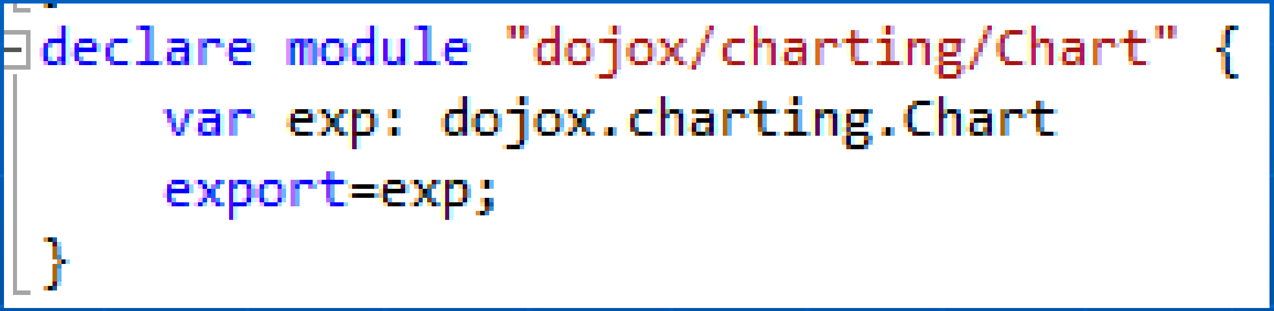
- Type definition files are at the DefinitelyTyped repository:

  github.com/DefinitelyTyped/DefinitelyTyped/tree/master/dojo

- However, module declarations may need editing to get your project to compile

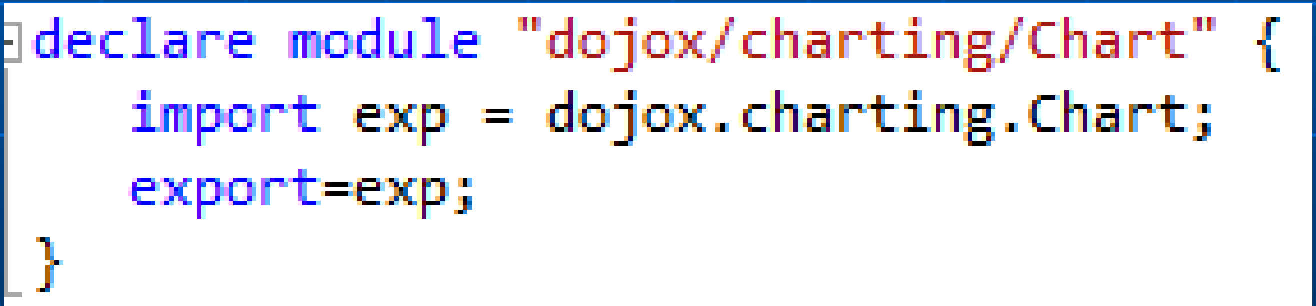
Dojo Module Declarations

Wrong:



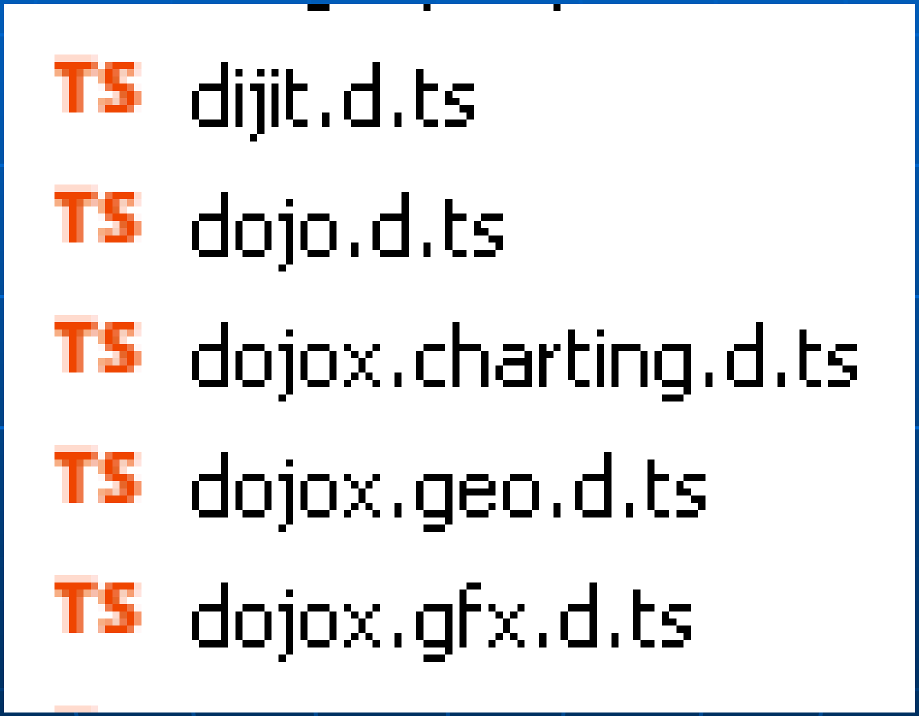
```
declare module "dojox/charting/Chart" {
    var exp: dojox.charting.Chart
    export=exp;
}
```

Right:



```
declare module "dojox/charting/Chart" {
    import exp = dojox.charting.Chart;
    export=exp;
}
```

Definition Files Required to Support Charting



- TS dijit.d.ts
- TS dojo.d.ts
- TS dojox.charting.d.ts
- TS dojox.geo.d.ts
- TS dojox.gfx.d.ts

DEMO 2:
Creating a custom pie chart widget

Questions?

- Mark Cederholm
  mcederholm@uesaz.com
- This presentation and sample code may be downloaded at:

http://www.pierssen.com/arcgis10/server.htm

ALSO SEE:
Web AppBuilder for ArcGIS: Development Tools and Techniques
Friday 8:30 AM – 9:30 AM/1:00 PM – 2:00 PM, Smoketree A/B/C/D/E

4:28:20 PM