

Drew Skwiers-Koballa



Azure Data Studio Extension Development



- Visit the Sponsor Booths
- Lots of Great Raffle Prizes!
- Get your parking paid via Sponsor Bingo

Thank you Sponsors!

Platinum Sponsor:



Gold Sponsors:



Global Alliance Partners:



Microsoft Azure



PASS



PASSMN – News/Info



Thanks to all our sponsors of 2019!

We need Speakers & Sponsors for 2020 PASSMN Meetings!

- Sign up to present at one of the monthly meetings!

Monthly Meetup: 3rd Tuesday of Each Month (except Oct) at Microsoft MTC in Edina (food usually provided)

Signup on Meetup: <https://www.meetup.com/MN-SQL-Server-User-Group-PASSMN/>

Board Member Elections in November/December:

- Your chance to help out the MN SQL community!



The logo for PASS Summit 2019 features a stylized 'A' shape on the left, composed of two overlapping curved lines in red and teal. To the right of this is a small icon of a star or flower with five points in red, green, blue, and purple, followed by the word 'PASS' in a black, sans-serif font. Below 'PASS' is the word 'SUMMIT' in a large, bold, black, sans-serif font, and to its right is the year '2019' in the same large, bold, black, sans-serif font.

PASS SUMMIT 2019

Join the brightest data professionals focused on the
Microsoft Data Platform!

November 3th Through November 8th

- Pre-Conference Sessions – Monday/Tuesday
- Conference – Wednesday through Friday



SQLSaturday #913 – After Party



Location: 4th Floor of Mall of America

Time: 6:30PM – 10PM

There will be drinks and appetizers as well as free game cards and bowling!

Hang out with some new friends you've made.



Agenda

1. Why Develop an Extension?
2. Extension Development Framework*
3. Building Extension Features*
4. Package and Publish*
5. Wrap Up

*Demo interspersed in 2-4

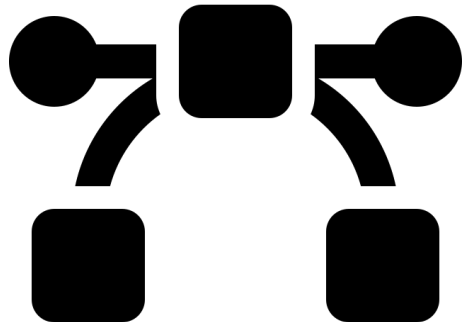


Why Develop an Extension?



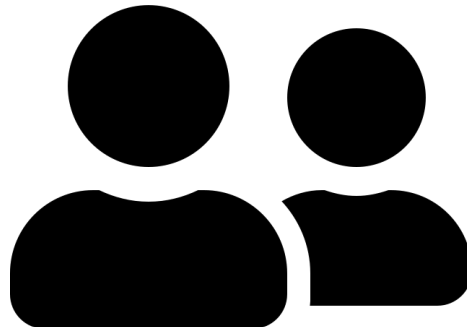
Improve Your Workflow

Enterprise process
Specific shortcut



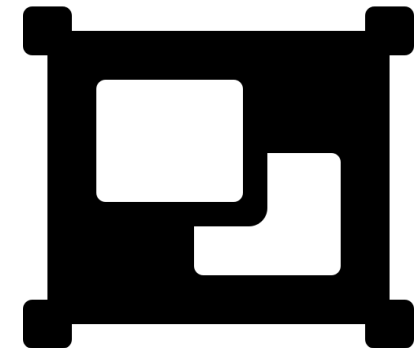
Share with the SQL Community

Better tools
Best practices



Close To Your Comfort Zone

Development
related to data
platform

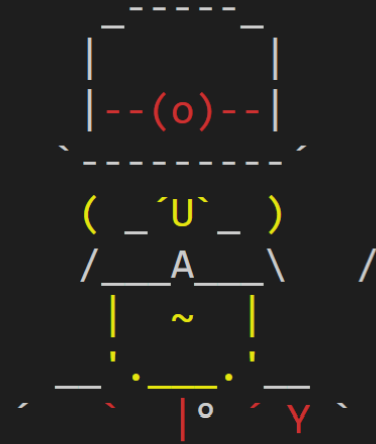


Extension Types

Chose One for a Template:

- TypeScript/JavaScript
- Dashboard Insight
- Color Theme
- Code Snippets
- Keymap
- Extension Pack
- Language Pack

```
PS C:\Users\drewk> yo azuredatstudio
```



Welcome to the Azure
Data Studio Extension
generator!

? What type of extension do you want to create?

> New Extension (TypeScript)

New Extension (JavaScript)

New Dashboard Insight

New Color Theme

New Code Snippets

New Keymap

New Extension Pack

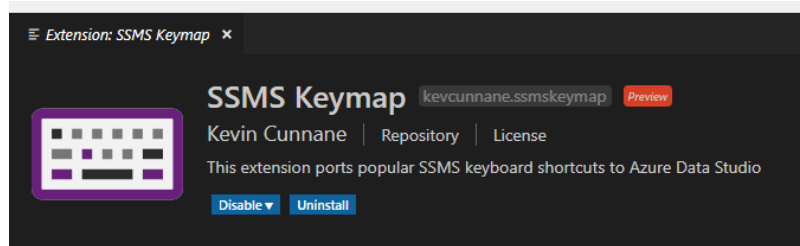
(Move up and down to reveal more choices)



Type Examples

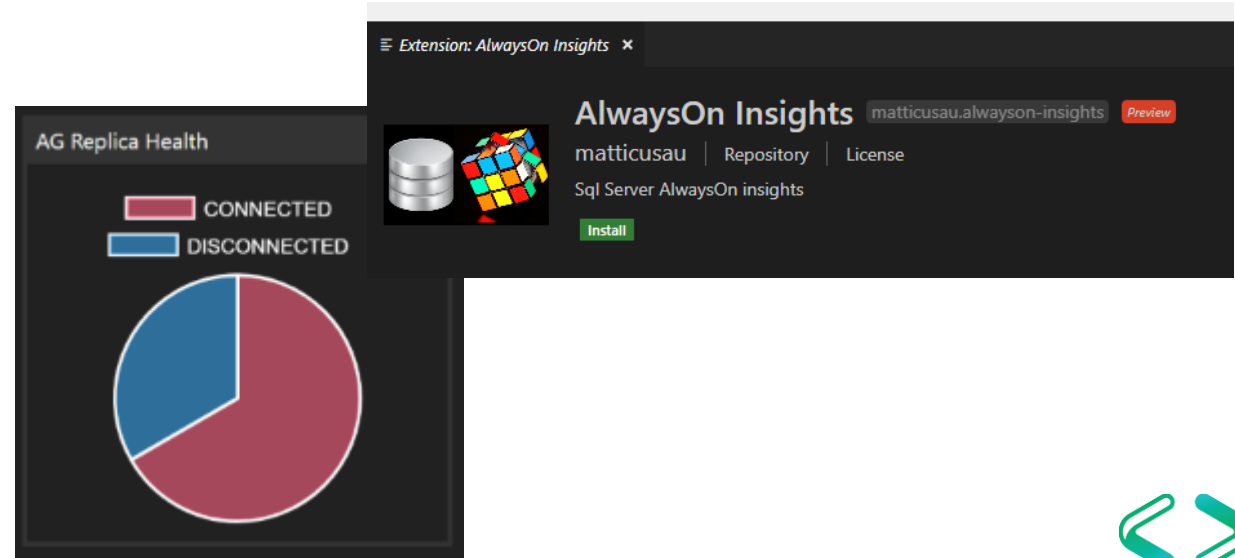
Keymap

Ties keystrokes to commands



Dashboard/Insight

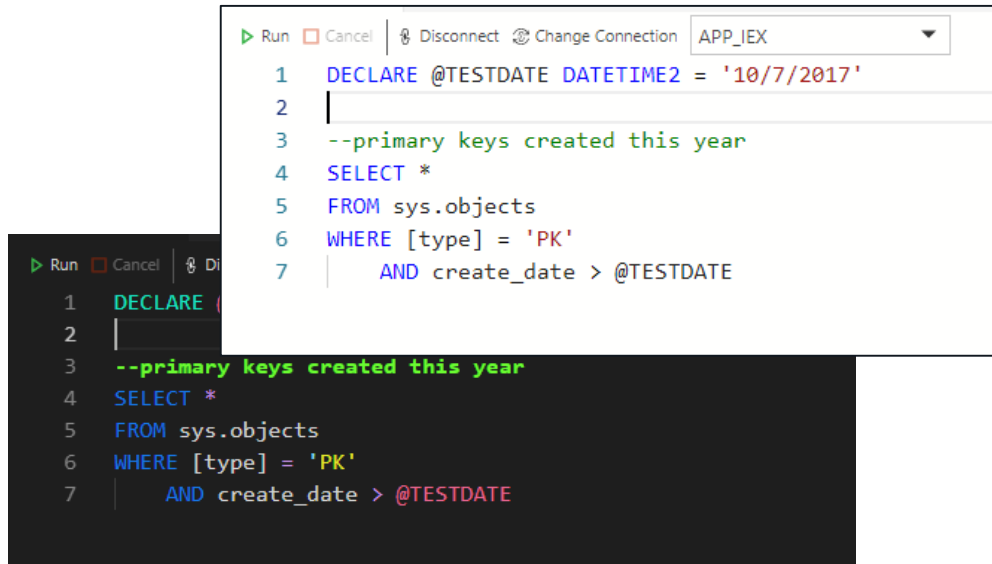
Packaged visualizations for database or server dashboards



Type Examples

Color Theme

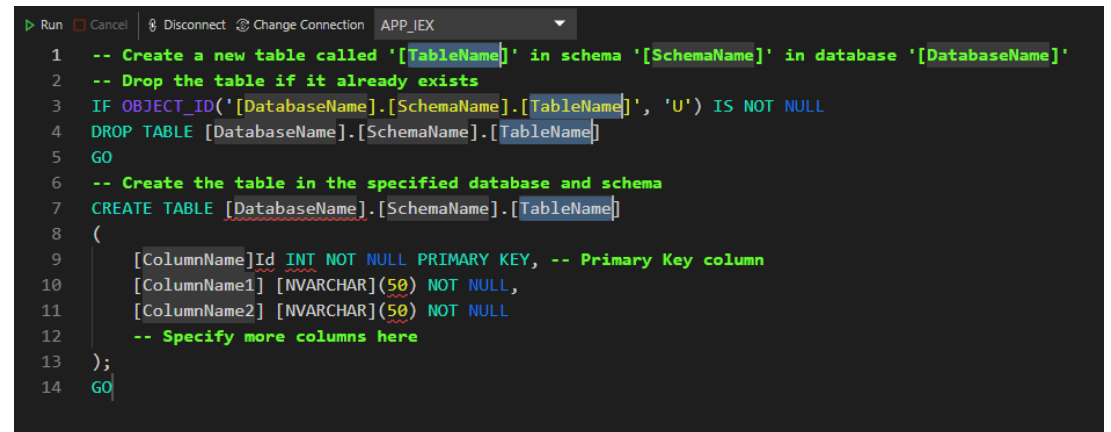
Editor color customization



```
1 DECLARE @TESTDATE DATETIME2 = '10/7/2017'
2
3 --primary keys created this year
4 SELECT *
5 FROM sys.objects
6 WHERE [type] = 'PK'
7     AND create_date > @TESTDATE
```

Snippets

Packaged TSQL with tabstops



```
1 -- Create a new table called '[TableName]' in schema '[SchemaName]' in database '[DatabaseName]'
2 -- Drop the table if it already exists
3 IF OBJECT_ID('[DatabaseName].[SchemaName].[TableName]', 'U') IS NOT NULL
4 DROP TABLE [DatabaseName].[SchemaName].[TableName]
5 GO
6 -- Create the table in the specified database and schema
7 CREATE TABLE [DatabaseName].[SchemaName].[TableName]
8 (
9     [ColumnName]Id INT NOT NULL PRIMARY KEY, -- Primary Key column
10    [ColumnName1] [NVARCHAR](50) NOT NULL,
11    [ColumnName2] [NVARCHAR](50) NOT NULL
12    -- Specify more columns here
13 );
14 GO
```



Type Examples

Extension Pack

Admin Pack for SQL Server is a collection of extensions that you will download the following

- [SQL Server Agent](#)
 - List SQL Server Agents
 - View Job History with details
 - Basic Job Control tools
- [SQL Server Profiler](#)
 - Browse through extended events
 - View and manage extended events
 - Filter search of events
- [SQL Server Import](#)
 - Use the Import Wizard
- [SQL Server dacpac](#)
 - Use the Data-Tier ...

TypeScript/JavaScript

`sp_executesql`

2

SQL

sp_executesql to SQL

Pejman Nikram | Repository

Convert sp_executesql to sql

Install



Combine Scripts

Bateleur IO | Repository

Create a single combined script

Install





Extension Development Framework

Your Prerequisites

Workstation OS

Windows, Mac, or Linux OS
**ChromeOS*

Knowledge Requirements

Beginner Git
Beginner TypeScript





A Quick Poll: Do You Want To Play Along?

1. Yes
2. No
3. Undecided

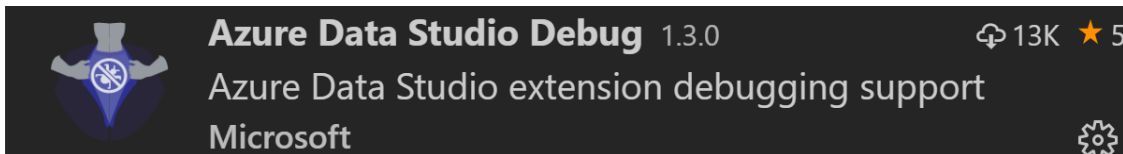
System Prerequisites

Applications

[VS Code](#)

[Azure Data Studio Debug](#)
(extension for VS Code)

[Azure Data Studio](#)



Development Tools

[Git](#)

[NodeJS](#)



Checking Installs

Powershell

```
PS C:\Users\drewk> git --version  
git version 2.22.0.windows.1
```

```
PS C:\Users\drewk> node -v  
v10.16.0
```

Bash

```
drewsk@drewsk-2018:~$ git --version  
git version 2.17.1  
drewsk@drewsk-2018:~$
```

```
drewsk@drewsk-2018:~$ nodejs -v  
v8.10.0  
drewsk@drewsk-2018:~$
```

You don't need command line experience to get started with extension development.



System Prerequisites

Install Through *npm*

- TypeScript
 - JavaScript with type checking + more
- Yeoman Extension Generator
 - *yo*
 - Open source extension template builder
- VS Code Extension Manager
 - *vsce*
 - Packages extension into .vsix for installing into Azure Data Studio

```
npm install -g typescript
```

```
npm install -g yo azuredatstudio
```

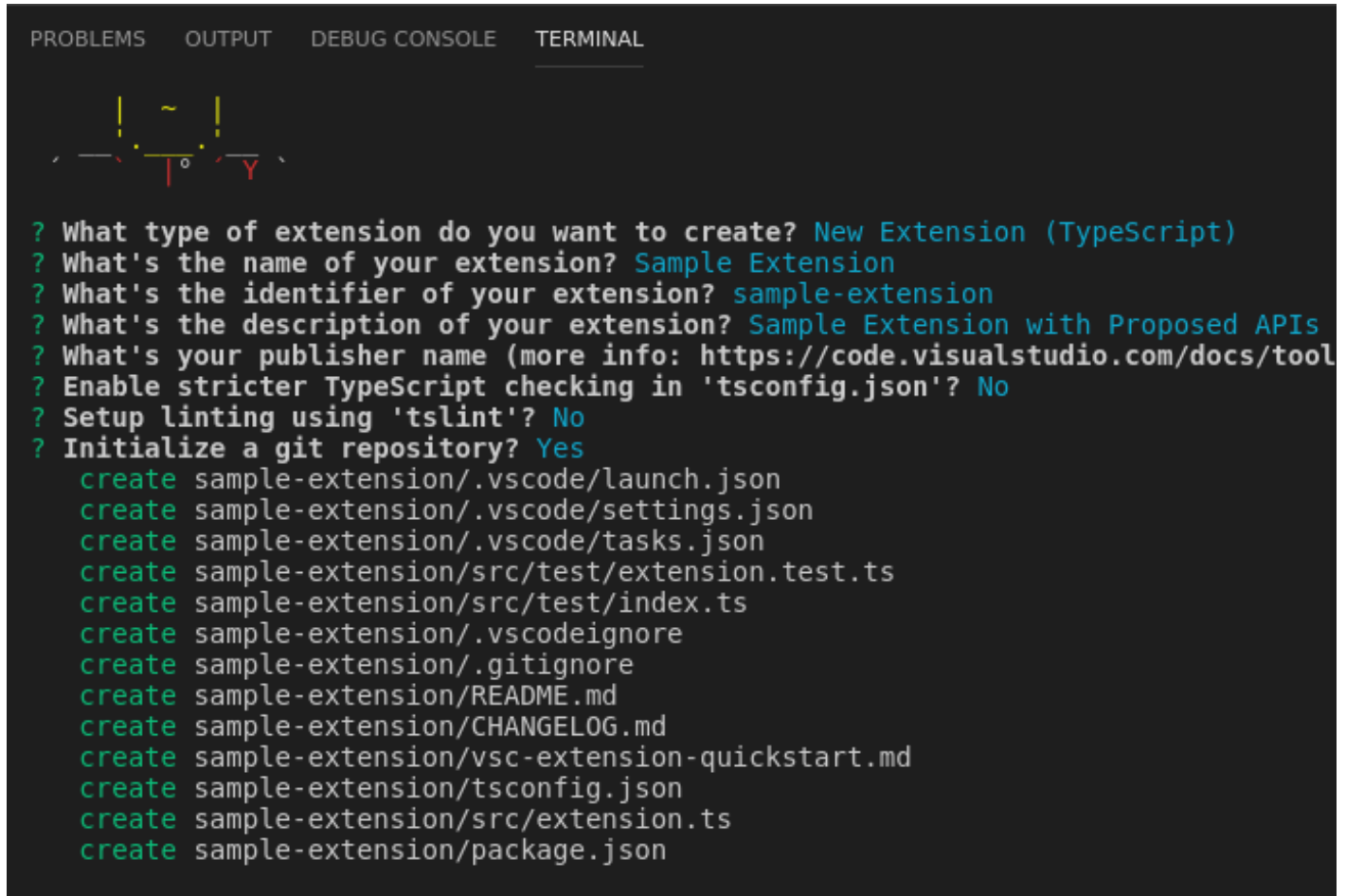
```
npm install -g vsce
```



Yo

yo azuredatastudio

- For this sample, leave the *Typescript Extension* selection
- Enter a name, identifier, description, and your publishing name
- Decline stricter checking and linting
- Select the initialization of a Git repository



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
? What type of extension do you want to create? New Extension (TypeScript)
? What's the name of your extension? Sample Extension
? What's the identifier of your extension? sample-extension
? What's the description of your extension? Sample Extension with Proposed APIs
? What's your publisher name (more info: https://code.visualstudio.com/docs/tool
? Enable stricter TypeScript checking in 'tsconfig.json'? No
? Setup linting using 'tslint'? No
? Initialize a git repository? Yes
create sample-extension/.vscode/launch.json
create sample-extension/.vscode/settings.json
create sample-extension/.vscode/tasks.json
create sample-extension/src/test/extension.test.ts
create sample-extension/src/test/index.ts
create sample-extension/.vscodeignore
create sample-extension/.gitignore
create sample-extension/README.md
create sample-extension/CHANGELOG.md
create sample-extension/vsc-extension-quickstart.md
create sample-extension/tsconfig.json
create sample-extension/src/extension.ts
create sample-extension/package.json
```



Demo Outline

We're Going to Explore:

- Activation events: when the extension is “started”, code entry point
- Contribution points: additions to the application interface
- VS Code APIs + Azure Data Studio APIs
- *Alternative NodeJS packages*
- *Extended extension architecture*



Initial Extension Files

```
.
├── .vscode
│   ├── launch.json
│   └── tasks.json
├── .gitignore
├── README.md
├── src
│   └── extension.ts
├── package.json
└── tsconfig.json
```

.vscode folder controls how VS Code interacts with the project

README.md stores documentation

Extension.ts contains code entry point(s)

Package.json = Extension Manifest



Package.json

Activation Events

```
13     ],  
14     "activationEvents": [  
15         "onCommand:extension.sayHello",  
16         "onCommand:extension.showCurrentConnection"  
17     ],  
18     "main": "out/extension"
```

When the extension is loaded – consumes CPU/memory

- onCommand
- workspaceContains
- onFileSystem
- onView
- * (*startup*)



Package.json

Contribution Points

Commands

Configuration

Menu

Keybindings

Themes

Snippets

Dashboard

Container

```
"contributes": {
  "commands": [ ...
  ],
  "keybindings": [ ...
  ],
  "configuration": [
    {
      "title": "NewQueryTemplate",
      "properties": {
        "newquerytemplate.DefaultQueryTemplate": {
          "type": "array",
          "default": [
            "--set a default new query template with",
            "",
            ""
          ],
          "description": "Query text to insert into n
        },
        "newquerytemplate.DefaultQueryLine": {
          "type": "number",
          "default": -1
        },
        "newquerytemplate.DefaultQueryCharacter": {
          "type": "number",
          "default": -1
        }
      }
    }
  ]
}
```

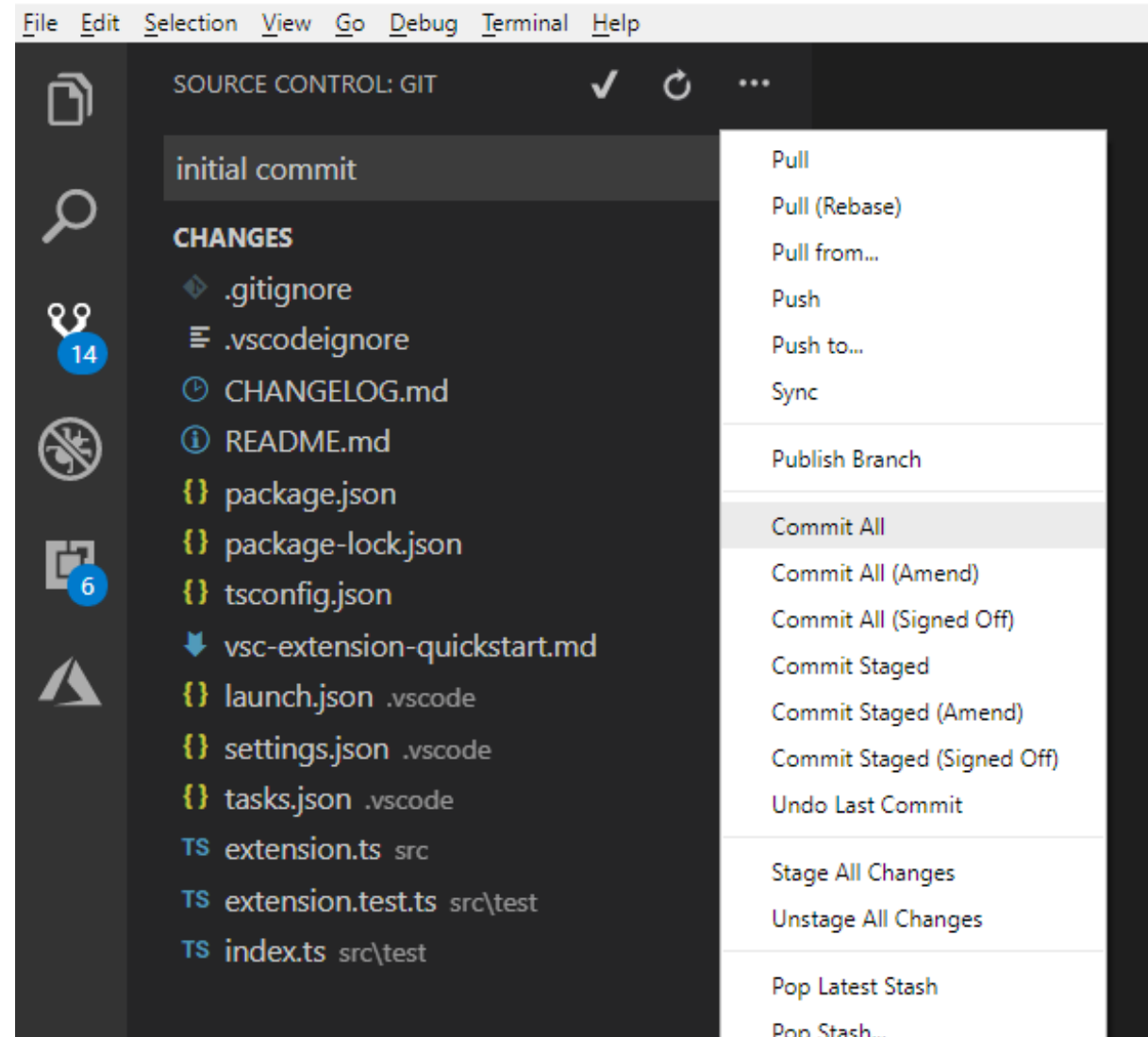


Initial Commit

- Open the folder for your extension in VS Code
- Stage and Commit all changes

Optional:

- Create a repository on GitHub without initializing
- Run command to add Git remote in VS Code

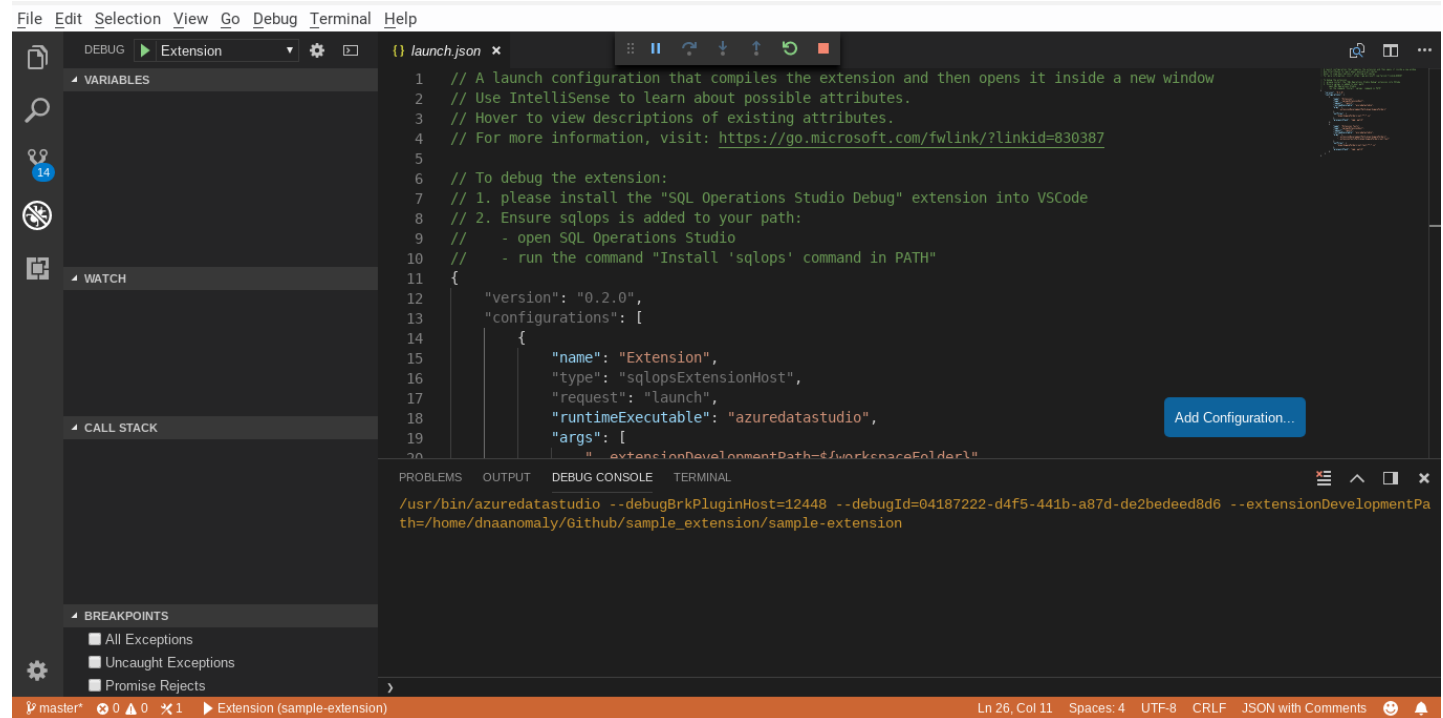


Test Extension

Select Start Debugging from the Debug menu

A special *Extension Development* instance of ADS opens

Test the extension from the command palette (ctrl+shift+P)





Building Extension Features

Azure Data Studio APIs

A few examples...



connection

ConnectionProfile
getCurrentConnection()
ServerInfo

objectexplorer

NodeInfo
Traverse object tree
findNodes()



Azure Data Studio APIs

DataProvider extends to:

ConnectionProvider – server connections

MetadataProvider – object info

ScriptingProvider – “script as”

QueryProvider – execute queries

ProfilerProvider – extended events

AgentServicesProvider – SQL agent

BackupProvider – backup and restore



VS Code APIs



window

Text editors

Terminals

User inputs

workspace

Currently open
folder

FileSystem

tasks

Background
scripts/processes

```
const editor = vscode.window.activeTextEditor;  
let cursorPosition = editor.selection.start;  
let lineCount = editor.document.lineCount;
```

More Info: <https://code.visualstudio.com/api/references/vscode-api>



Proposed APIs

azdata.proposed.d.ts

- APIs in preview
- Subject to breaking changes

Using Proposed APIs

- To install from command line after Yeoman generator:
npm run proposedapi
- To install manually:
Copy file to *src/typings* folder



<https://github.com/microsoft/azuredatatstudio/blob/master/src/sql/azdata.proposed.d.ts>



NodeJS Packages

Additional Functionality

Parsing SQL, JSON, etc

HTTP requests

Filesystem access

Operating system info

npm/yarn

Package managers for NodeJS:

- Yarn
- npm

Start here: <https://www.npmjs.com/>



A large, teal-colored abstract graphic on the left side of the slide. It consists of several thick, curved lines that sweep from the top left towards the bottom right, creating a sense of motion and flow. The lines are layered, with some appearing in front of others, and they all curve towards the right, pointing towards the text.

Package and Publish

Package to .vsix

- (update README.md)
- Run **vsce package** from terminal
- Load .vsix into ADS via command palette

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
1: bash
drewsk@drewsk-2018:~/Documents/Git/Projects/sample-extension$ vsce package
Executing prepublish script 'npm run vscode:prepublish'...

> sample-extension@0.0.1 vscode:prepublish /home/drewsk/Documents/Git/Projects/sample-extension
> npm run compile

> sample-extension@0.0.1 compile /home/drewsk/Documents/Git/Projects/sample-extension
> tsc -p ./

Error: Make sure to edit the README.md file before you publish your extension.
drewsk@drewsk-2018:~/Documents/Git/Projects/sample-extension$
```

```
SQLOPS-FIRSTRESPONDERKIT
├─ .vscode
├─ images
├─ node_modules
├─ out
├─ src
├─ typings
│   └─ sqlops.proposed.d.ts
├─ .gitignore
├─ .vscodeignore
├─ CHANGELOG.md
├─ extension_listing.json
├─ firstresponderkit-0.1.1.vsix
├─ firstresponderkit-0.2.0.vsix
├─ LICENSE
├─ package-lock.json
├─ package.json
├─ README.md
├─ resources.txt
├─ tsconfig.json
└─ vsc-extension-quickstart.md
```



Options for Publishing

A philosophical question:

Open Source

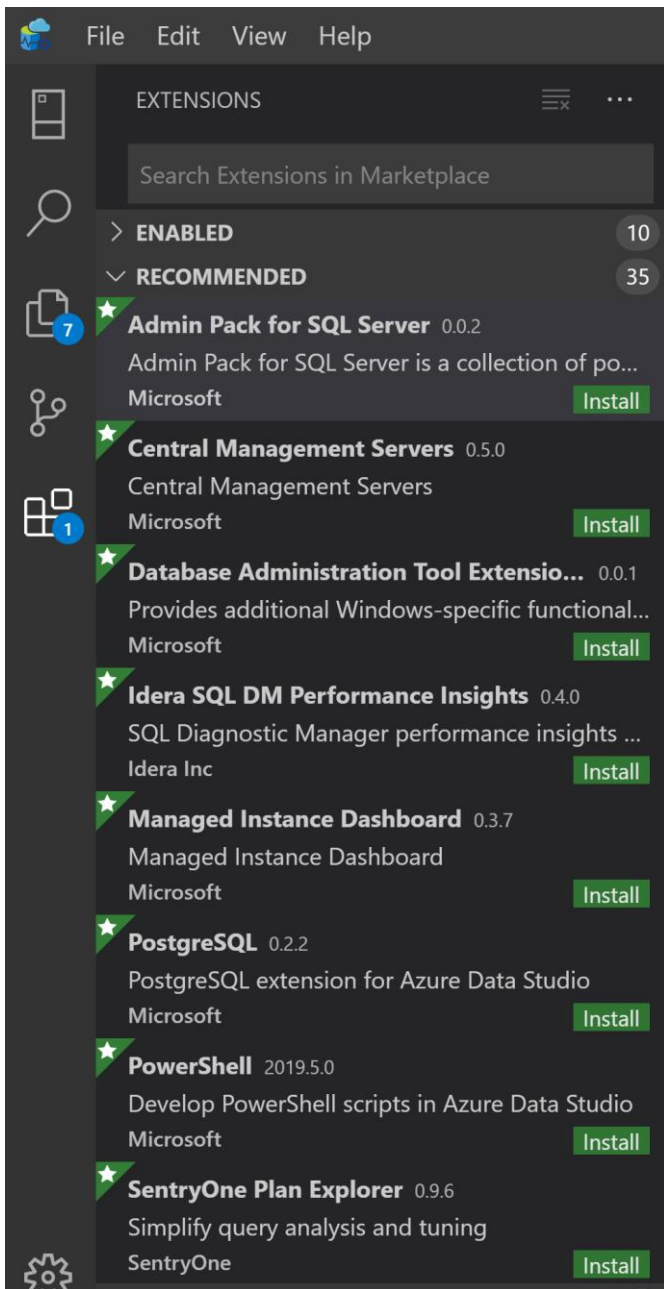
Others can view your code, contribute improvements or fixes

Closed Source

Protect your intellectual property

Others can download and install the .vsix





Azure Data Studio Marketplace

Pull Request to release/extensions

Update [extensionsGallery.json](#)



extensionsGallery.json

files

**Values don't have
to be links to
GitHub**

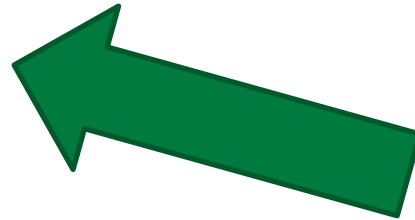
- SQLOps.DownloadPage
- VisualStudio.Services.Icons.Default, *optional*
- VisualStudio.Services.Content.Details
- VisualStudio.Code.Manifest
- VisualStudio.Services.Content.License



extensionsGallery.json

properties

- VisualStudio.Code.ExtensionDependencies
- VisualStudio.Services.Links.Source, *optional*
- AzDataEngine
- VisualStudio.Code.Engine



**Set minimum
versions**





Wrap Up

Key Takeaways

Building an Extension in Azure Data Studio

- You can do it.
- VS Code extension + Azure Data Studio APIs
- Extension functionality can derive from both NodeJS and extended capabilities
- Extensions can be monetized and/or proprietary
- Open source contributions elevate the data platform community's capabilities



Additional Resources

- https://github.com/dzsquared/AzureDataStudio_ExtensionDevelopment
- <https://code.visualstudio.com/docs/extensions/overview>
- <https://medium.com/@kevcunnane/extending-sql-operations-studio-hello-connected-world-part-1-of-n-e868542c6157>
- <https://www.drewsk.tech/2018/10/sql-saturday-796-minnesota-2018/>
- <https://medium.com/ingeniouslysimple/how-we-built-an-extension-for-sql-operations-studio-f93532ce4456>
- <https://cultivatehq.com/posts/how-we-built-a-visual-studio-code-extension-for-iot-prototyping/>

Drew - @sysadmindrew drew.koballa@gmail.com

