PASS SUMMIT2019

Azure Data Studio Extension Development

Drew Skwiers-Koballa

Director of IT, Inside Edge CIS

Please silence cell phones

(and use the camera to use the QR code link to the slides)

Explore
everything PASS has to offer

Free Online Resources

Newsletters

PASS.org

24HOURS OF PASS

Free online webinar events

PASS LOCAL GROUPS

Local user groups around the world

PASS SQLSATURDAY

Free 1-day local training events

PASS MARATHON

Online special interest user groups

PASS VIRTUAL GROUPS

Business analytics training

PASS VOLUNTEERS

Get involved

Drew Skwiers-Koballa

Director of IT
Inside Edge CIS

/drew-skwiers-koballa/

@sysadmindrew

/dzsquared

Extension Developer:
First Responder Kit
Query Editor Boost

Query Editor Boost

Dynamics SL UG
Board President

Data Platform MCSE
2017-2019

Agenda

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

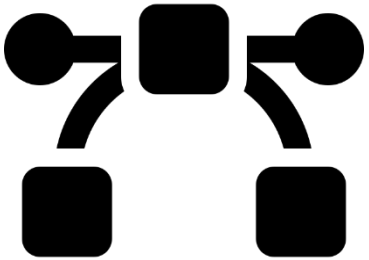
*Demo interspersed in 2-4

Why Develop an Extension?

Improve Your Workflow

Enterprise process

Specific shortcut



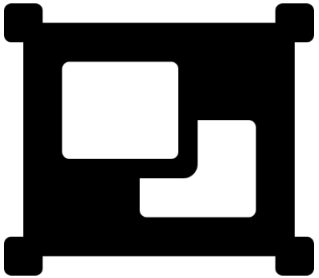
Share with the SQL Community

Better tools

Best practices

Close To Your Comfort Zone

Development related to data platform



Extension *Types*

Chose One for a Template:

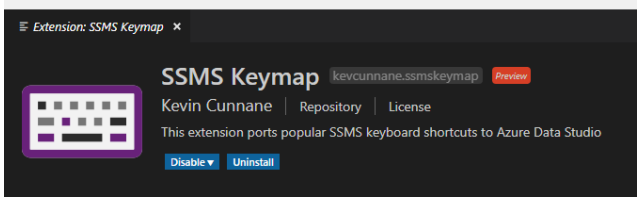
- TypeScript/JavaScript
- Dashboard Insight
- Color Theme
- Code Snippets
- Keymap
- Extension Pack
- Language Pack

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

     _-----_     ╭──────────────────────╮
    |       |    │ Welcome to the Azure │
    |--(o)--|    │ Data Studio Extension│
   `---------´   │      generator!      │
    ( _´U`_ )    ╰──────────────────────╯
    /___A___\   /
     |  ~  |
   __'.___.'__
 ´   `  |° ´ Y `

? 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

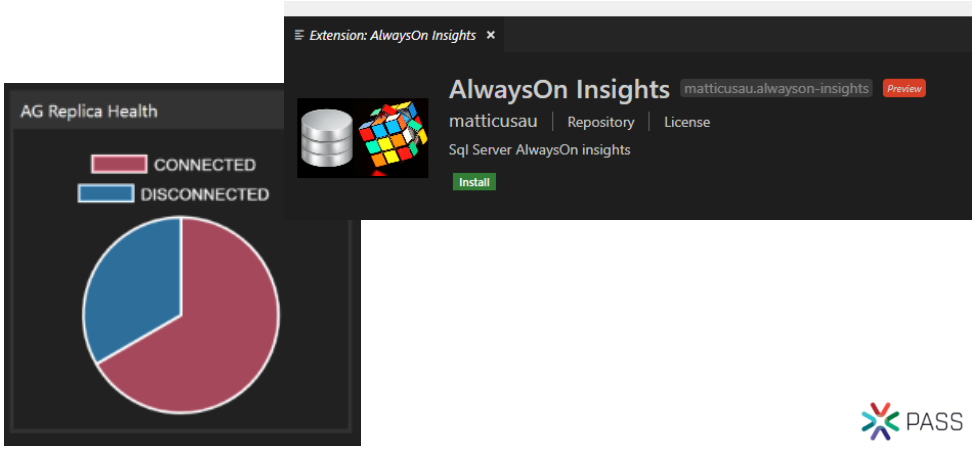
Keymap

Ties keystrokes to commands



Dashboard/Insight

Packaged visualizations for database or server dashboards



PASS

Type Examples

Color Theme

Editor color customization

Run Cancel Disconnect Change Connection APP_IEX

```
DECLARE @TESTDATE DATETIME2 = '10/7/2017'

--primary keys created this year
SELECT *
FROM sys.objects
WHERE [type] = 'PK'
    AND create_date > @TESTDATE
```

Snippets

Packaged TSQL with tabstops

Run Cancel Disconnect Change Connection APP_IEX

```
-- Create a new table called '[TableName]' in schema '[SchemaName]' in database '[DatabaseName]'
-- Drop the table if it already exists
IF OBJECT_ID('[DatabaseName].[SchemaName].[TableName]', 'U') IS NOT NULL
DROP TABLE [DatabaseName].[SchemaName].[TableName]
GO
-- Create the table in the specified database and schema
CREATE TABLE [DatabaseName].[SchemaName].[TableName]
(
    [ColumnName]Id INT NOT NULL PRIMARY KEY, -- Primary Key column
    [ColumnName1] [NVARCHAR](50) NOT NULL,
    [ColumnName2] [NVARCHAR](50) NOT NULL
    -- Specify more columns here
);
GO
```

Type Examples

Extension Pack

Admin Pack for SQL Server is a c
you will download the following

- SQL Server Agent
  - List SQL Server Age
  - View Job History w
  - Basic Job Control t
- SQL Server Profiler
  - Browse through ex
  - View and manage
  - Filter search of eve
- SQL Server Import
  - Use the Import Fla
- SQL Server dacpac
  - Use the Data-Tier A

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

PASS

Extension Development Framework

Your Prerequisites

Workstation OS

Windows, Mac, or Linux OS

**ChromeOS*

Knowledge Requirements

Beginner Git

Beginner TypeScript
or Similar Language

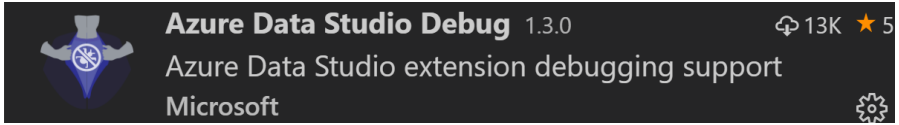
System Prerequisites

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

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

System Prerequisites

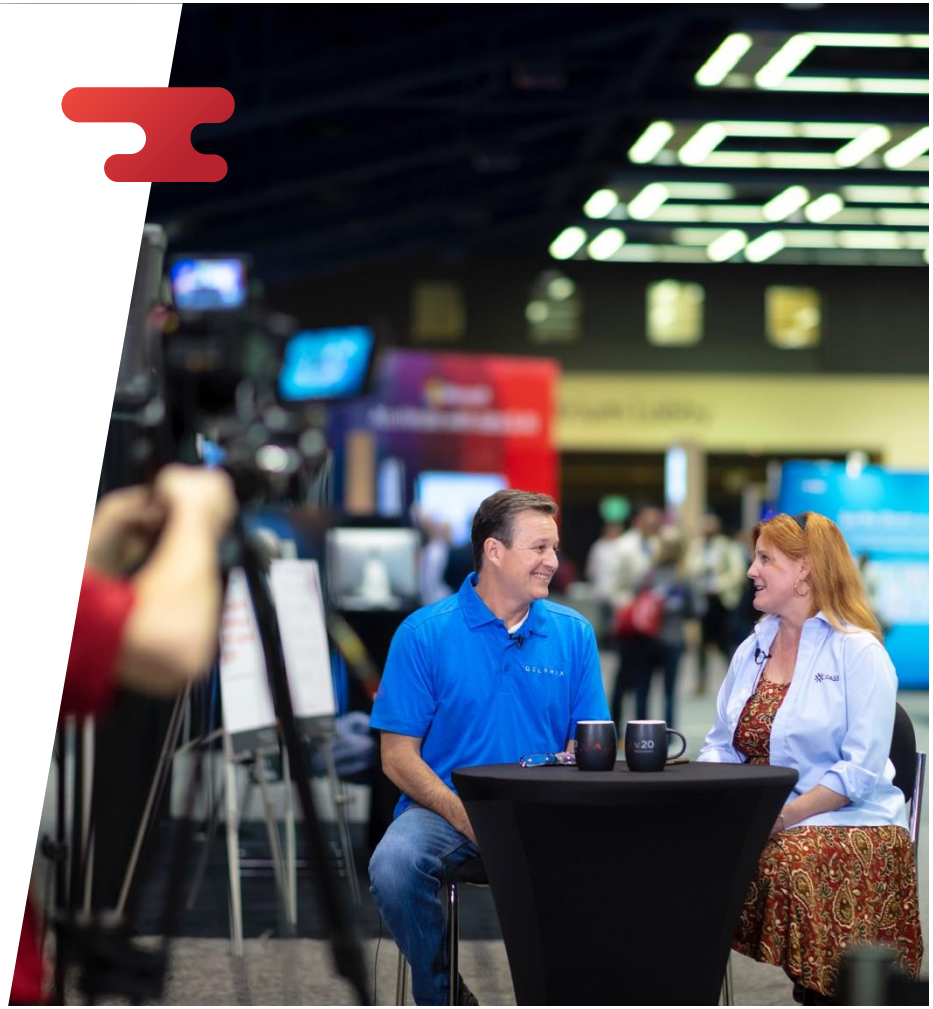
Install Through *npm*

```
npm install -g typescript

npm install -g yo azuredatastudio

npm install -g vsce
```

- 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

DEMO

Let’s Build an Extension



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

     |  ~  |
 , __`  |°  ´Y `

? 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

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

Activation Events

When the extension is loaded – consumes CPU/memory

- onCommand
- workspaceContains
- onFileSystem
- onView
- * *(only when necessary)*

PASS

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 wit
                        "",
                        ""
                    ],
                    "description": "Query text to insert into n
                },
                "newquerytemplate.DefaultQueryLine": {
                    "type": "number",
                    "default": -1
                },
                "newquerytemplate.DefaultQueryCharacter": {
                    "type": "number",
                    "default": -1
                }
            }
        }
    ]
```

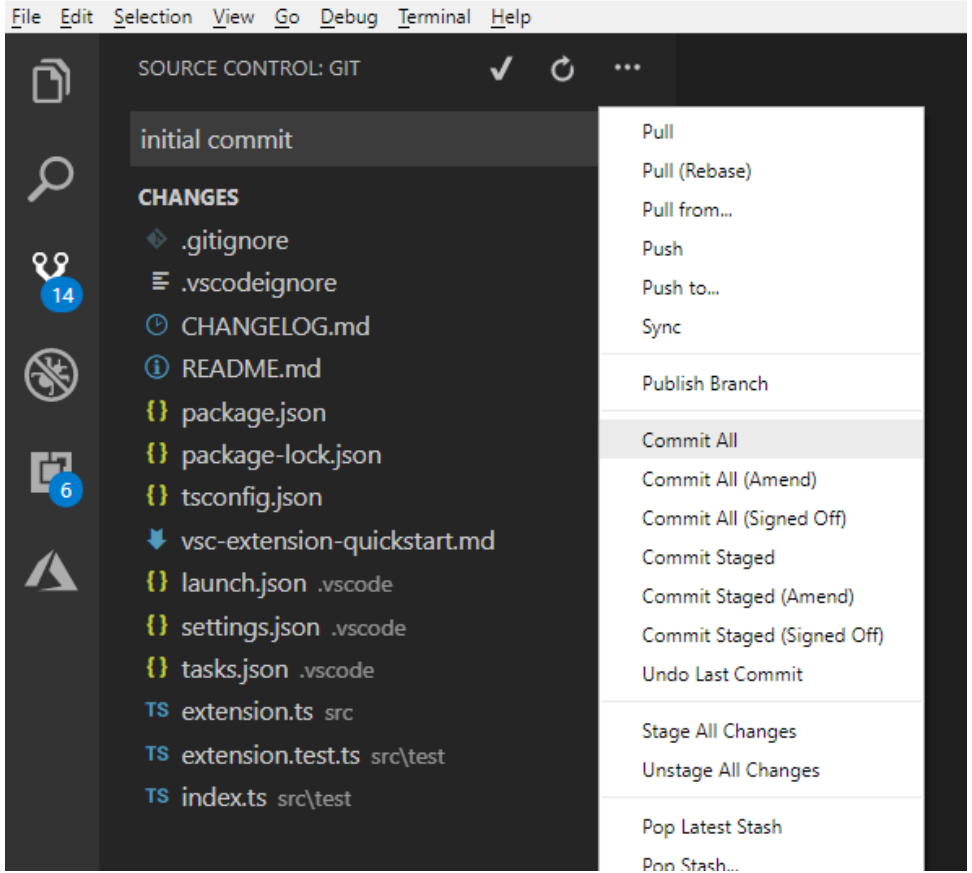
PASS

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

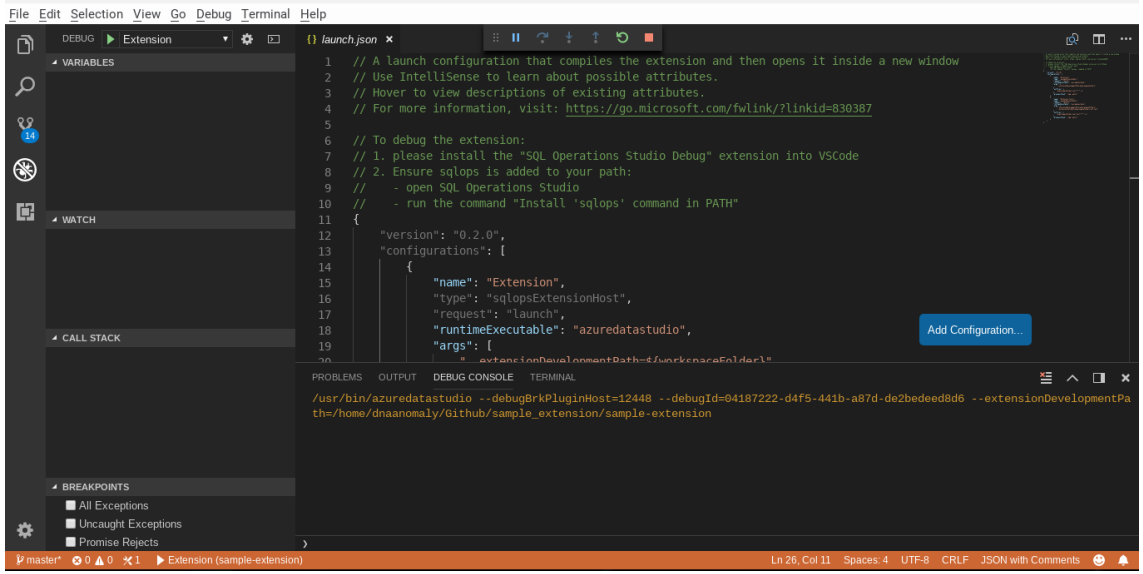


Test Extension

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)



PASS

Building Extension Features

Azure Data Studio APIs

A few examples…

connection

ConnectionProfile

getCurrentConnection()

ServerInfo

objectexplorer

NodeInfo

Traverse object tree

findNodes()

https://github.com/microsoft/azuredatastudio/blob/master/src/sql/azdata.d.ts

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

https://github.com/microsoft/azuredatastudio/blob/master/src/sql/azdata.d.ts

Let’s Try It.

How many downvotes does the StackOverflow user with the most downvotes have?

We need to run a command that pulls some data from the database.

1. Get the current connection
2. Setup the query
3. Display the results to the user

extension.ts

```
29
30  context.subscriptions.push(vscode.commands.registerCommand('extension.getDown', async () => {
31      let connection = await azdata.connection.getCurrentConnection();
32      let connectId = connection.connectionId;
33      if (connection && connectId) {
34          let query = `SELECT MAX(DOWNVOTES) AS DOWNVOTES FROM USERS`;
35
36          let connectionUri = await azdata.connection.getUriForConnection(connectId);
37          let queryProvider = azdata.dataprotocol.getProvider<azdata.QueryProvider>(connection.providerId,
            azdata.DataProviderType.QueryProvider);
38
39          let results = await queryProvider.runQueryAndReturn(connectionUri, query);
40          let cell = results.rows[0][0];
41
42          let downvotes = cell.displayValue;
43          vscode.window.showInformationMessage('The max downvotes is: '+downvotes);
44
45      }
46
47  }));
```

PASS

package.json

```
13      ],
14      "activationEvents": [
15          "onCommand:extension.sayHello",
16          "onCommand:extension.showCurrentConnection",
17          "onCommand:extension.getDown"
18      ],
19      "main": "./out/extension",
20      "contributes": {
21          "commands": [
22              {
23                  "command": "extension.sayHello",
24                  "title": "Hello World"
25              },
26              {
27                  "command": "extension.showCurrentConnection",
28                  "title": "Show Current Connection"
29              },
30              {
31                  "command": "extension.getDown",
32                  "title": "Show Max Downvotes"
33              }
34          ]
35      },
```

PASS

File Edit View Help

[Extension Development Host] - Welcome - Azure Data Studio

CONNECTIONS

SERVERS

Inside Edge

Demos

drewsktech.datab...

Tables

Views

Synonyms

Programmability

External Resour...

Storage

Security

AZURE

Welcome

Start

New connection
New query
New notebook
Open file

Recent

No recent folders

Help

Getting started
Documentation
Report issue or feature request
GitHub repository
Release notes

Show welcome page on startup

Customize

Extensions
Download extensions that you need, including the SQL Server Admin pack an...

Keyboard Shortcuts
Find your favorite commands and customize them

Color theme
Make the editor and your code look the way you love

Learn

Find and run all commands
Rapidly access and search commands from the Command Palette (Ctrl+Shift+P)

Discover what's new in the latest release
New monthly blog posts each month showcasing our new features

Follow us on Twitter
Keep up to date with how the community is using Azure Data Studio and to t...

The max downvotes is: 980920

0 0

drewsktech.database.windows.net : StackOverflow2013

1

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

PASS

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 *typings* folder

https://github.com/microsoft/azuredatastudio/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 - performance
- npm - interoperability

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

Install an Additional Node Package

npm install <package-name>

Run from the terminal

import * as <alias> from '<package-name>';

Add to the top of a Typescript file

Let’s Try It.

What operating system (OS) is the user running?

We need to run a command that is different based on the OS of the user.

1. Use an additional JS package for operating system information.
2. Use the OS type to return different information to the user.

```
import * as vscode from 'vscode';

// The module 'azdata' contains the Azure Data Studio extensibility API
// This is a complementary set of APIs that add SQL / Data-specific functionality to the app
// Import the module and reference it with the alias azdata in your code below

import * as azdata from 'azdata';
import * as os from 'os';
```

• • • • •

```
// The commandId parameter must match the command field in package.json
context.subscriptions.push(vscode.commands.registerCommand('extension.sayHello', () => {
    // The code you place here will be executed every time your command is executed

    // Display a message box to the user
    vscode.window.showInformationMessage('Hello World!');

    let hostname = os.hostname();

    if (os.type() != 'Windows_NT') {
    vscode.window.showInformationMessage('Aren\'t you glad Azure Data Studio runs on other platforms
    } else {
    vscode.window.showInformationMessage('You can still use SSMS when you have the RAM to spare!');
    }

}));
```

PASS

Make the editor and your code look the way you love

Learn

Find and run all commands
Rapidly access and search commands from the Command Palette (Ctrl+Shift+P)

Discover what's new in the latest release
New monthly blog posts each month showcasing our new features

Follow us on Twitter
Keep up to date with how the community is using Azure Data Studio and to talk directly wit

You can still use SSMS when you have the RAM to spare!

Hello World!

2

Extended Architecture

Native Code

Node binaries

.NET Core

Package with extension or install on activation

Cross Platform

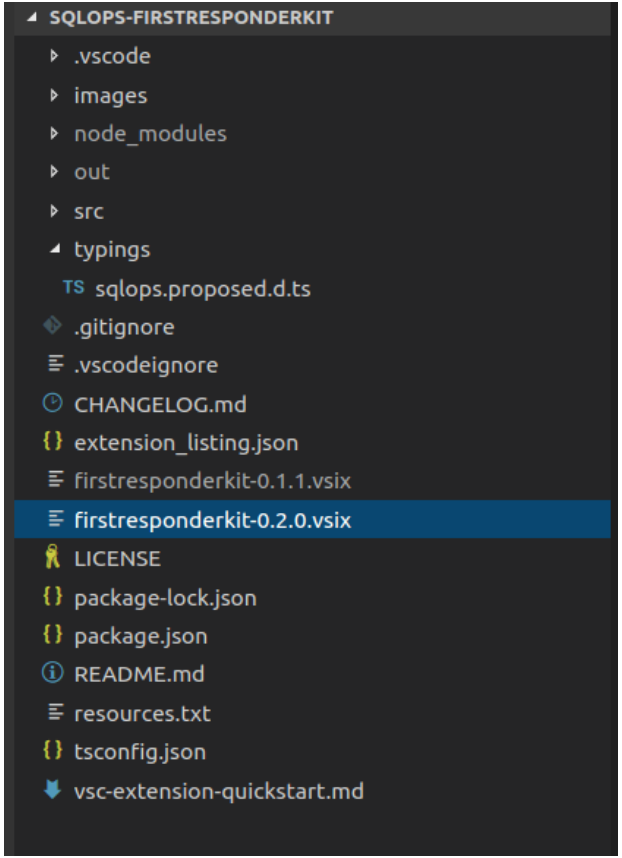
Web Framework

Support for rendering content in a Webview

Package and Publish

Package to .vsix

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



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell

```
PS C:\Users\drewk\Documents\AzureDataStudio_Dev\best-pass-summit> vsce package
Executing prepublish script 'npm run vscode:prepublish'...

> best-pass-summit@0.0.1 vscode:prepublish C:\Users\drewk\Documents\AzureDataStudio_Dev\best-pass-summit
> npm run compile


> best-pass-summit@0.0.1 compile C:\Users\drewk\Documents\AzureDataStudio_Dev\best-pass-summit
> tsc -p ./

 WARNING  A 'repository' field is missing from the 'package.json' manifest file.
Do you want to continue? [y/N] y
 DONE  Packaged: C:\Users\drewk\Documents\AzureDataStudio_Dev\best-pass-summit\best-pass-summit-0.0.1.vsix (12 files, 6.67KB)
 INFO
The latest version of vsce is 1.69.0 and you have 1.65.0.
Update it now: npm install -g vsce
PS C:\Users\drewk\Documents\AzureDataStudio_Dev\best-pass-summit>
```

Options for Publishing

A philosophical question:

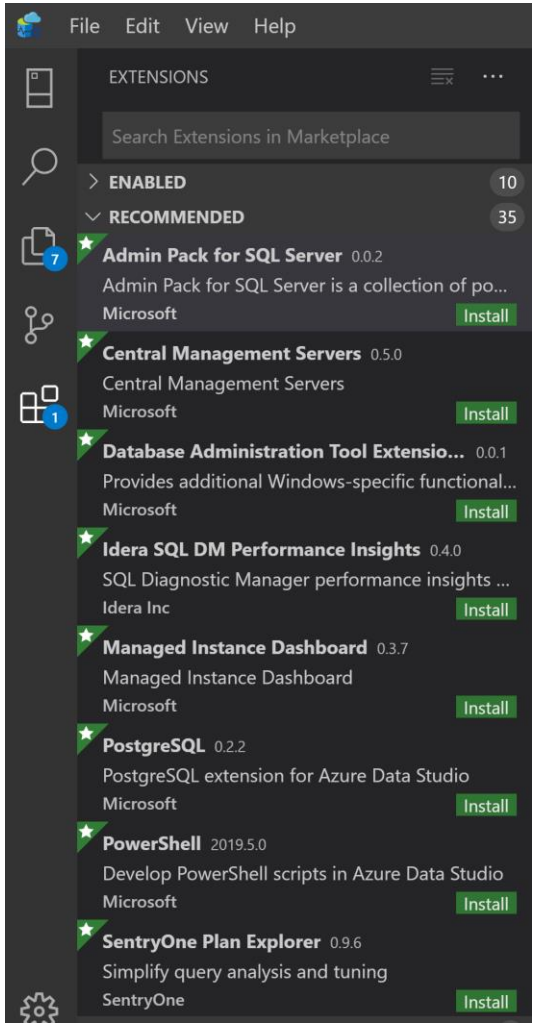
Open Source

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 branch

Update [extensionsGallery.json]

PASS

extensionsGallery.json

Values don't have to be links to GitHub

files

- 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/

Session Evaluations

Submit by 5pm Friday, November 15th to win prizes.

3 WAYS TO ACCESS

- Go to PASSsummit.com
- Download the GuideBook App and search: PASS Summit 2019
- Follow the QR code link on session signage

← Slides

Thank You

Drew
Skwiers-Koballa

@sysadmindrew

drew@drewsk.tech

PASS