

Visual Studio Code, RPG, and IBM i

Index

- Introduction Of VS Code
 - Basic features and its structure module
 - Keyboard shortcuts
- Some IBM Code Extension
 - **IBMi Languages**
 - **IBM Z Open Editor**
 - **Other Extensions – RPG , IBM git client , IBM Cloud**
 - How to install Extensions and Some other tips
- Demo*
 - User Interface, Some shortcuts
 - IBM I
 - My Extension

Introduction on VS Code

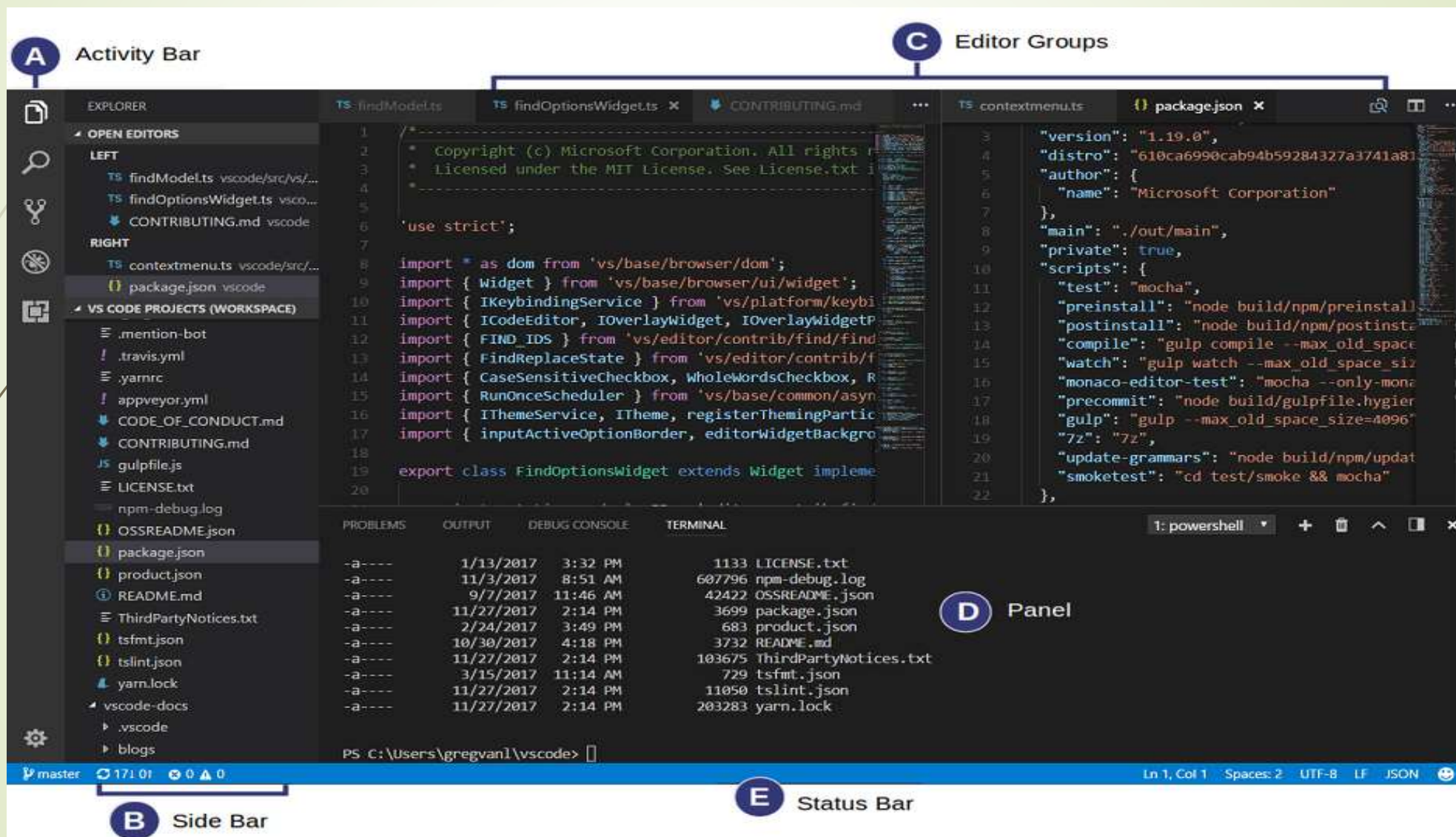
- **Visual Studio Code** is a source-**code editor** made by Microsoft for Windows, Linux and macOS.
- Features include support for debugging, syntax highlighting, intelligent **code** completion, snippets, **code** refactoring, and embedded Git.

Core feature :-

- Edit, build, and debug with ease
- Make it your own
- Robust and extensible architecture

June 2021 (version 1.58)

User Interface



Keyboard Shortcuts

5

1. General
2. Basic editing
3. Navigation
4. Search and replace
5. Multi cursor and selection
6. Rich lang editing
7. Editor management
8. File management
9. Display
10. Debug
11. Integrated terminal

Abraham Lincoln

“Give me six hours to chop down a tree and I will spend the first four sharpening the axe.”

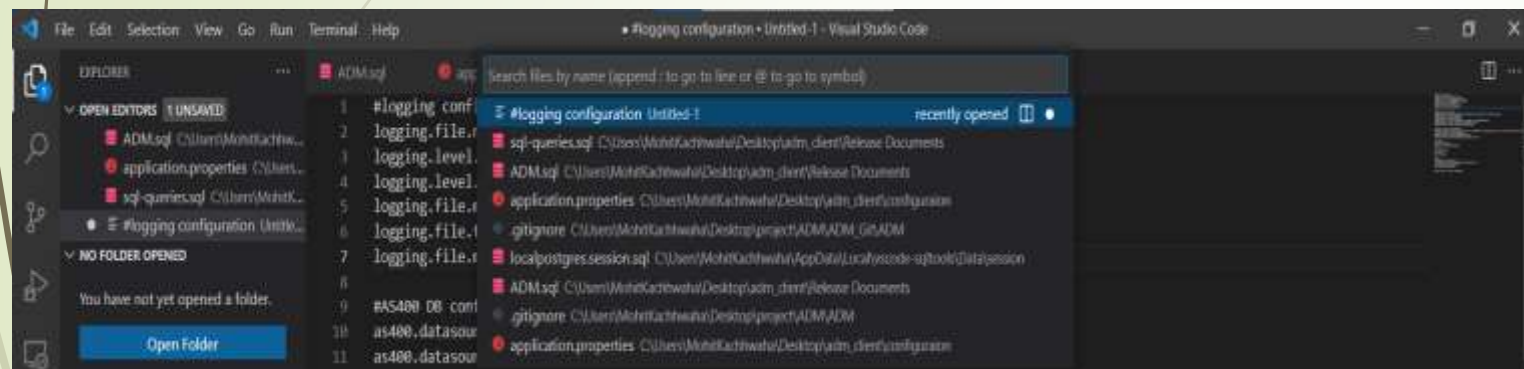
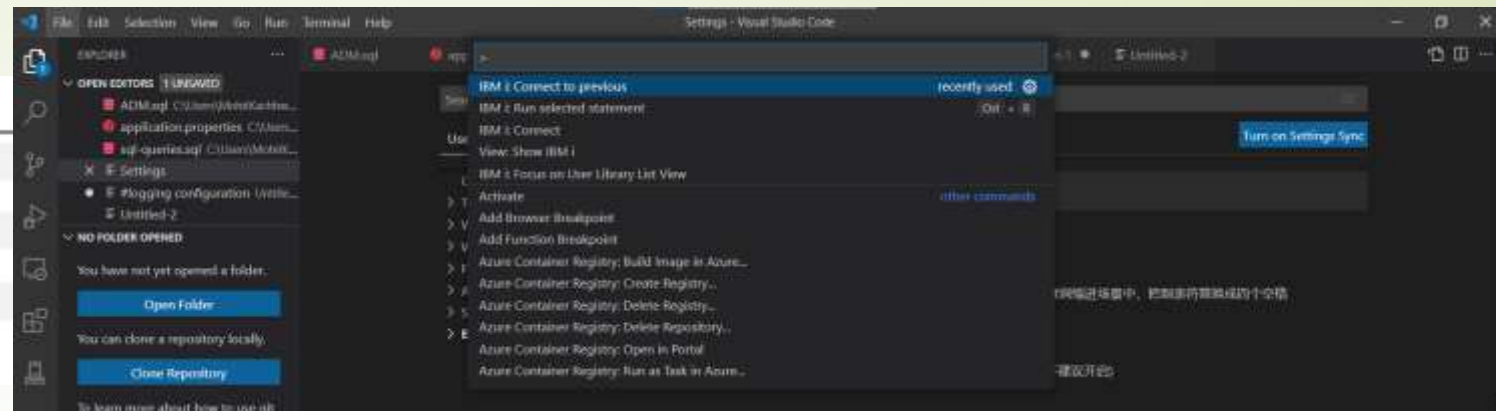
Resource:- <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf>



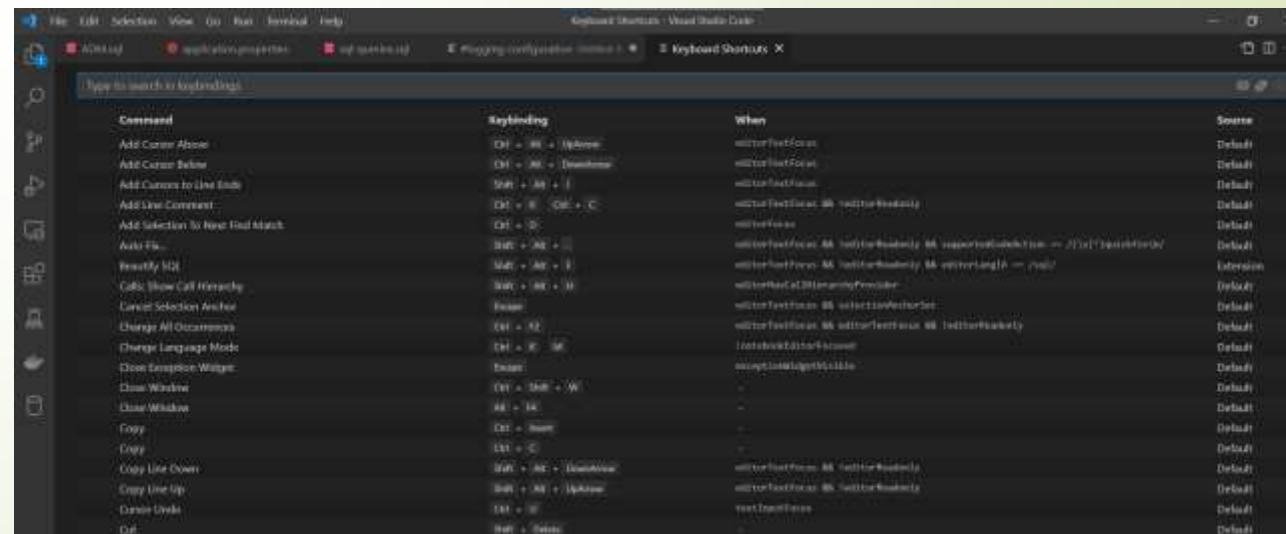
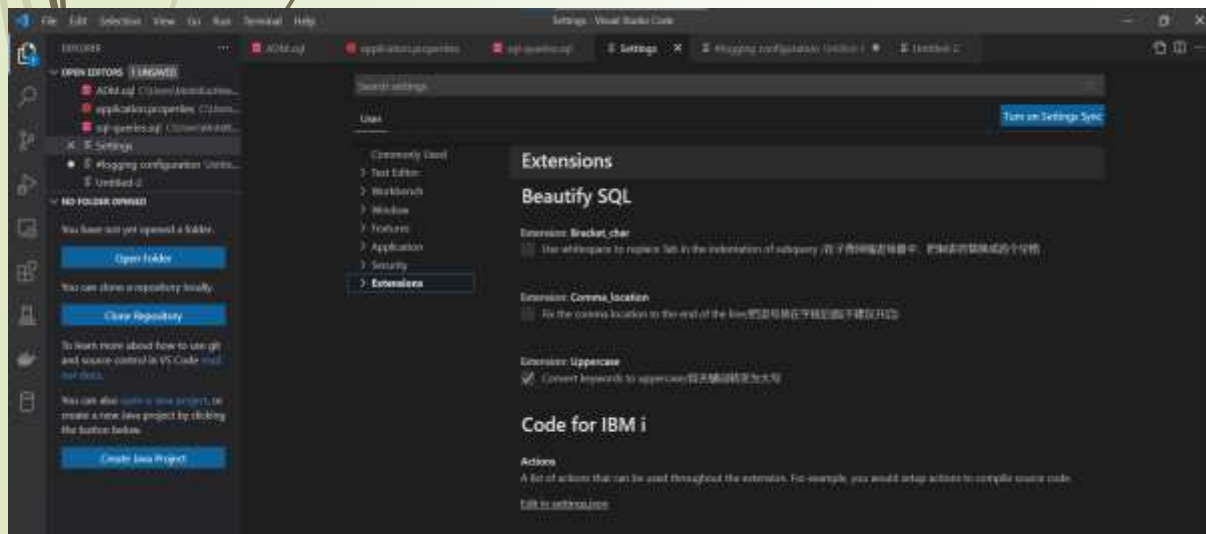
6

General

Ctrl+Shift+P, F1	Show Command Palette
Ctrl+P	Quick Open, Go to File...
Ctrl+Shift+N	New window/instance
Ctrl+Shift+W	Close window/instance
Ctrl+,	User Settings
Ctrl+K Ctrl+S	Keyboard Shortcuts



Connect (Ctrl+K) was pressed. Waiting for second key of chord...



Basic editing

Ctrl+X	Cut line (empty selection)
Ctrl+C	Copy line (empty selection)
Alt+ ↑ / ↓	Move line up/down ←
Shift+Alt+ ↑ / ↓	Copy line up/down ←
Ctrl+Shift+K	Delete line
Ctrl+Enter	Insert line below
Ctrl+Shift+Enter	Insert line above
Ctrl+Shift+\	Jump to matching bracket ←
Ctrl+] / [Indent/outdent line
Home / End	Go to beginning/end of line ←
Ctrl+Home	Go to beginning of file ←
Ctrl+End	Go to end of file ←
Ctrl+↑ / ↓	Scroll line up/down
Alt+PgUp / PgDn	Scroll page up/down ←
Ctrl+Shift+[Fold (collapse) region ←
Ctrl+Shift+]	Unfold (uncollapse) region
Ctrl+K Ctrl+[Fold (collapse) all subregions
Ctrl+K Ctrl+]	Unfold (uncollapse) all subregions
Ctrl+K Ctrl+0	Fold (collapse) all regions
Ctrl+K Ctrl+J	Unfold (uncollapse) all regions
Ctrl+K Ctrl+C	Add line comment
Ctrl+K Ctrl+U	Remove line comment
Ctrl+/	Toggle line comment ←
Shift+Alt+A	Toggle block comment ←
Alt+Z	Toggle word wrap

Navigation

Ctrl+T	Show all Symbols
Ctrl+G	Go to Line... ←
Ctrl+P	Go to File... ←
Ctrl+Shift+O	Go to Symbol... ←
Ctrl+Shift+M	Show Problems panel
F8	Go to next error or warning
Shift+F8	Go to previous error or warning
Ctrl+Shift+Tab	Navigate editor group history
Alt+ ← / →	Go back / forward ←

Search and replace

Ctrl+F	Find ←
Ctrl+H	Replace
F3 / Shift+F3	Find next/previous ←
Alt+Enter	Select all occurrences of Find match ←
Ctrl+D	Add selection to next Find match
Ctrl+K Ctrl+D	Move last selection to next Find match
Alt+C / R / W	Toggle case-sensitive / regex / whole word

Multi-cursor and selection

Alt+Click	Insert cursor ←
Ctrl+Alt+ ↑ / ↓	Insert cursor above / below ←
Ctrl+U	Undo last cursor operation ←
Shift+Alt+I	Insert cursor at end of each line selected
Ctrl+L	Select current line
Ctrl+Shift+L	Select all occurrences of current selection
Ctrl+F2	Select all occurrences of current word
Shift+Alt+→	Expand selection
Shift+Alt+←	Shrink selection
Shift+Alt+ +	Column (box) selection ←
(drag mouse)	
Ctrl+Shift+Alt+ + (arrow key)	Column (box) selection
Ctrl+Shift+Alt+PgUp/PgDn	Column (box) selection page up/down

Rich languages editing

Ctrl+Space	Trigger suggestion	←
Ctrl+Shift+Space	Trigger parameter hints	←
Shift+Alt+F	Format document	
Ctrl+K Ctrl+F	Format selection	←
F12	Go to Definition	←
Alt+F12	Peek Definition	
Ctrl+K F12	Open Definition to the side	
Ctrl+.	Quick Fix	←
Shift+F12	Show References	
F2	Rename Symbol	←
Ctrl+K Ctrl+X	Trim trailing whitespace	
Ctrl+K M	Change file language	

Editor management

Ctrl+F4, Ctrl+W	Close editor	←
Ctrl+K F	Close folder	
Ctrl+\	Split editor	←
Ctrl+ 1 / 2 / 3	Focus into 1 st , 2 nd or 3 rd editor group	←
Ctrl+K Ctrl+ ← / →	Focus into previous/next editor group	
Ctrl+Shift+PgUp / PgDn	Move editor left/right	
Ctrl+K ← / →	Move active editor group	

File management

Ctrl+N	New File	
Ctrl+O	Open File...	
Ctrl+S	Save	
Ctrl+Shift+S	Save As...	
Ctrl+K S	Save All	
Ctrl+F4	Close	
Ctrl+K Ctrl+W	Close All	
Ctrl+Shift+T	Reopen closed editor	←
Ctrl+K Enter	Keep preview mode editor open	
Ctrl+Tab	Open next	←
Ctrl+Shift+Tab	Open previous	←
Ctrl+K P	Copy path of active file	←
Ctrl+K R	Reveal active file in Explorer	←
Ctrl+K O	Show active file in new window/instance	

Display

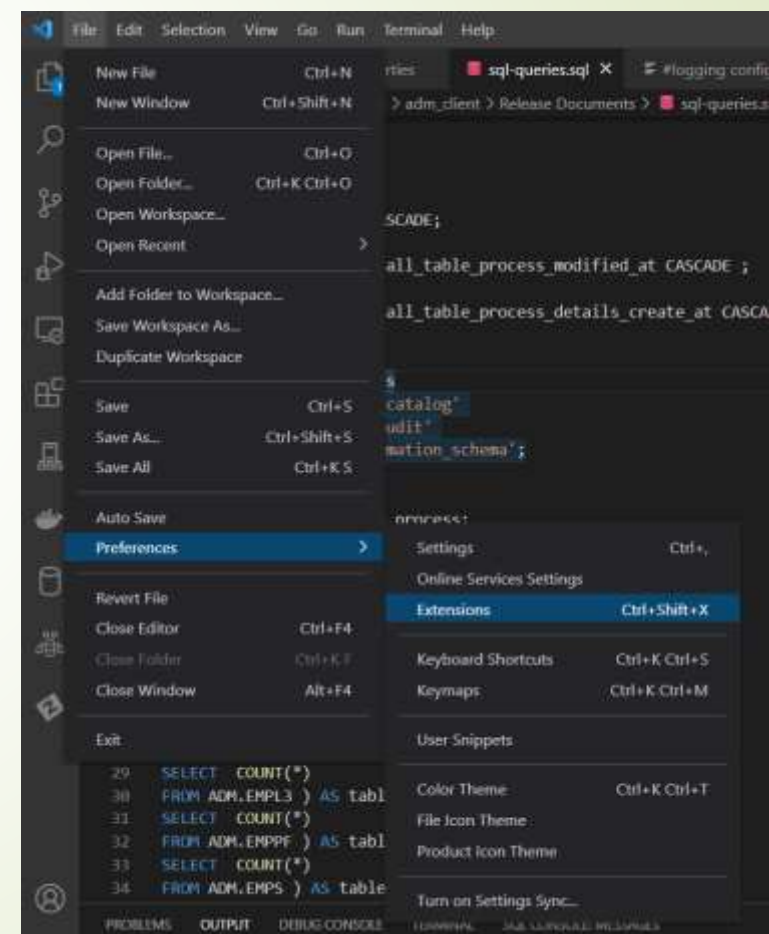
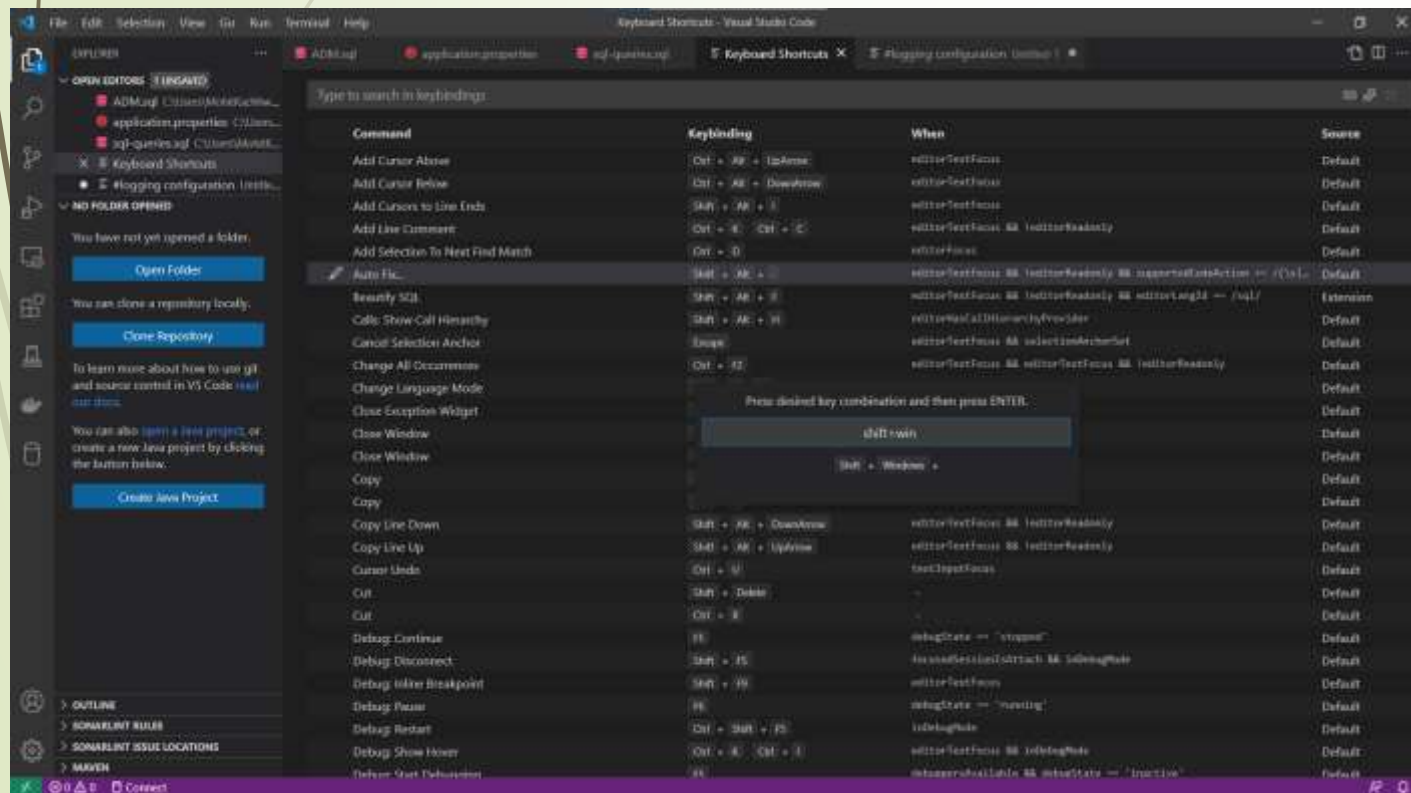
F11	Toggle full screen	←
Shift+Alt+0	Toggle editor layout (horizontal/vertical)	
Ctrl+ = / -	Zoom in/out	←
Ctrl+B	Toggle Sidebar visibility	
Ctrl+Shift+E	Show Explorer / Toggle focus	
Ctrl+Shift+F	Show Search	
Ctrl+Shift+G	Show Source Control	
Ctrl+Shift+D	Show Debug	
Ctrl+Shift+X	Show Extensions	
Ctrl+Shift+H	Replace in files	
Ctrl+Shift+J	Toggle Search details	
Ctrl+Shift+U	Show Output panel	←
Ctrl+Shift+V	Open Markdown preview	
Ctrl+K V	Open Markdown preview to the side	
Ctrl+K Z	Zen Mode (Esc Esc to exit)	

Debug

F9	Toggle breakpoint
F5	Start/Continue
Shift+F5	Stop
F11 / Shift+F11	Step into/out
F10	Step over
Ctrl+K Ctrl+I	Show hover

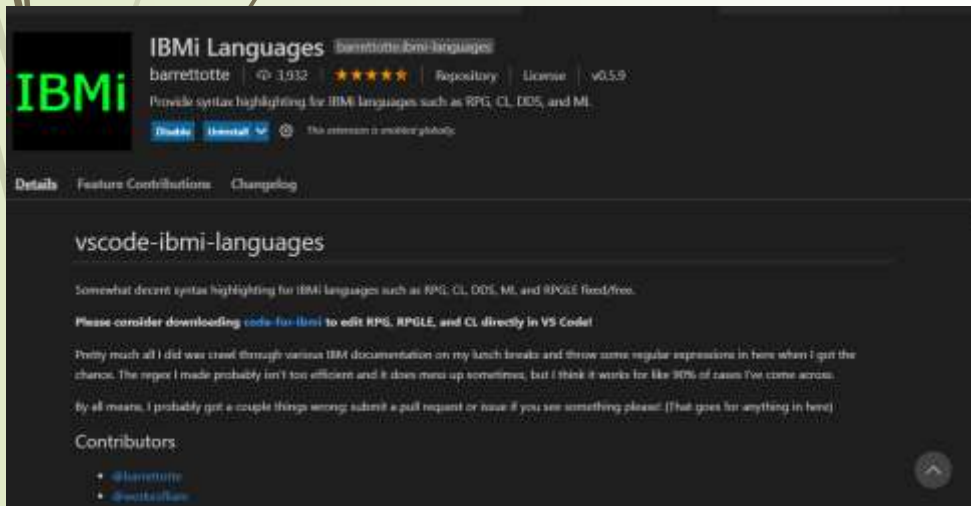
Integrated terminal

Ctrl+`	Show integrated terminal
Ctrl+Shift+`	Create new terminal
Ctrl+C	Copy selection
Ctrl+V	Paste into active terminal
Ctrl+↑ / ↓	Scroll up/down
Shift+PgUp / PgDn	Scroll page up/down
Ctrl+Home / End	Scroll to top/bottom



IBM i Languages by barrettotte

- extension for the majority of native languages like RPG, CL, DDS, and RPGLE fixed/free. This extension provides syntax highlighting, keyword prompting, etc.



```

test.rpgle
1  **FREE
2  Ctl-Opt DftActGrp(*No) Option(*Srcstmt : *NodebugIO);
3  //=====
4  // Demo Program to show syntax highlighting
5  //=====
6  Dcl-F InputFile;
7  Dcl-F DisplayFile  WORKSTN;
8  //-----
9  // Variable declaration
10 //-----
11 Dcl-S CharField      Char(30);
12 Dcl-S IntegerField   Int(5);
13 //=====
14 // Program data structure
15 //-----
16 Dcl-DS *N  PSDS;
17     #Prog      Char(10)      Pos(1);
18     #JobName    Char(10)      Pos(244);
19     #User       Char(10)      Pos(254);
20     #JobNo      Zoned(6:0)    Pos(264);
21 End-DS;
22 //=====
23 // Main processing
24 //-----
25 Exsr Initialization;
26 Exsr MainProcessing;
27 Exsr Exit;
28 //-----
  
```

This is for RPGLE Free.

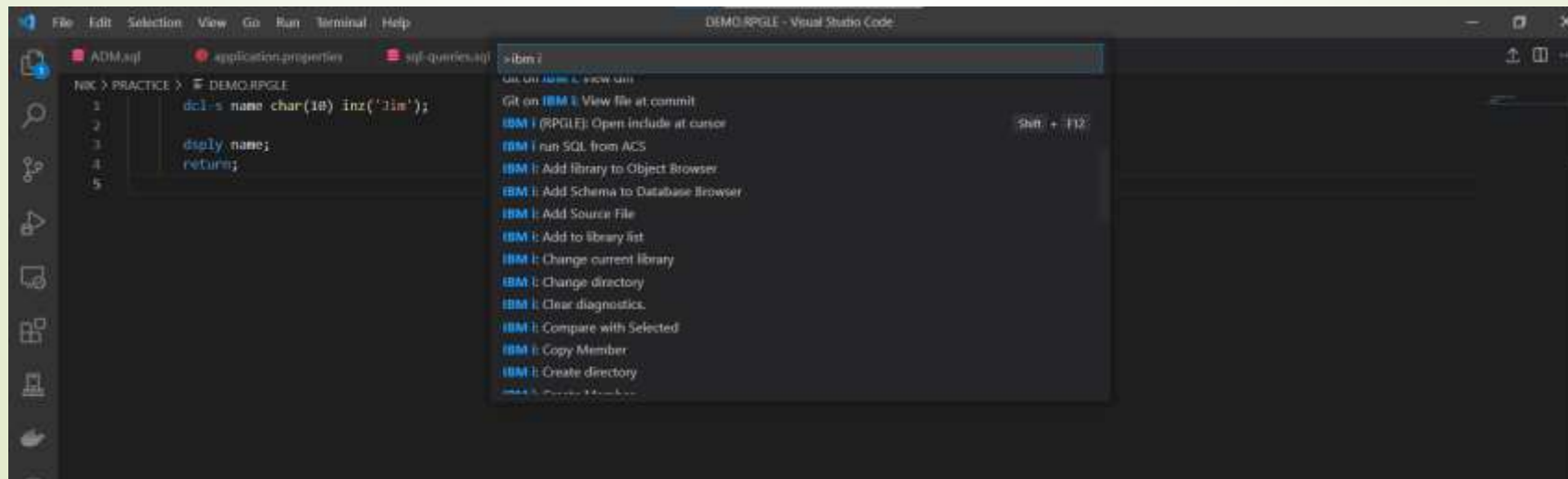
This is for CLLE.

And one for DDS.

```
SSH FS - ssh-pub400 > CLLE > democl.cle
1  PGM          PARM(&CUSTNO)
2
3  DCL          VAR(&CUSTNO) TYPE(*DEC) LEN(10)
4  DCL          VAR(&CUSTNM) TYPE(*CHAR) LEN(20) VALUE(' ')
5  DCL          VAR(&ADDR1) TYPE(*CHAR) LEN(20)
6  DCL          VAR(&ADDR2) TYPE(*CHAR) LEN(20)
7  DCL          VAR(&PHONE) TYPE(*DEC) LEN(10)
8
9  OVRDBF       FILE(CUSTMAST) TOFILE(APKHEKALE/CUSTMAST)
10 CALL         PGM(GETCUST) PARM(&CUSTNO &CUSTNM &ADDR1 &ADDR2 &PHONE)
11
12 ENDPGM
13
```

```
SSH FS - ssh-pub400 > CLLE > custmast.pf
1  A                                     UNIQUE
2  A                                     R CUSTMASTR
3  *
4  A          CUSTNO          10S 0      TEXT('Customer Number')
5  A                                     COLHDG('Customer' 'Number')
6  A          CUSTNM          20A        TEXT('Customer Name')
7  A                                     COLHDG('Customer' 'Name')
8  A          CUADDR1         20A        TEXT('Address Line 1')
9  A                                     COLHDG('Address' 'Line 1')
10 A          CUADDR2         20A        TEXT('Address Line 2')
11 A                                     COLHDG('Address' 'Line 2')
12 A          CUPHONE         10S 0      TEXT('Phone Number')
13 A                                     COLHDG('Phone' 'Number')
14 A          K CUSTNO
```

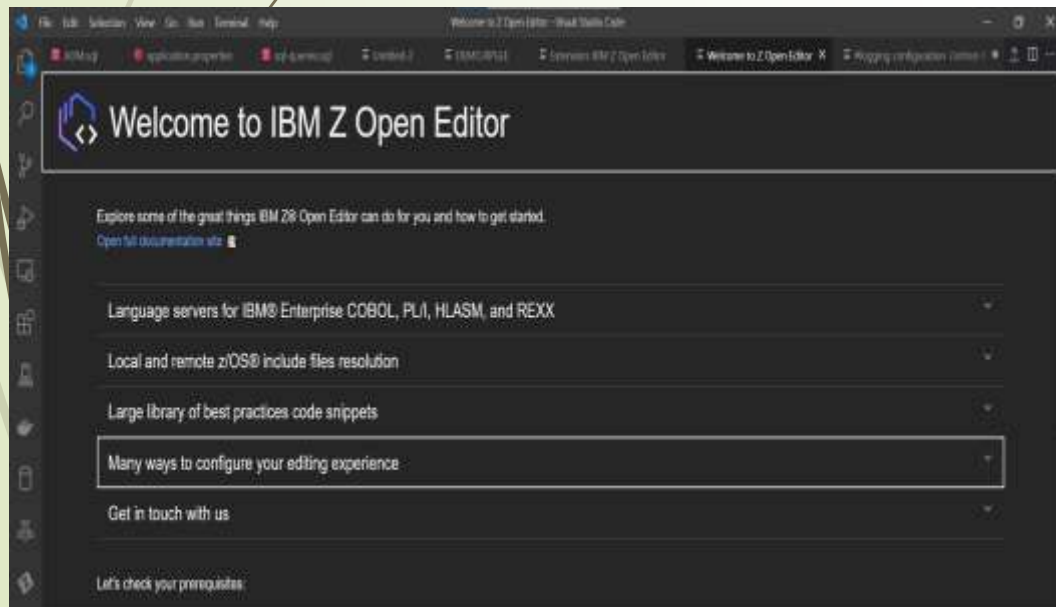
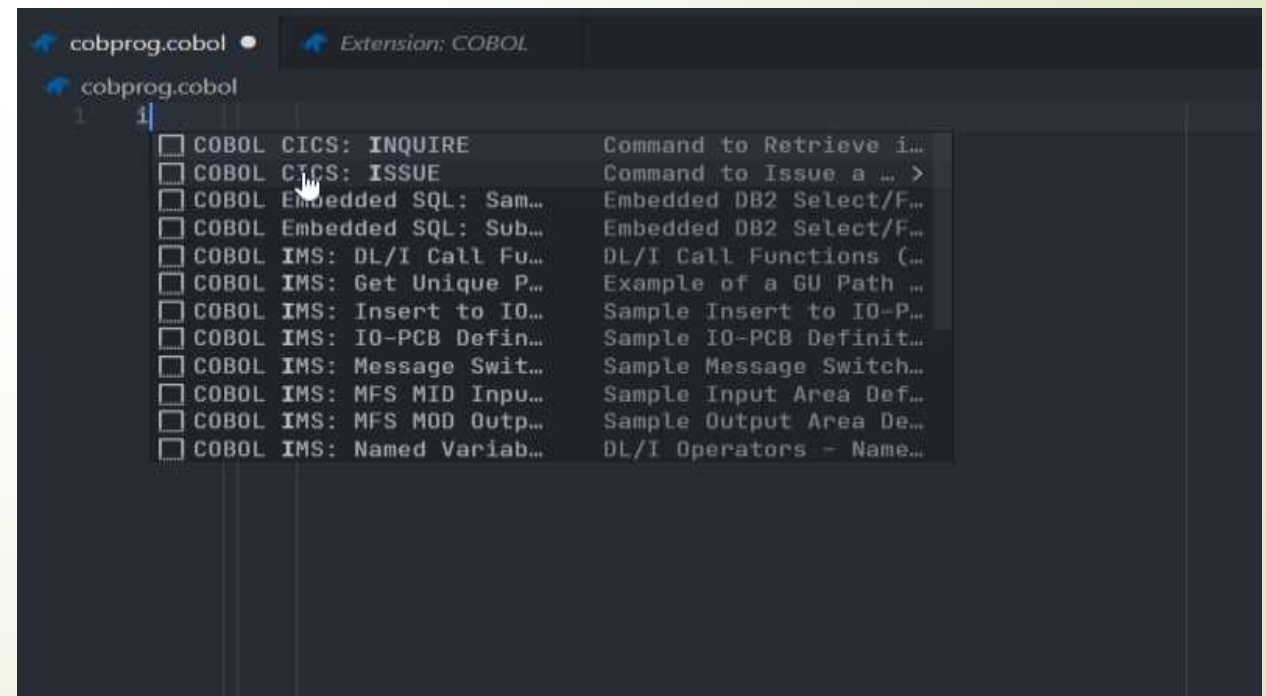
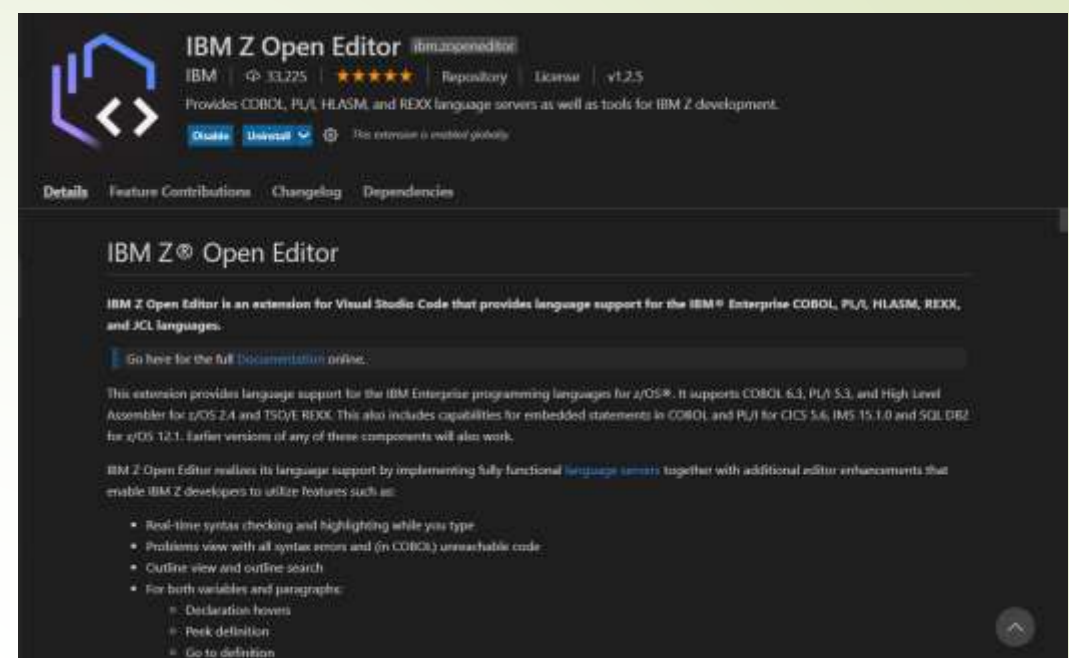
IBM I : Commands



12

2) IBM Z Open Editor

This COBOL extension is so powerful, with few keystrokes, most of your program gets generated. Let's see, when I start typing IDENT and hit Enter, the entire IDENTIFICATION DIVISION gets generated using the snippets provided by the extension.



Other Extensions

INSTALLED 24

- Code for IBM i** 0.6.1 ← 1K ★ 5
Maintain your RPGLE, CL, COBOL, C/CPP on IBM i right from Visual Studio Code.
Halcyon Tech Ltd
- Db2 Connect** 2.2.0 ← 4K ★ 4
VS Code Extension for IBM Data Server
IBM
- Debugger for Java** 0.34.0 9.3M ★ 4
A lightweight Java debugger for Visual Studio Code
Microsoft
- Docker** 1.14.0 9.6M ★ 4.5
Makes it easy to create, manage, and debug containerized applications.
Microsoft
- Git Client for IBM i** 0.0.4 ← 140
A git client that works on a remote IBM i
Halcyon Tech Ltd
- IBM i SQL** 0.0.2 ← 257 ★ 5
IBM i run SQL from ACS
Niels Lissberg
- IBM Z Open Editor** 1.2.5 ★ 33K ★ 5
Provides COBOL, PL/I, HLASM, and REXX language servers as well as tools for IBM Z ...
IBM
- IBMi Languages** 0.5.9 ★ 3K ★ 4.5
Provide syntax highlighting for IBMi languages such as RPG, CL, DDS, and MI.
barrettotte

EXTENSIONS: MARKETPLACE

ibm

- IBM Blockchain Platform** 2.0.3 71K ★ 4.5
End to end extension for Hyperledger Fabric developers. ...
IBM Blockchain [Install](#)
- IBM Z Open Editor** 1.2.5 33K ★ 5
Provides COBOL, PL/I, HLASM, and REXX language server...
IBM
- IBM Cloud Account** 1.0.6 34K
The IBM Cloud account extension for Visual Studio Code ...
IBM [Install](#)
- IBM Streams** 1.1.2 1K ★ 5
IBM Streams Support for Visual Studio Code
IBM [Install](#)
- Code for IBM i** 0.6.1 1K ★ 5
Maintain your RPGLE, CL, COBOL, C/CPP on IBM i right fr...
Halcyon Tech Ltd
- ibm-jcl** 0.11.0 2K ★ 5
IBM Job Control Language
Dan Kelosky [Install](#)
- IBM Color Scheme [DARK]** 1.5.0 5K ★ 4.5
A dark color theme based on the IBM design resources c...
PaperFanz [Install](#)
- ibm-assembler** 0.8.1 2K ★ 5
IBM Assembler Language Highlighter
Dan Kelosky [Install](#)
- IBM Developer Tools** 0.0.14 10K ★ 5
Extension for VS Code editor to enable IBM Bluemix deve...
IBM [Install](#)
- IBM Theme** 1.6.0 576 ★ 5
A dark color theme with a unique aesthetic and attention...
Marvin Engelmann [Install](#)
- IBM Carbon Color Theme** 1.1.0 1K ★ 5
Dark color theme based on IBM's Carbon Design System ...
eXotech-code [Install](#)
- IBM API Connect Tools** 0.0.5 4K

Extensions/ Themes for Open- source Languages

14

Extension	Description
Bracket Pair Colorizer	A customizable extension for colorizing matching brackets - Link .
Code Spell Checker	Spelling checker for source code - Link .
Indenticator	Highlights your current indent depth - Link .
One Dark Pro Theme	Atom's iconic One Dark theme for Visual Studio Code - Link .
Python	Linting, Debugging (multi-threaded, remote), Intellisense, Jupyter Notebooks, code formatting - Link .
SSH FS	File system provider using SSH. - Link .
Visual Studio IntelliCode	AI-assisted development. - Link .
vscode-icons	Icons for Visual Studio Code. - Link .

How to install Extensions & some other Tips



- Click on the extension symbol as displayed by the image, type the desired extension in the search bar, when the extension appears, click on the install button, and you are done. More information [here](#).

How to write code directly in IFS on IBM i?

- It becomes convenient if we don't have to upload the code often to the IFS directory on IBM i manually. One extension always comes in handy that is SSH-FS, which makes the IFS folder available to us in the VS Code.

SSH-FS needs the below details, along with the name of the connection.

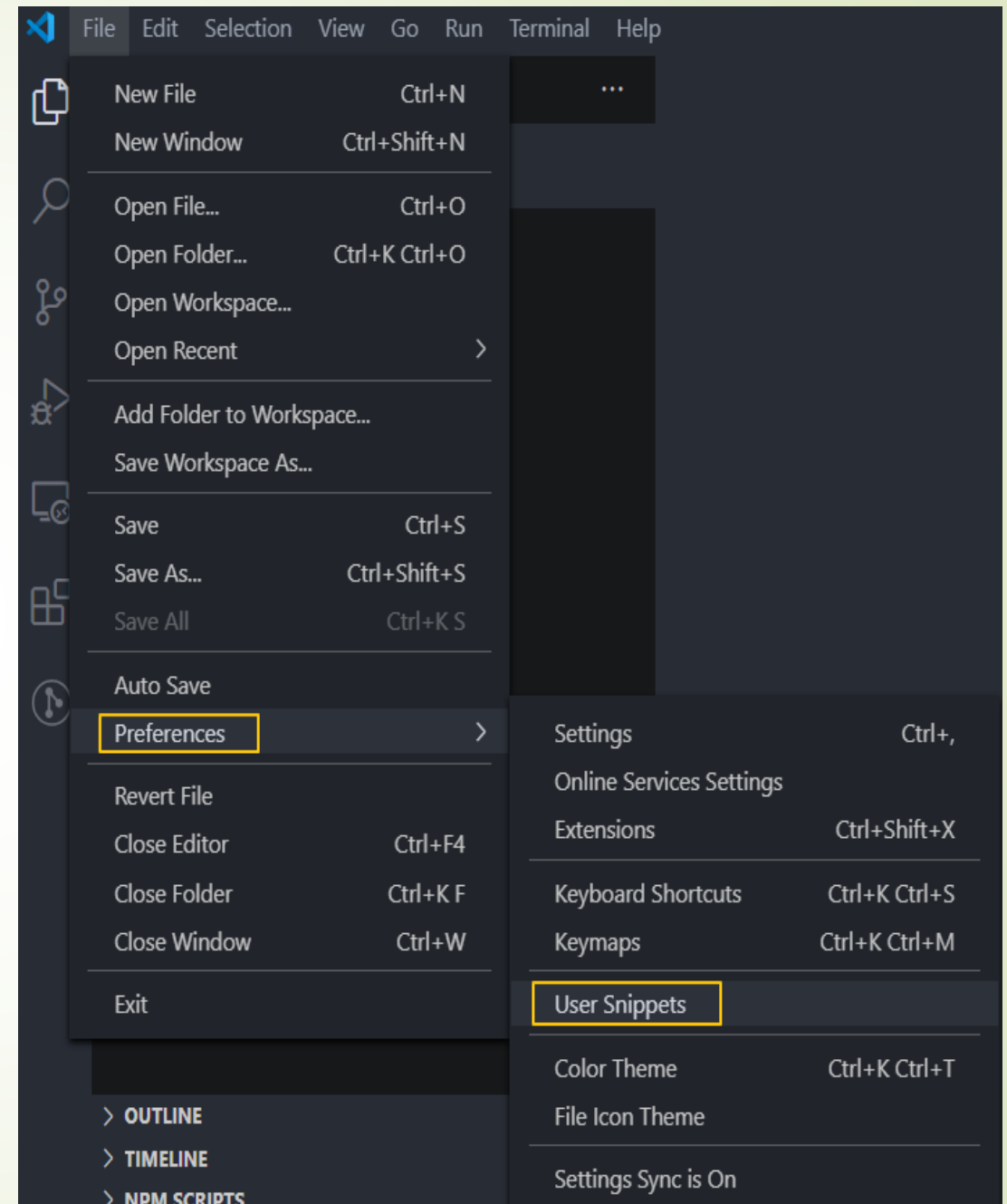
- 1.The IP/Domain name of your IBM i system
- 2.SSH Port (generally port 22, PUB400 has 2222 port configured for SSH)
- 3.User Id
- 4.Password
- 5.Root folder (/home/USERID)

Use of Snippet

What are Snippets?

- VS Code has this feature called Snippets, where we can assign a shortcut to a block of code that we write quite often. We can create our snippet for the language we want to use. We must create a file in JSON format in
- File → Preferences → User Snippets.

```
// Example:
{
  "ControlOption": {
    "prefix": "CTLOPT",
    "body": "ctl-Opt DftActGrp(*No) Option(*%rcstnt : *NodebugID);$0",
    "description": "Add ctl-Opt at the begining of the code"
  },
  "DefineSpec": {
    "prefix": "DF",
    "body": "Del-F $1;$0",
    "description": "Define File in F Specs"
  },
  "DefineWorkstation": {
    "prefix": "DW",
    "body": "Del-F $1 \\WORKSTN;$0",
    "description": "Define Display file"
  },
  "DefineOSpecCharacter": {
    "prefix": "DC",
    "body": "Del-S $1\\t\\tChar($2);$0",
    "description": "Define Character field in O Specs"
  },
  "DefineOSpecVarCharacter": {
    "prefix": "DV",
    "body": "Del-S $1\\t\\tVarchar($2);$0",
    "description": "Define Variable Character field in O Specs"
  },
  "DefineOSpecInteger": {
    "prefix": "DI",
    "body": "Del-S $1\\t\\tInt($2);$0",
    "description": "Define Integer field in O Specs"
  }
}
```



Thank You