

Visual Studio Code MultiValue Extension



MULTIVALUE

[Licence](#) Copyright (c) 2019 MVExtensions Group

MIT License Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

All other trademarks and service marks are property of their respective holders.

Table of Contents

- [Visual Studio Code MultiValue Extension](#)
- [Table of Contents](#)
- [Purpose of this guide](#)
- [1. Introduction](#)
- [2. Prerequisites](#)
- [3. Installing VSCode](#)
- [4. Installing MVBasic Extension](#)
- [5. Connecting to a MultiValue Server](#)
 - [5.1 Install the MVGateway](#)
 - [5.2 Configure the MVGateway](#)
 - [5.3 Testing the connection](#)
 - [5.4 Troubleshooting the Connection](#)
- [6. MV Developer Features](#)
 - [6.1 Syntax Highlighting](#)
 - [6.2 Intellisense](#)
 - [6.3 Find All References](#)
 - [6.4 Goto/Peek Definition.](#)
 - [6.5 Internal Subroutine lookup](#)
 - [6.6 Compiling and Cataloging your programs.](#)
 - [6.7 Formatting Programs](#)
- [7. Sample Settings Files](#)
 - [7.1 Universe](#)
 - [7.2 Unidata](#)
 - [7.3 OpenQM](#)
 - [7.4 jBASE](#)
 - [7.5 D3](#)
 - [7.6 mvBase](#)
 - [7.7 MVON](#)
 - [7.8 Associating Programs with the MVextension](#)
 - [7.9 Additional MVBasic Developer Settings](#)

Purpose of this guide

This document describes how to install and use the MultiValue Basic Visual Studio Code extension in a MultiValue Development Environment. For purposes of this guide:

- MV refers to Pick-style MultiValue application and database environments
- VSCode refers to Microsoft Visual Studio Code
- MVBasic refers to the MV VSCode Extension

[\(top\)](#)

1. Introduction

VSCode is a free, open source, feature rich, IDE that allows programmers to develop and debug code in various languages. There is also a community of developers that have developed extensions that provide functionality for VSCode that isn't built into the main program.

The MVBasic extension provides developers to gain the features of VSCode with MV BASIC programs. This extension provides connectivity to your MultiValue database, reading and writing code and is currently available for jBASE, OpenQM, MVON#, D3, Universe and Unidata. If the source code files are stored in O/S directories that are accessible by the user's system, then it can access other variants of MultiValue database.

Details on how to connect to the different databases are explained in later chapters.

This extension includes the following features:

1. Code highlighting for MV Basic Programs
2. Intellisense for the MV Basic Statements and Functions
3. Code folding
4. Code formatting
5. Goto/Peek Definition. Automatically jump to and peek internal subroutines
6. Goto/Peek Definition. Automatically peek/load CALL, CHAIN and INCLUDE routines
7. Syntax checking for GOTO/GOSUB, LOOPS, CASE STATEMENTS and IF THE/ELSE statements
8. Access your remote MV files and programs
9. Find all References of a word in current program

VSCode is available on Windows, Linux and Mac OSX.

[\(top\)](#)

2. Prerequisites

The following environment is required in order to use the extension.

1. Windows, Linux or Mac OSX machine.
2. VSCode
3. MVGateway installed (Optional, Windows only for remote editing)

[\(top\)](#)

3. Installing VSCode

VSCode can be downloaded from the following link: <https://code.visualstudio.com/download>

Select the version for your operating system. If you are running Windows, run either the 32 bit or the 64 bit installer. To check if you are 32 or 64-bit go to control panel/system or in Windows 10 search settings for 32.

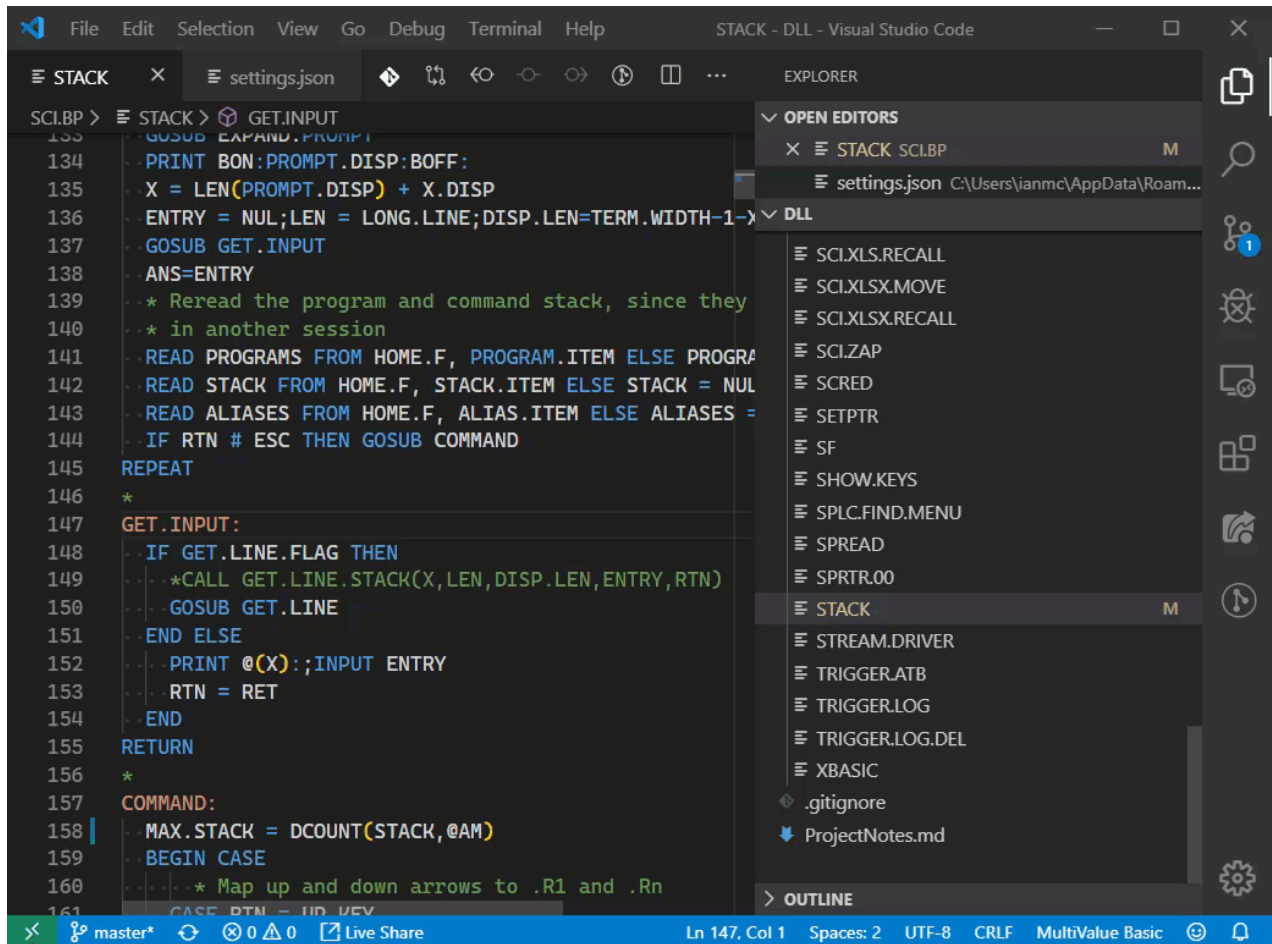
[\(top\)](#)

4. Installing MVBasic Extension

Before we can start using the MV features for VSCode, we need to install the MVBasic extension. VSCode has an automated download and installation process for extensions.

Start VSCode and select the Extensions Button. In the search box, type "mvbasic".

Select the MV Developer Extension by clicking on the item in the list then choose install. Once the extension is installed we are ready to start using VSCode on your MV source code.



(top)

5. Connecting to a MultiValue Server

The extension allows us to connect to MV servers and edit, compile and catalog BASIC programs.

5.1 Install the MVGateway

In order for VSCode to communicate to your MV server install the MVGateway provided with the extension. The MVGateway runs as a Windows service and is installed using the Windows setup program provided. This setup program is included in %userprofile%\vscode\extension. There will be a directory specific to the MVBasic extension. The name of this directory will change slightly with each new release of the extension because it contains the version number of the extension. In that directory is a MVGateway directory with the Windows MSI executable (setup program). Double-click on the MSI file and install the MVGateway service. The gateway does not need any special configuration or setup. The gateway can be installed on any Windows system in your network that has access to your computer and your MV server. Many users install it on their local system if they are running a Windows system.

5.2 Configure the MVGateway

Once you have installed the MVGateway the next step is to configure a VSCode Workspace. The Workspace will contain all the parameters required to connect and login to the remote MV Server. It is recommended that you use a directory on your system where you will save the Workspace definitions. If you have multiple servers and/or

multiple accounts on each server, you will create a Workspace that points to a each particular server and account. In order to do that we first need to configure a VSCode **Workspace**.

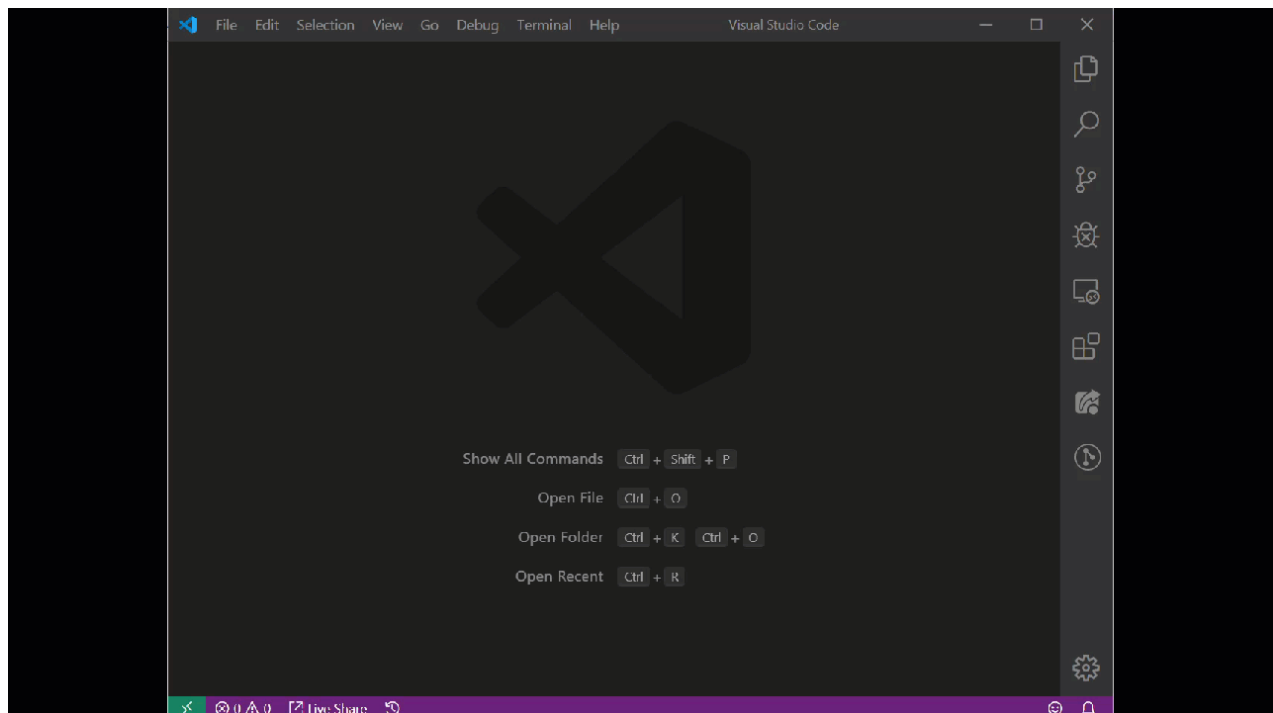
To connect to an MV Server, the following information is required:

1. Hostname or IP Address of the MV server.
2. User name to login into the server
3. Password for the user above
4. Account name to connect to on the MV Server

To create a new Workspace:

1. Select "**Save Workspace As**" from **File** Menu. In this example, a folder called VSCode is used to store the Workspace definitions. As shown in the above screenshot, a blank Workspace called Demo is created that can now be configured to point to your MV server.
2. To configure the connection parameters open the Workspace settings. One way to do that is to use the menu and select **File, Preference, Setting**.
3. This will bring up the Settings pane in VSCode. Once this is open you **MUST** select the **Workspace Tab**. This is one area that VSCode has changed so your screen may appear a little different than these screenshots. For many users it is easier to edit the JSON version of the configuration rather than searching for each individual parameter. This is done by clicking on the Open Settings (JSON) button near the top right corner of your VSCode. You can then paste in the appropriate settings for the platform from section 7.

- [7.1 Universe](#)
- [7.2 Unidata](#)
- [7.3 OpenQM](#)
- [7.4 jBASE](#)
- [7.5 D3](#)
- [7.6 mvBase](#)
- [7.7 MVON](#)



After adding all the parameters to the workspace, your settings should be like this:

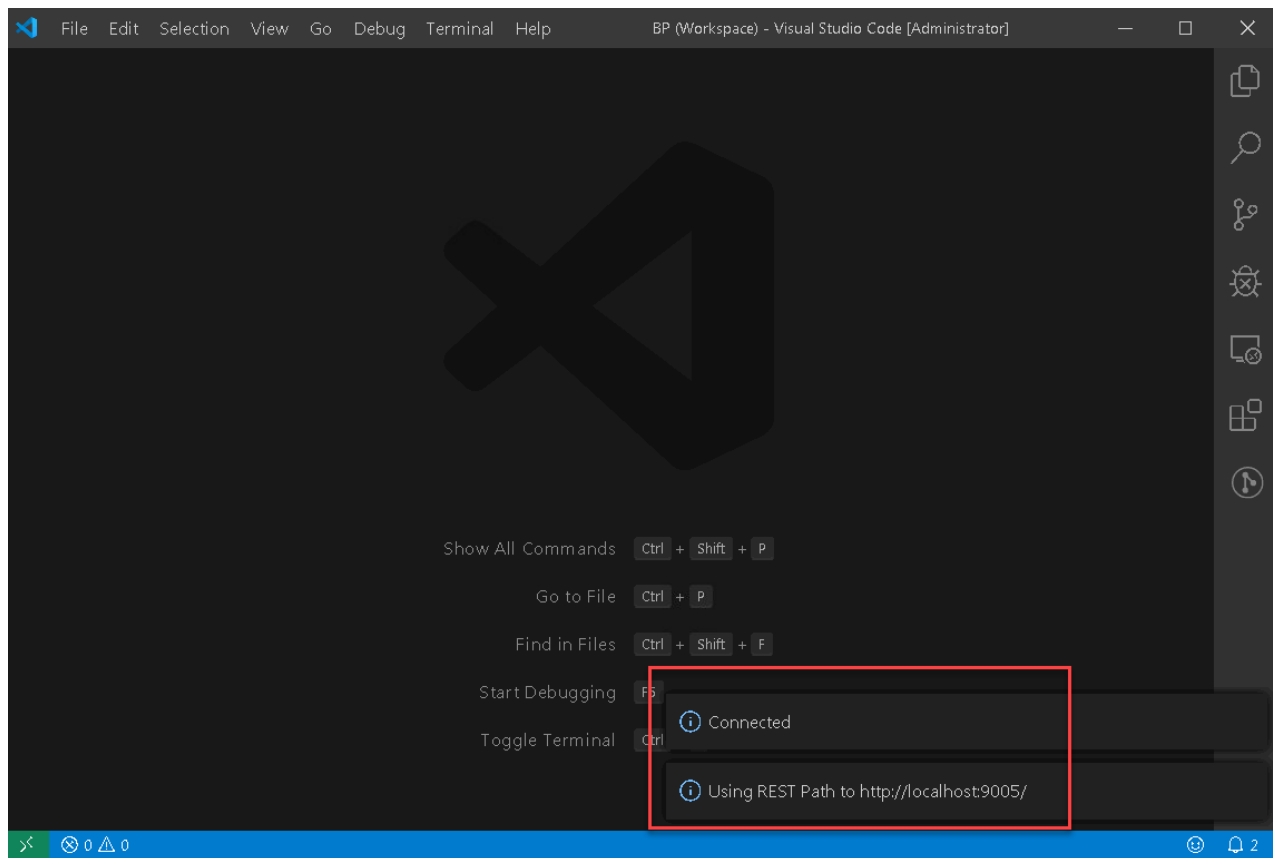
```
{
  "folders":[
    {
      "uri": "RestFS:/",
      "name": "Account - DEMO",
    }
  ],
  "settings": {
    "mvbasic.RestPath": "http://localhost:9005/",
    "mvbasic.GatewayType": "Universe",
    "mvbasic.UseGateway": true,
    "mvbasic.RemoteHost": "192.168.1.2",
    "mvbasic.UserName": "myUserName",
    "mvbasic.Password": "mvPassword",
    "mvbasic.Account": "DEMO",
    "files.associations": {
      "*.md": "markdown",
      "*.js": "javascript",
      "\\*" : "mvbasic"
    }
  }
}
```

These are the base settings required to connect to an Universe MV Server. Press Ctl-S to save your settings.

5.3 Testing the connection

We can test to if the connection to MV server works by Pressing **F1** or Cmd/Ctrl-Shift-P. VSCode will prompt you for the command to run. Type **Connect** to display all commands with Connect in it and choose "Connect to Multivalue REST FS".

The extension will connect to the server and retrieve a list of Directory files from the server. If the connection is successful, the following messages will appear at the bottom left of the screen.



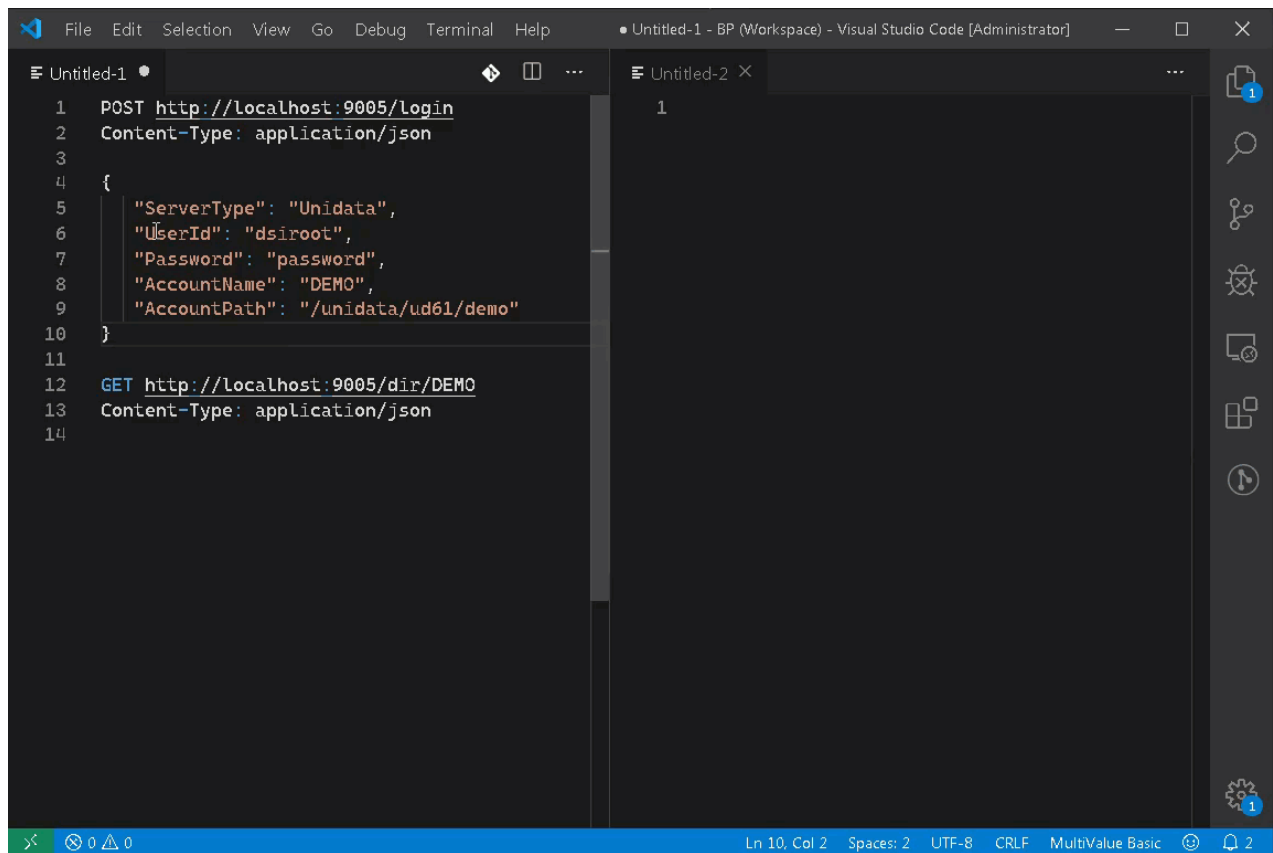
5.4 Troubleshooting the Connection

See [section 7](#) for sample configuration files for each MV platform.

To turn on logging in the MV Gateway, add the following to the settings section of the config file and the MVGateway service should log to c:\temp.

```
"MVBasic.gatewayDebug": true,  
"MVBasic.trace.server": "verbose"
```

To test the RestFS connection manually, install a [REST client](#) and review some request/response pairs to make sure the MVGateway service is connecting.



Sample REST requests, these will obviously have to be changed with specific values:

```
POST http://localhost:9005/login
Content-Type: application/json

{
  "ServerType": "Unidata",
  "UserId": "dsiroot",
  "Password": "password",
  "AccountName": "DEMO",
  "AccountPath": "/unidata/ud61/demo"
}
```

```
GET http://localhost:9005/dir/DEMO
Content-Type: application/json
```

[\(top\)](#)

6. MV Developer Features

The following is a list of features that the extensions offer MV Developers when using VSCode.

6.1 Syntax Highlighting

Code is highlighted based on the current theme selected for VSCode.

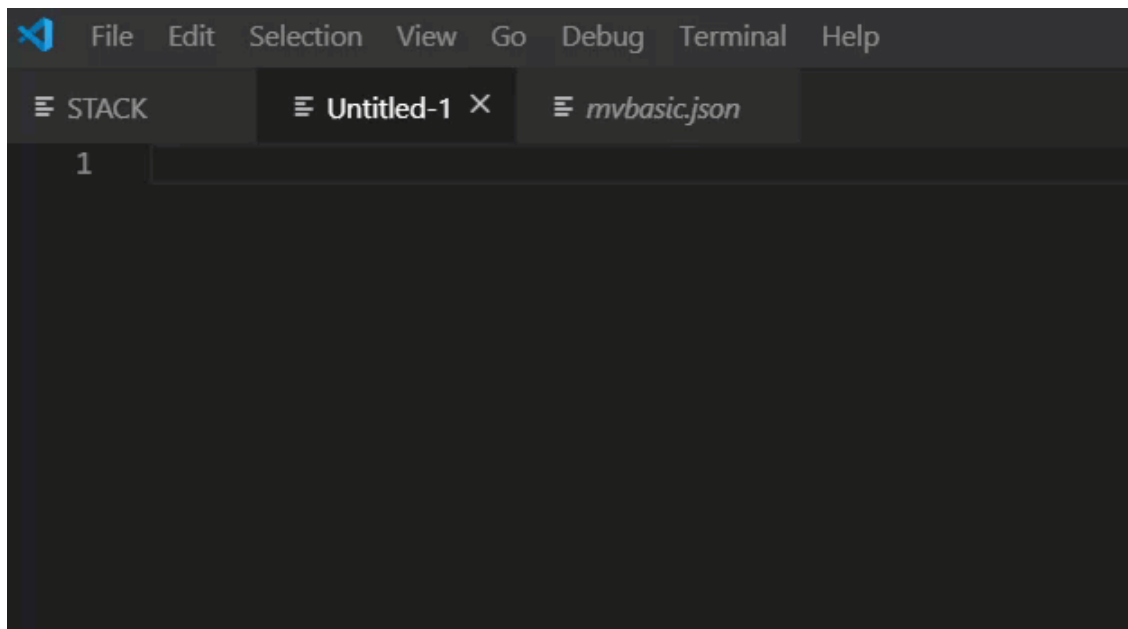

```

133  ..GOSUB EXPAND.PROMPT
134  ..PRINT BON:PROMPT.DISP:BOFF:
135  ..X = LEN(PROMPT.DISP) + X.DISP
136  ..ENTRY = NUL;LEN = LONG.LINE;DISP.LEN=TERM.WIDTH-1-X
137  ..GOSUB GET.INPUT
138  ..ANS=ENTRY
139  ..* Reread the program and command stack, since they may be modified
140  ..* in another session
141  ..READ PROGRAMS FROM HOME.F, PROGRAM.ITEM ELSE PROGRAMS = NUL
142  ..READ STACK FROM HOME.F, STACK.ITEM ELSE STACK = NUL
143  ..READ ALIASES FROM HOME.F, ALIAS.ITEM ELSE ALIASES = NUL
144  ..IF RTN # ESC THEN GOSUB COMMAND
145  REPEAT
146  *
147  GET.INPUT:
148  ..IF GET.LINE.FLAG THEN
149  ...*CALL GET.LINE.STACK(X,LEN,DISP.LEN,ENTRY,RTN)
150  ...GOSUB GET.LINE
151  ..END ELSE
152  ..PRINT @(X);;INPUT ENTRY
153  ..RTN = RET
154  ..END
155  RETURN
156  *
157  COMMAND:
158  ..MAX.STACK = DCOUNT(STACK,@AM)
159  ..BEGIN CASE
160  ...* Map up and down arrows to .R1 and .Rn
161  ..CASE RTN = UP_KEY

```

6.2 Intellisense

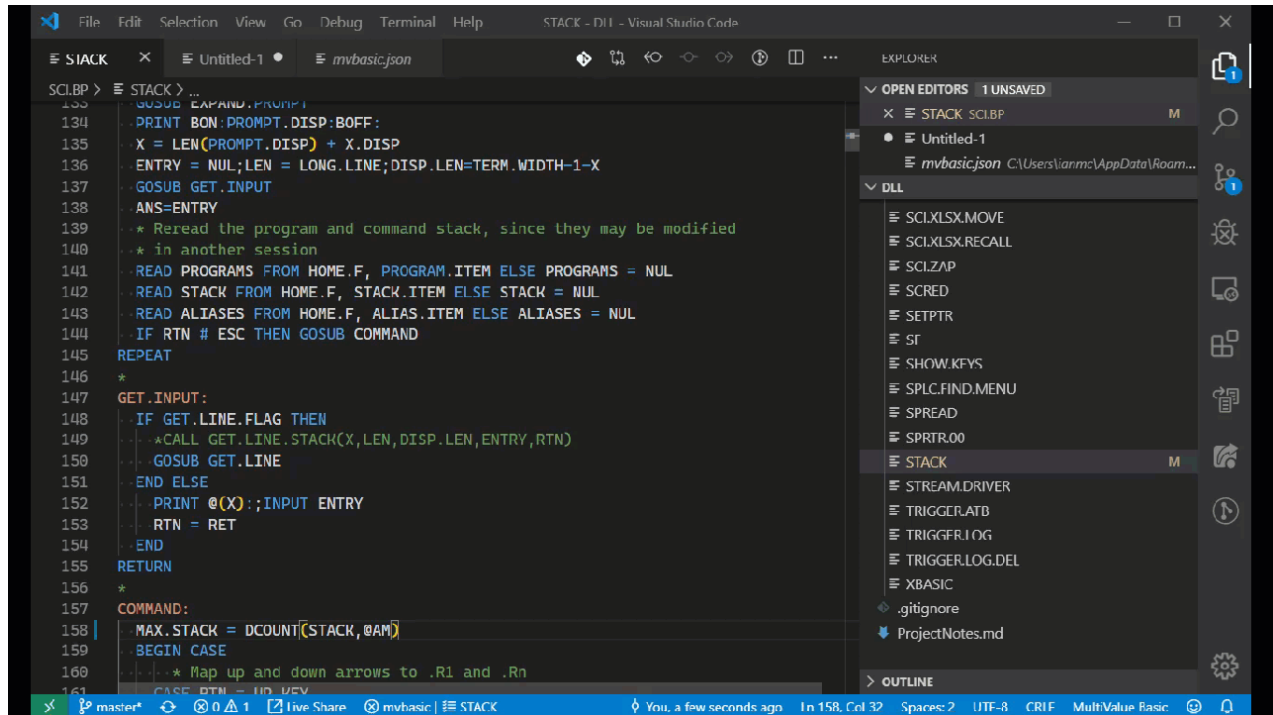
As you type your program, you will be prompted with available statements and functions including the syntax and description.



6.3 Find All References

You can find all references to a word in your program by **right clicking** on a word and selecting **Find All References** from the menu.

The display consists of 2 panels, the right containing the line that the word is in and the actual code block is in the left. Clicking on a line in the right panel will take you to the code block.



6.4 Goto/Peek Definition.

If you **right click** on a internal or external subroutine name and select **Peek Definition**, a window appears showing the internal or external subroutine.

If you select **Goto Definition**, the cursor is moved to start of the subroutine.

```

133  GOSUB EXPAND.PROMPT
134  PRINT BON:PROMPT.DISP:BOFF:
135  X = LEN(PROMPT.DISP) + X.DISP
136  ENTRY = NUL;LEN = LONG.LINE;DISP.LEN=TERM.WIDTH-1-X
137  GOSUB GET.INPUT
138  ANS=ENTRY
139  * Reread the program and command stack, since they may be modified
140  * in another session
141  READ PROGRAMS FROM HOME.F, PROGRAM.ITEM ELSE PROGRAMS = NUL
142  READ STACK FROM HOME.F, STACK.ITEM ELSE STACK = NUL
143  READ ALIASES FROM HOME.F, ALIAS.ITEM ELSE ALIASES = NUL
144  IF RTN # ESC THEN GOSUB COMMAND
145  REPEAT
146  *
147  GET.INPUT:
148  IF GET.LINE.FLAG THEN
149  *CALL GET.LINE.STACK(X,LEN,DISP.LEN,ENTRY,RTN)
150  GOSUB GET.LINE
151  END ELSE
152  PRINT @(X);;INPUT ENTRY
153  RTN = RET
154  END
155  RETURN
156  *
157  COMMAND:
158  MAX.STACK = DCOUNT(STACK,@AM)
159  BEGIN CASE
160  * Map up and down arrows to .R1 and .Rn
161  CASE RTN = UP_KEY

```

6.5 Internal Subroutine lookup

Pressing "**Ctrl-space**" after the word GOTO, GOSUB or GO TO, will allow you to select from defined internal subroutines in your program.

6.6 Compiling and Cataloging your programs.

Right Clicking inside the code window allows you to select 3 options:

1. Catalog Basic Program – catalogs the BASIC program
2. Compile Basic Program – compiles the basic program.
3. Compile Basic Program with Debug – compiles with the debug flag set.

After the option is selected, the results will be displayed in message box at the bottom of the screen. If an error is detected, the editor will place the cursor on the line where error occurs.

6.7 Formatting Programs

Right Clicking and selecting **Format Document**, will format your BASIC program. The formatting is based on the 2 settings, **mvbasic.indent** and **mvbasic.margin** that have default values of 3 and 5.

(top)

7. Sample Settings Files

7.1 Universe

```
{
  "folders":[
    {
      "uri": "RestFS:/",
      "name": "Account - Universe",
    }
  ],
  "settings": {
    "mvbasic.RestPath": "http://localhost:9005/",
    "mvbasic.UseGateway": true,
    "mvbasic.GatewayType": "Universe",
    "mvbasic.RemoteHost": "192.168.1.2",
    "mvbasic.UserName": " myUserName",
    "mvbasic.Password": "myPassword",
    "mvbasic.Account": "DEMO",
    "files.associations": {
      "\\*":"mvbasic"
    }
  }
}
```

Setting		Description
mvbasic.RestPath	http://localhost:9005/	Path to REST Gateway
mvbasic.UseGateway	true	Indicate that the gateway must be used.
mvbasic.RemoteHost	192.168.1.2	The servers IP/Host name that is running the Universe Database.
mvbasic.GatewayType	Universe	Connecting to a Universe server
mvbasic.UserName	myUserName	The Windows/UNIX user id to log into the server.
mvbasic.Password	myPassword	The password for the user above.
mvbasic.Account	DEMO	The account name on Universe to connect to. This must be defined in the UV.ACCOUNT file in the UV account.

7.2 Unidata

```
{
  "folders":[
    {
      "uri": "RestFS:/",
      "name": "Account - Unidata",
    }
  ],
  "settings": {
    "mvbasic.RestPath": "http://localhost:9005/",
    "mvbasic.UseGateway": true,
    "mvbasic.GatewayType": "Unidata",
    "mvbasic.RemoteHost": "192.168.1.2",
    "mvbasic.UserName": "myUserName",
  }
```

```

    "mvbasic.Password": "MyPassword",
    "mvbasic.Account": "DEMO",
    "mvbasic.AccountPath": "/usr/data/DEMO",
    "files.associations": {"*": "mvbasic"}
  }
}

```

Setting		Description
mvbasic.RestPath	http://localhost:9005	Path to REST Gateway
mvbasic.UseGateway	true	Indicate that the gateway must be used.
mvbasic.RemoteHost	192.168.1.10	The servers IP/Host name that is running the Unidata Database.
mvbasic.GatewayType	Unidata	Connecting to a Unidata server
mvbasic.UserName	MyUserName	The Windows/UNIX user id to log into the server.
mvbasic.Password	MyPassword	The password for the user above.
mvbasic.Account	DEMO	A name for this account.
mvbasic.AccountPath	/usr/data/DEMO	The path on the Unidata machine to the Unidata account.

7.3 OpenQM

```

{
  "folders": [
    {
      "uri": "RestFS:/",
      "name": "Account - QM",
    }
  ],
  "settings": {
    "mvbasic.RestPath": "http://localhost:9005/",
    "mvbasic.UseGateway": true,
    "mvbasic.GatewayType": "QM",
    "mvbasic.RemoteHost": "192.168.1.2",
    "mvbasic.UserName": "MyUserName",
    "mvbasic.Password": "MyPassword",
    "mvbasic.Account": "DEMO",
    "files.associations": {"*": "mvbasic"}
  }
}

```

Setting		Description
mvbasic.RestPath	http://localhost:9005/	Path to REST Gateway
mvbasic.UseGateway	true	Indicate that the gateway must be used.
mvbasic.remoteHost	192.168.1.2	The servers IP/Host name that is running the OpenQM Database.
mvbasic.gatewayType	QM	Connecting to a OpenQM server

Setting		Description
mvbasic.UserName	MyUserName	The Windows/UNIX user id to log into the server.
mvbasic.Password	MyPassword	The password for the user above.
mvbasic.Account	DEMO	The account name on the QM server to connect to. This must be defined in the ACCOUNTS file in the QMSYS account.

7.4 jBASE

```
{
  "folders":[
    {
      "uri": "RestFS:/",
      "name": "Account - jBASE",
    }
  ],
  "settings": {
    "mvbasic.RestPath": "http://localhost:9005/",
    "mvbasic.UseGateway": true,
    "mvbasic.GatewayType": "jBASE",
    "mvbasic.RemoteHost": "192.168.1.2",
    "mvbasic.UserName": "MyUserName",
    "mvbasic.Password": "MyPassword",
    "mvbasic.Account": "",
    "files.associations": {"\\*":"mvbasic"}
  }
}
```

Setting		Description
mvbasic.RestPath	http://localhost:9005/	Path to REST Gateway
mvbasic.UseGateway	true	Indicates that the gateway must be used.
mvbasic.RemoteHost	192.168.137.2	The servers IP name that is running the jBASE Database.
mvbasic.GatewayType	jBASE	Connecting to a jBASE server
mvbasic.UserName	MyUserName	The Windows/UNIX user id to log into the server.
mvbasic.Password	MyPassword	The password for the user above.
mvbasic.Account		This is blank, jBASE uses the default path of the user for the account.

A record in the **MD** called **MVONFILES** can used as a list of available files, alternatively all files are displayed.

7.5 D3

```
{
  "folders":[
    {
```

```

        "uri": "RestFS:/",
        "name": "Account - D3",
    }
],
"settings": {
    "mvbasic.RestPath": "http://localhost:9005/",
    "mvbasic.UseGateway": true,
    "mvbasic.GatewayType": "D3",
    "mvbasic.RemoteHost": "192.168.1.2",
    "mvbasic.UserName": "dm",
    "mvbasic.AccountPassword": "",
    "mvbasic.Account": "dm",
    "files.associations": {"\\*": "mvbasic"}
}
}

```

Setting		Description
mvbasic.RestPath	http://localhost:9005/	Path to the REST Gateway
mvbasic.UseGateway	true	Indicates that the gateway must be used.
mvbasic.RemoteHost	192.168.137.102	The servers IP name that is running the D3 Database.
mvbasic.GatewayType	D3	Connecting to a D3 server
mvbasic.UserName	dm	The D3 User name to log in with
mvbasic.AccountPassword		Specify the account password if a password is set on the account.
Mvon.Account	dm	The D3 account to connect to.

MSVP must be configured for the above account and the user must have MSVP access. A record in the **MD** called **VSCodeFILES** can be used as a list of available files, alternatively all files are displayed.

7.6 mvBase

```

{
    "folders": [
        {
            "uri": "RestFS:/",
            "name": "Account - mvBase",
        }
    ],
    "settings": {
        "mvbasic.RestPath": "http://localhost:9005/",
        "mvbasic.UseGateway": true,
        "mvbasic.GatewayType": "mvBase",
        "mvbasic.RemoteHost": "192.168.1.2",
        "mvbasic.UserName": "MyUserName ",
        "mvbasic.AccountPassword": "MyPassword",
        "files.associations": {"\\*": "mvbasic"}
    }
}

```

Setting	Description	
mvbasic.RestPath	http://localhost:9005/	Path to the REST Gateway
mvbasic.UseGateway	true	Indicates that the gateway must be used.
mvbasic.RemoteHost	192.168.137.2	The servers IP name that is running mvBase.
mvbasic.GatewayType	mvBase	Connecting to a mvBase server
mvbasic.UserName	MyUserName	The User name to log in with
Mvon.AccountPassword	MyPassword	Specify the account password if a password is set on the account.

MSVP must be configured for the above account and the user must have MSVP access. A record in the **MD** called **VSCodeFILES** can be used as a list of available files, alternatively all files are displayed.

7.7 MVON#

```
{
  "folders":[
    {
      "uri": "RestFS:/",
      "name": "Account - MVON#",
    }
  ],
  "settings": {
    "mvbasic.RestPath": "http://192.168.1.2/mvonrest",
    "mvbasic.UseGateway": false,
    "mvbasic.UserName": "MyUserName ",
    "mvbasic.Password": "MyPassword",
    "mvbasic.Account": "Netbasic",
    "mvbasic.RemoteDebug": true,
    "files.associations": {"\\*":"mvbasic"}
  }
}
```

Setting	Description	
mvbasic.RestPath	http://192.168.1.2/mvonrest	URL of the MVON# REST service
mvbasic.UseGateway	false	Indicates that the gateway is not required and may be omitted from the configuration.
mvbasic.UserName	MyUserName	The User name to log in with
mvbasic.Password	MyPassword	Specify the account password if a password is set on the account.
mvbasic.Account	Netbasic	Name of the MVON# account you are connecting to.
mvbasic.RemoteDebug	True	This enables the MVON# remote debugging feature allowing a rich debugging environment in VSCode

MVON# connects differently from other MV servers. It does not require the MVGateway service, providing a direct connection through the MVON# REST server. You must have this server configured before connecting.

7.8 Associating Programs with the MVextension

Most programming languages have an extension that says what language it is. Python is .py, C# is .cs etc. MV Basic typically does not follow this concept.

In order to tell VSCode that we are editing a MV BASIC program in order to enable Syntax highlighting, Intellisense, Linting and other features, we need to tell VSCode that files in the Workspace are linked to this MV BASIC extension. This is achieved by adding the following setting to your Workspace settings (see red highlight below).

```
{
  "folders": [
    {
      "uri": "RestFS:/",
      "name": "Account - DEMO",
    }
  ],
  "settings": {
    "mvbasic.RestPath": "http://localhost/mvonrest",
    "files.associations": {"\\*": "mvbasic"}
  }
}
```

7.9 Additional MVBasic Developer Settings

The following settings are available to customize your VSCode MVBasic Developer experience.

Setting	Description
mvbasic.margin	The number of characters to use as a margin when formatting.
mvbasic.indent	The number of characters to use when indenting code blocks.
mvbasic.useCamelCase	Use CamelCase for Intellisense keywords.
mvbasic.ignoreGotoScope	The linter will not highlight goto that jump into the middle of loops.
mvbasic.formattingEnabled	Set to false to disable code formatting.