Visual Studio Code

MultiValue Extension

MULTIVALUE

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.

Contents

PREFACE

Purpose of this guide

This document describes how to use the MultiValue Basic Visual Studio Code extension in a MultiValue Development Environment. For purposes of this guide MV refers to Pick-style application and database environments mostly known currently as MultiValue and VSCODE refers to Visual Studio Code.

1 INTRODUCTION

Visual Studio Code 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 MultiValue Basic extension provides developers to gain the features of Visual Studio Code 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's, 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

Visual Studio Code is available on Windows, Linux and Mac OSX.

2 PREREQUISITES

The following environment is required in order to use Visual Studio Code.

1. Windows, Linux or Mac OSX machine.

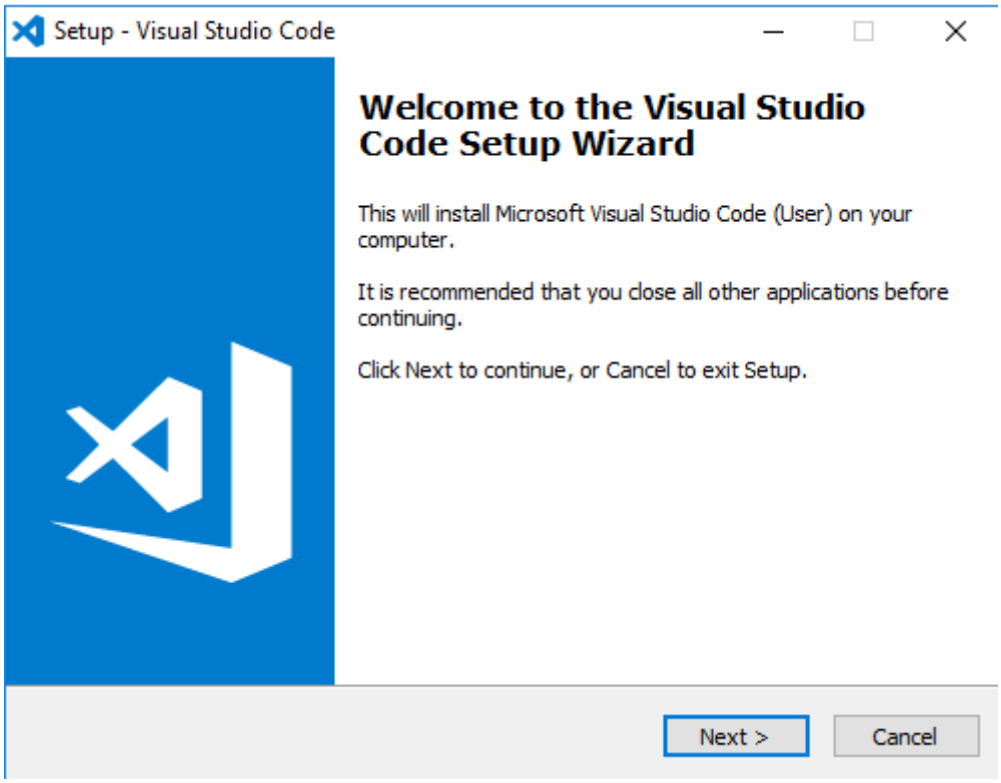
3 INSTALLING VISUAL STUDIO CODE

Visual Sudio Code can be downloaded from the following link:

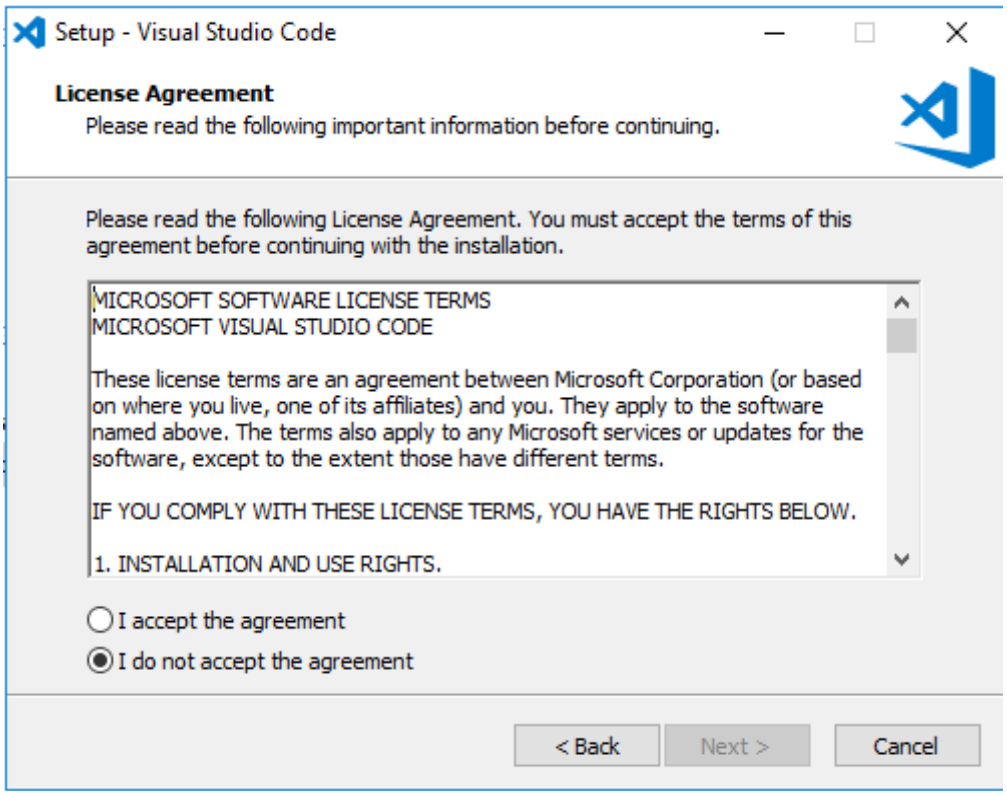
https://code.visualstudio.com/Download

You can select the version for your operating system. This guide describes how to install the Windows version of Visual Studio Code.

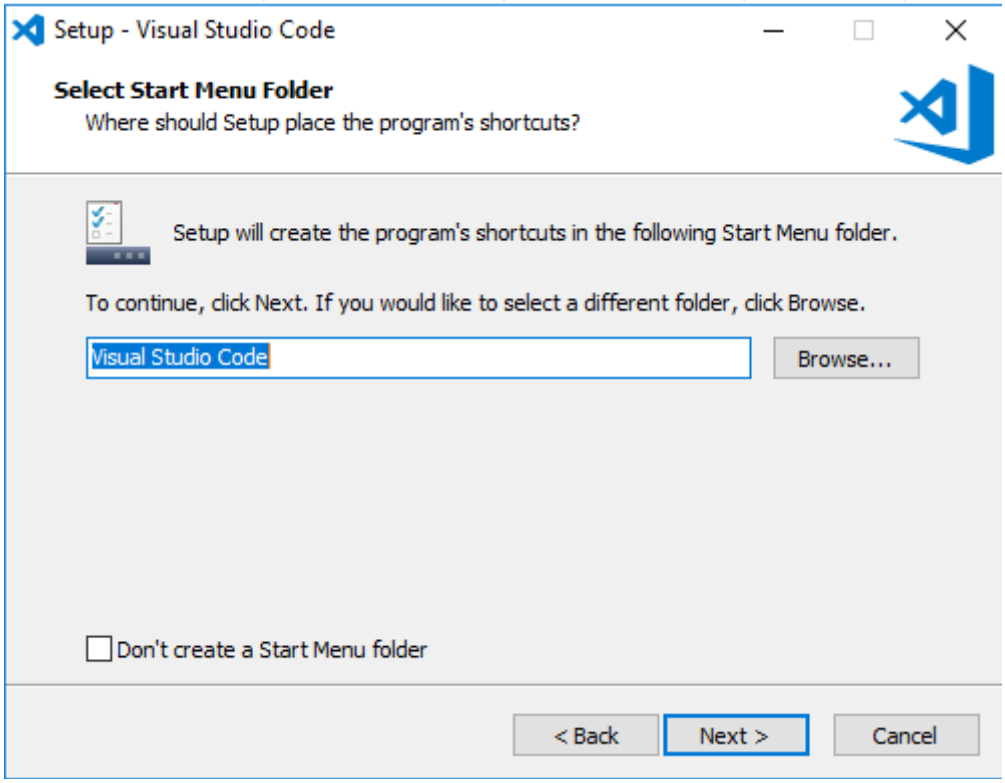
Depending on your Window operating system, run either the 32 bit **VSCodeUserSetup-ia32-1.26.1.exe** or the 64 bit **VSCodeUserSetup-x64-1.26.1.exe**



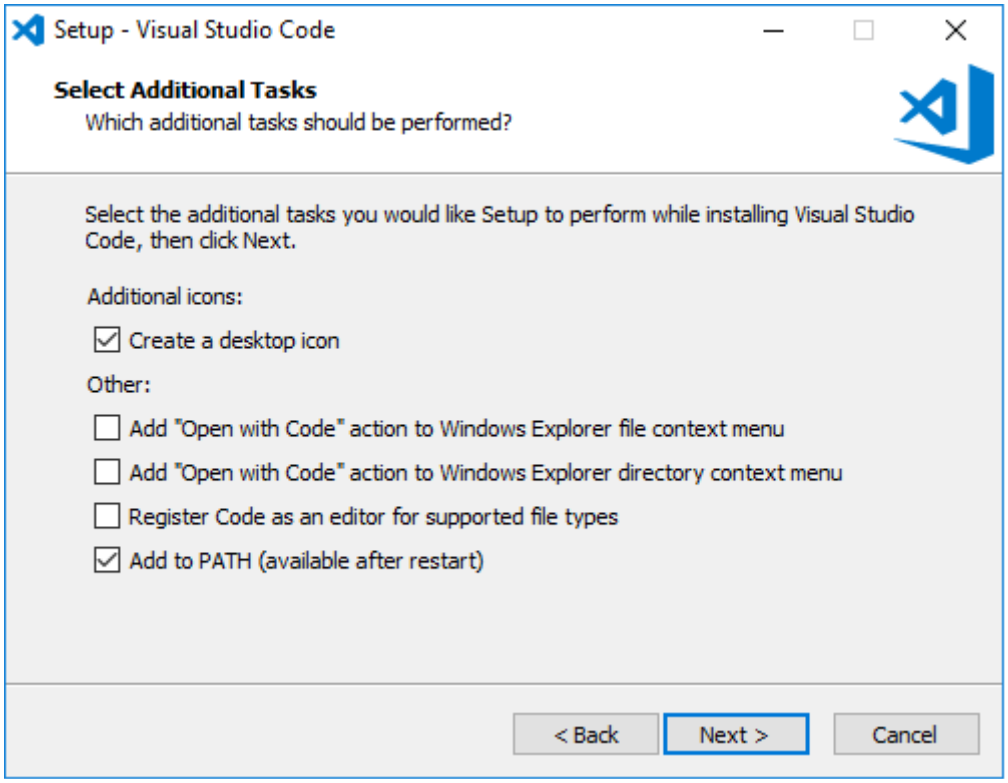
Select Next



Accept the License Agreement and select **Next**



Accept the defaults or specify your folder and select **Next**



Select the options you would like to include in the install and select **Next**

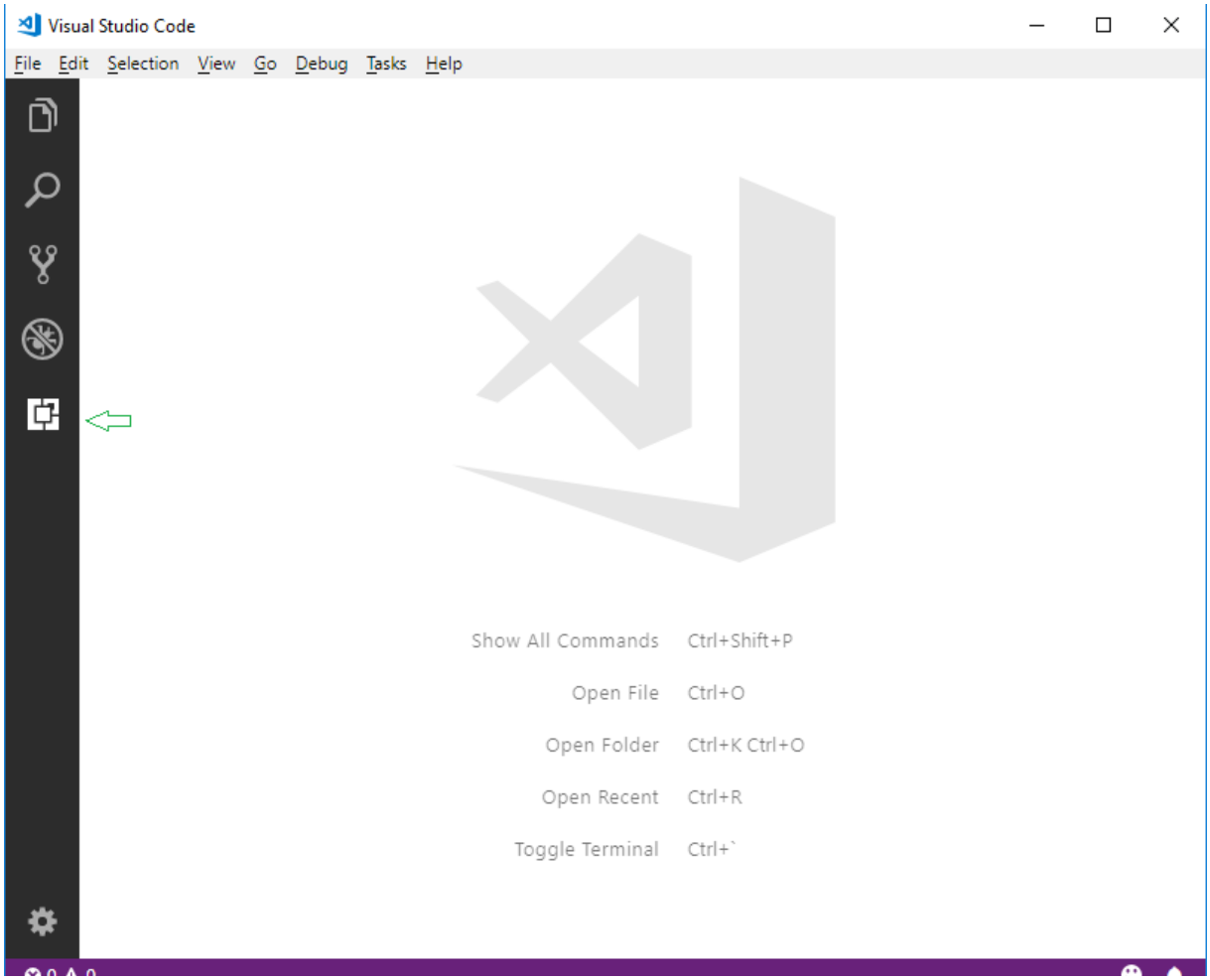
Visual Studio Code is now installed.

4 CONFIGURING VISUAL STUDIO CODE FOR MV.

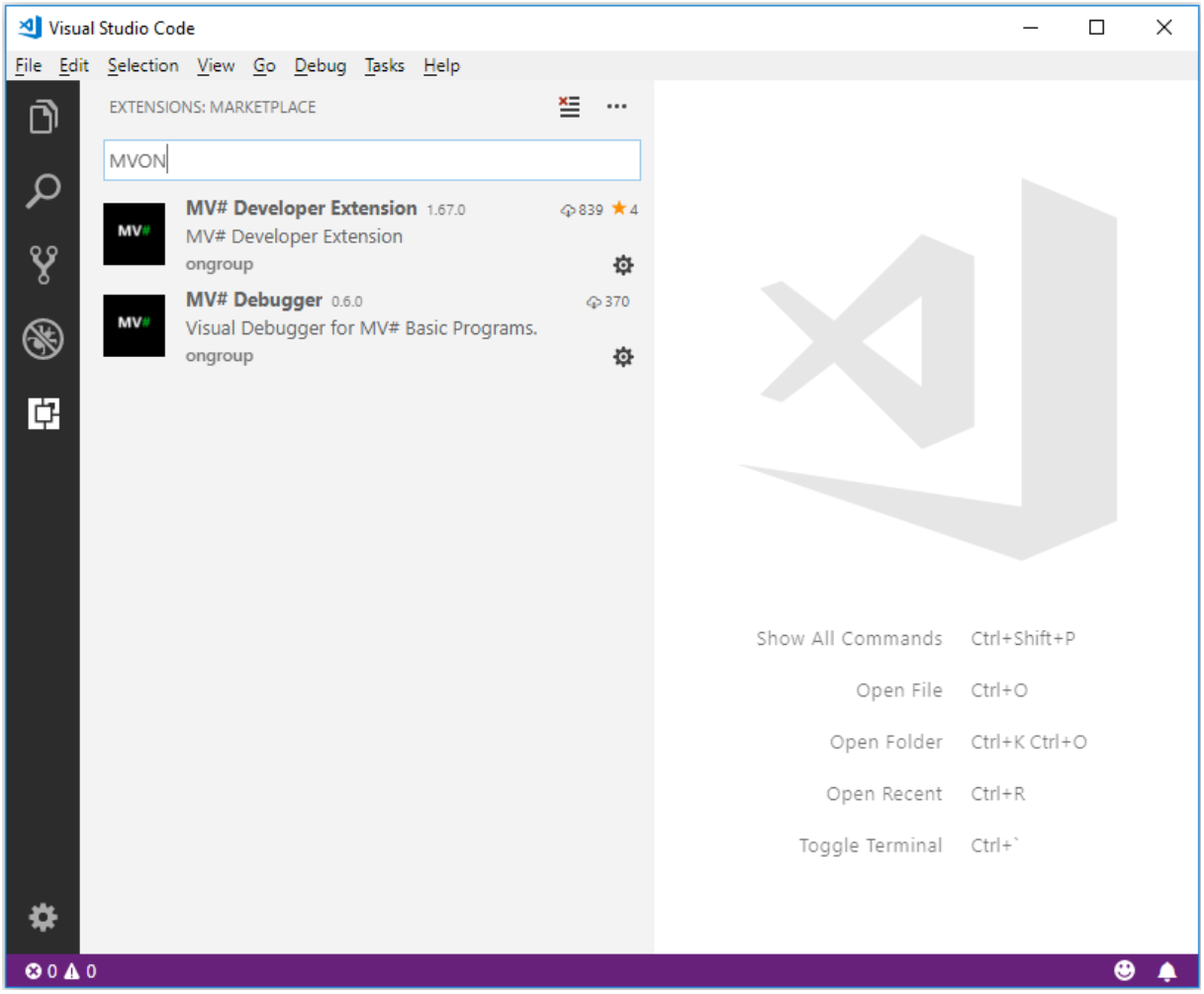
Before we can start using the MV features for Visual Studio Code, we need to install the MultiValue Basic extension. There are many extensions available for VSCODE in the Visual Studio Code Marketplace. These extensions are all free. VSCODE has a automated download and installation process for extensions.

NOTE: The current installation notes are based on the original extension provided by ONgroup called MV# Developer. After the first release the documenation will be updated to reflect the MultiValue Basic Extension. The screenshots are also from an older release of VSCODE and your screen may appear a little differently that what you see in the following images.

Start VSCODE and select the Extensions Button. This icon has changed to look like this: 



In the search box, type Multivalue.



Select the MV# Developer Extension by clicking on the item in the list then choose install.

Once the extenion is installed we are ready to start using VSCODE on your MV source code.

5 CONNECTING TO A MULTIVALUE SERVER

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

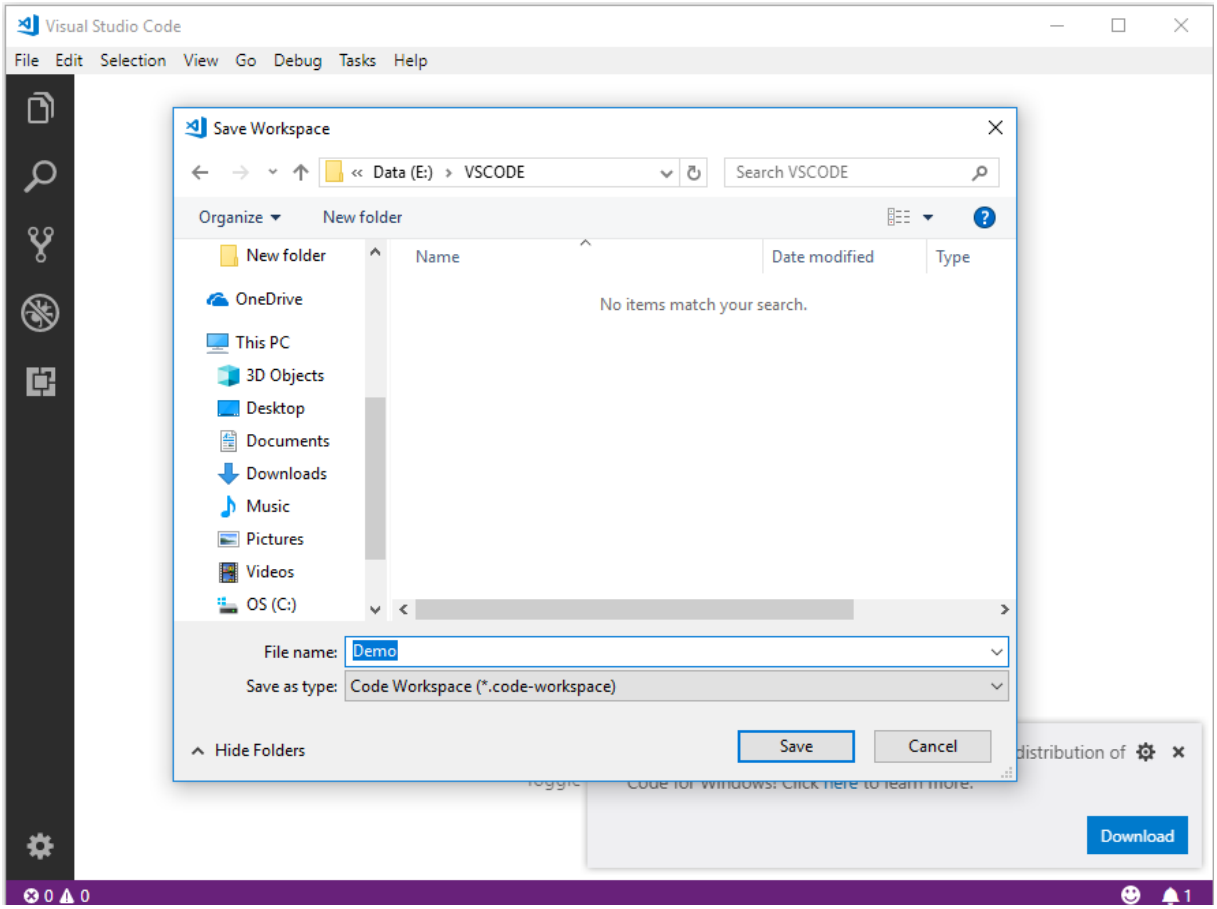
In order for VSCODE to communicate to your MV server you will need to install the MVGateway provided with the extension. The MVGateway runs as a Windows service and is installed using the Windows setup program provided. This setup program will be automatically downloaded to your system when you install the extension. When you install the extension a directory on your system is created in c:\users\{yourusername}\.vscode\extension. There will be a directory specific to this 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.

Once you have installed the MVGateway the next step is to configure a VSCODE Workspace. The Workspace will contain all the paramters 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**.

In order to connect to your MV Server, the following information is required:

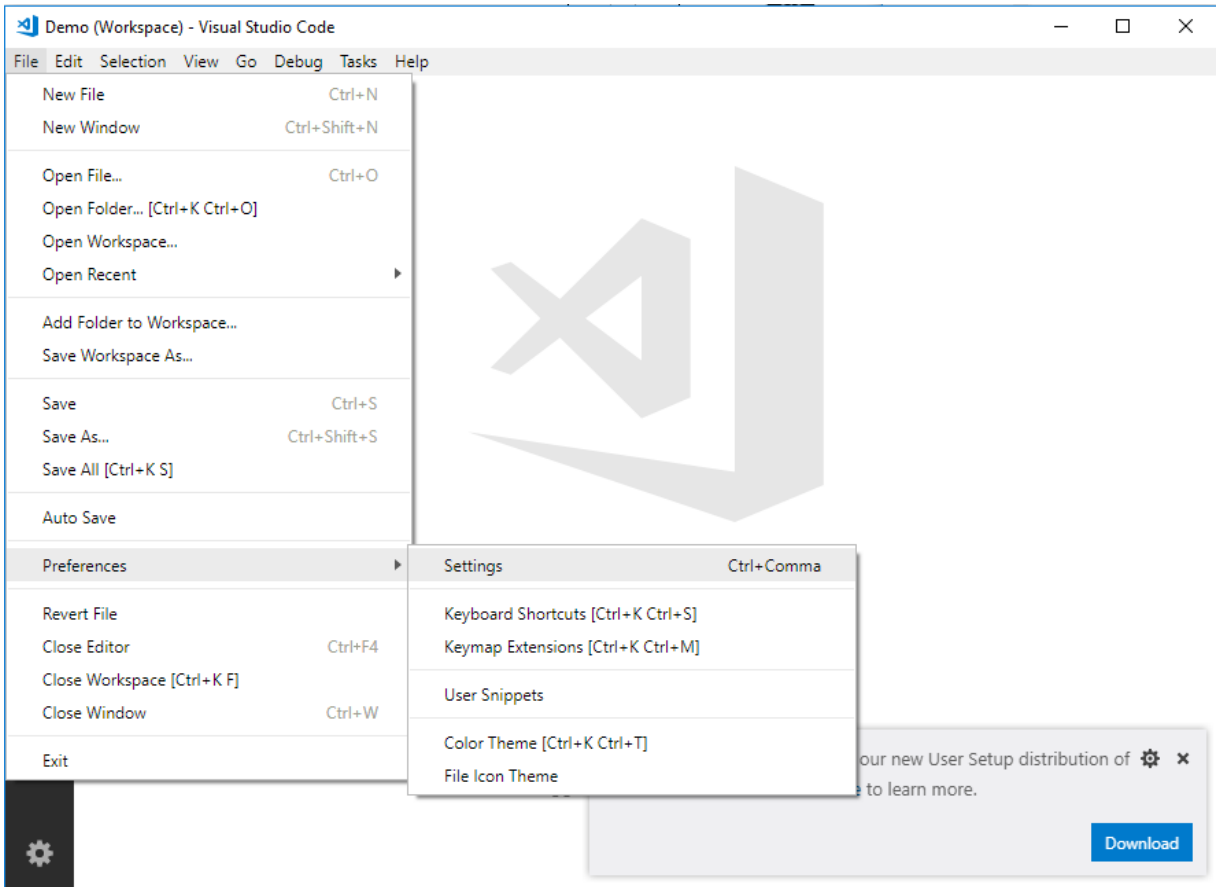
1. Hostname or IP Address of the MV server.
2. User name to login into the server
3. Password for the user above
4. Account name to connect to on the MV Server

To create a new Workspace, select "**Save Workspace As**" from **File** Menu. In this example, a folder called VSCODE on the E: drive us used to store the Workspace definitions.



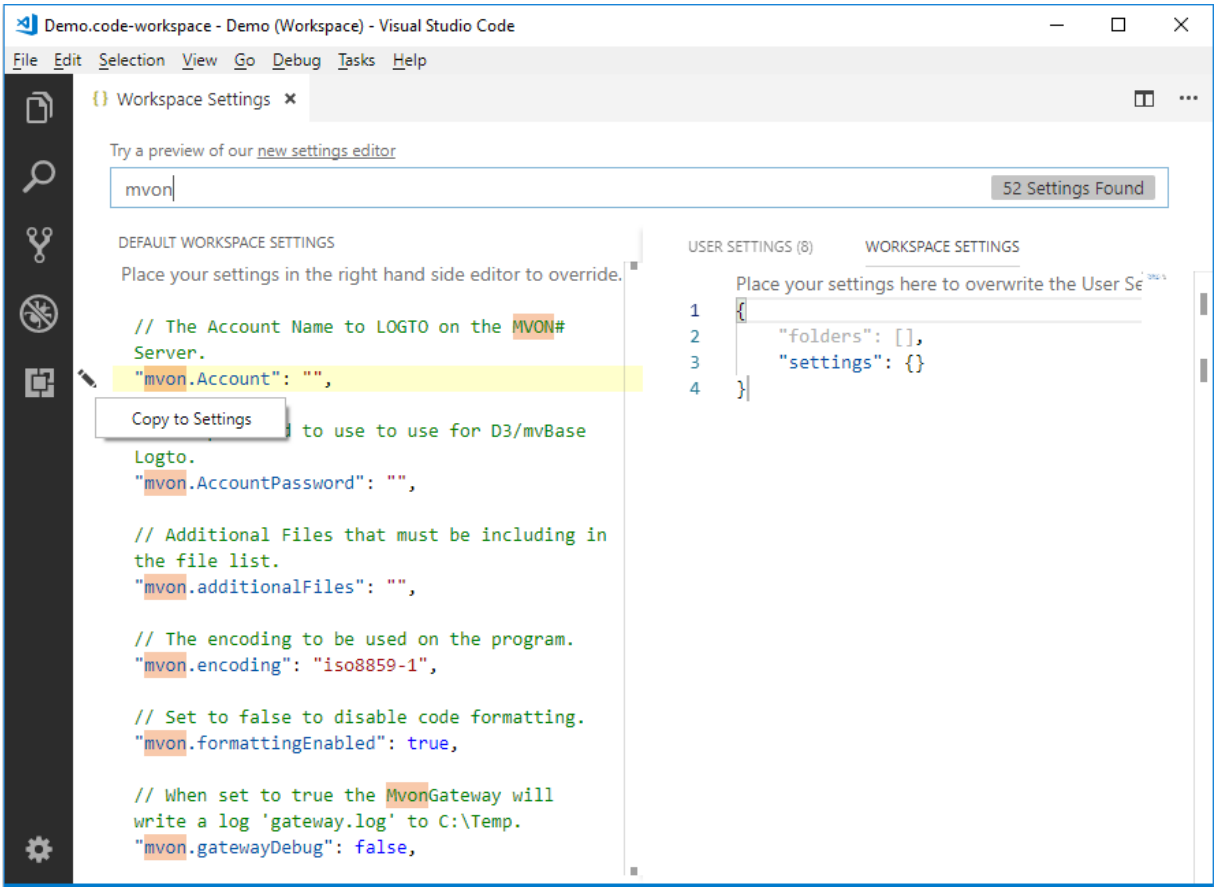
As shown in the above screenshot, a blank Workspace called Demo is created that can now be configured to point to your MV server.

To configure the connection parameters open the Workspace settings. One way to do that is to use the menu and select **File, Preference, Setting**.



This will bring up the Settings pane in VSCODE. Once this is open you MUST select the **Workspace Tab**, and then type **mv** in the search box. This will display a list off all the parameters that can be set for the MV Developer extension. This is one area that VSCODE has changed so your screen may appear a little different than these screenshots.

Also, 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. 



After adding all the parameters to the workspace, you setting should be like this:

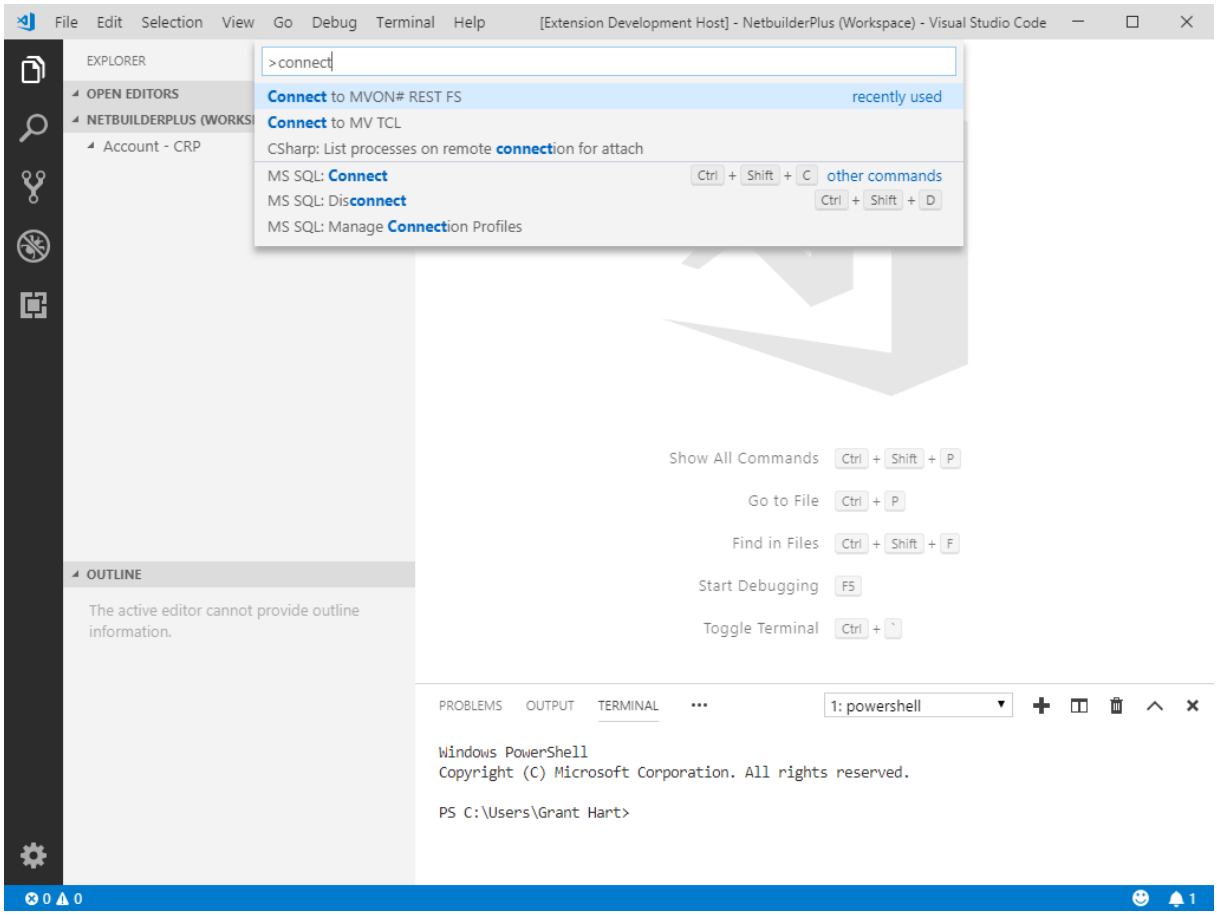
```
{
    "folders":[
        {
            "uri": "RestFS:/",
            "name": "Account - DEMO",
        }
    ],
    "settings": {
        "mvon.RestPath": "http://localhost:9005/",
        "mvon.GatewayType": "Universe",
        "mvon.UseGateway": true,
        "mvon.RemoteHost": "192.168.1.2",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

}

This is the base settings required to connect to an Universe MV Server. Press Ctl-S to save your settings. See example settings for other supported MV Servers below.

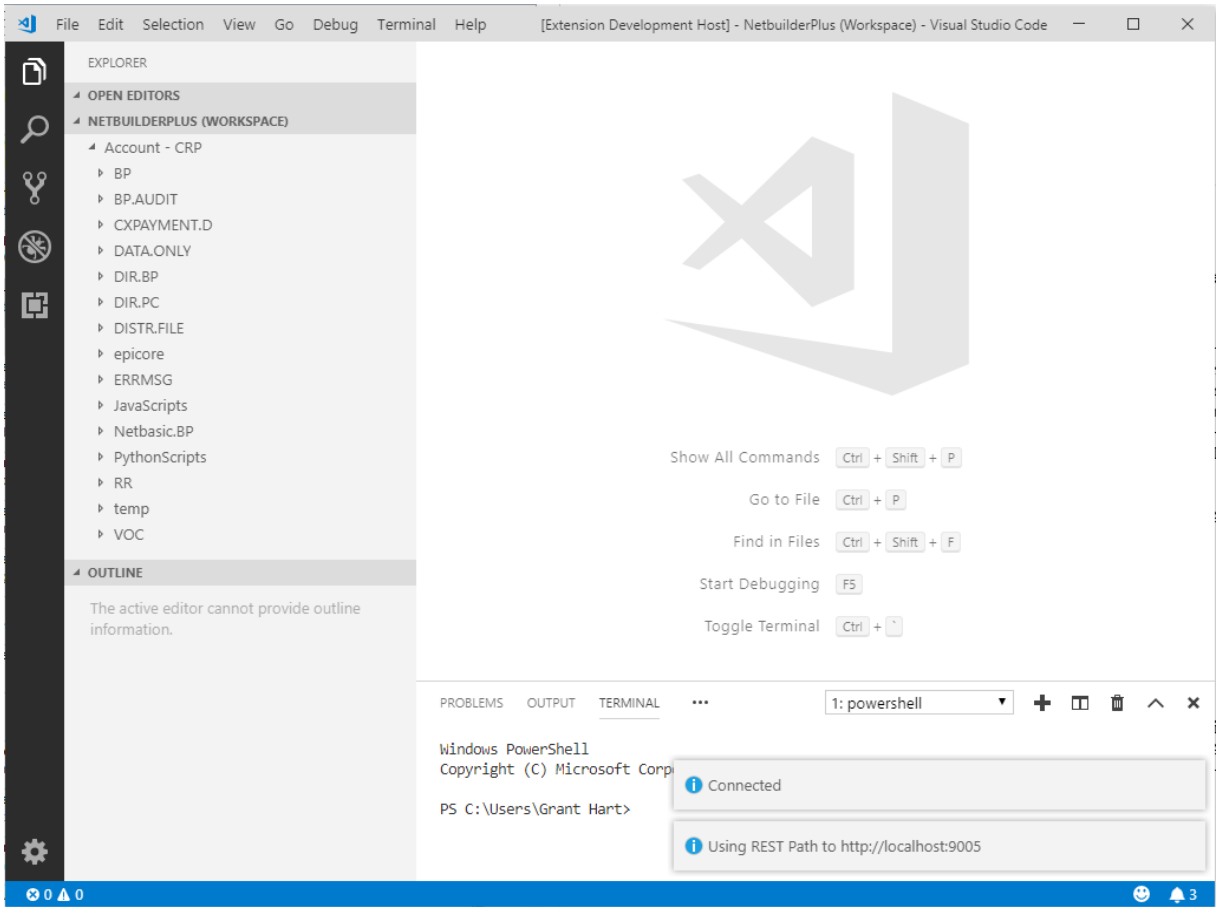
5.1 Testing the connection

We can test to if the connection to MV server works by Pressing **F1**. VSCODE will prompt you for the command to run. Type **Connect** to display all commands with Connect in it and is displays:



Select **Connect to MVON# REST FS** and the extension will connect to the server and retrieve a list of Directory files from the server.

If the connection is sucessful, the following messages will appear at the bottom left of the screen.



As each MV platform might require different parameters, a Workspace configuration example is provided for each of the following MV platforms.

1. Universe
2. Unidata
3. OpenQM
4. jBASE
5. D3
6. MvBase

5.2 Universe

```
{
    "folders":[
        {
            "uri": "RestFS:/",
            "name": "Account - Universe",
        }
    ],

    "settings": {
        “mvon.RestPath": “http://localhost:9005/”,
        "mvon.UseGateway": true,
```

```
        "mvon.GatewayType": "Universe",
        "mvon.RemoteHost": "192.168.1.2",
        "mvon.UserName": " myUserName",
        "mvon.Password": "myPassword",
        "mvon.Account": "DEMO",
        "files.associations": {
           "*":"mvon"
        }
    }
}
```

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

5.3 Unidata

```
{
    "folders":[
        {
            "uri": "RestFS:/",
            "name": "Account – Unidata",
        }
    ],

    "settings": {
        “mvon.RestPath”: “http://localhost:9005/”,
        "mvon.UseGateway": true,
        "mvon.GatewayType": "Unidata",
        "mvon.RemoteHost": "192.168.1.2",
        "mvon.UserName": "myUserName",
        "mvon.Password": "MyPassword",
        "mvon.Account": "DEMO",
        "mvon.AccountPath": "/usr/data/DEMO",
        "files.associations": {"*":"mvon"}
    }
}
```

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

5.4 OpenQM

```
{
    "folders":[
        {
            "uri": "RestFS:/",
            "name": "Account - QM",
        }
    ],

    "settings": {
        “mvon.RestPath”: “http://localhost:9005/”,
        "mvon.UseGateway": true,
        "mvon.GatewayType": "QM",
        "mvon.RemoteHost": "192.168.1.2",
        "mvon.UserName": "MyUserName",
        "mvon.Password": "MyPassword",
        "mvon.Account": "DEMO",
        "files.associations": {"*":"mvon"}
    }
}
```

Setting		Description
mvon.RestPath	http://localhost:9005/	Path to REST Gateway
Mvon.UseGateway	true	Indicate that the gateway must be used.
mvon.remoteHost	192.168.1.2	The servers IP/Host name that is running the OpenQM Database.
mvon.gatewayType	QM	Connecting to a OpenQM server
mvon.UserName	MyUserName	The Windows/UNIX user id to log into the server.
mvon.Password	MyPassword	The password for the user above.
Mvon.Account	DEMO	The account name on the QM server to connect to. This must be defined in the ACCOUNTS file in the QMSYS account.

5.5 jBASE

```
{
    "folders":[
        {
            "uri": "RestFS:/",
            "name": "Account - jBASE",
        }
    ],

    "settings": {
        “mvon.RestPath”: “http://localhost:9005/”,
        "mvon.UseGateway": true,
        "mvon.GatewayType": "jBASE",
        "mvon.RemoteHost": "192.168.1.2",
        "mvon.UserName": "MyUserName",
        "mvon.Password": "MyPassword",
        "mvon.Account": "",
        "files.associations": {"*":"mvon"}
    }
}
```

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

5.6 D3

```
{
    "folders":[
        {
            "uri": "RestFS:/",
            "name": "Account – D3",
        }
    ],

    "settings": {
        “mvon.RestPath”: “http://localhost:9005/”,
        "mvon.UseGateway": true,
        "mvon.GatewayType": "D3",
        "mvon.RemoteHost": "192.168.1.2",
        "mvon.UserName": "dm",
        "mvon.AccountPassword": "",
        "mvon.Account": "dm",
        "files.associations": {"*":"mvon"}
    }
}
```

Setting		Description
mvon.RestPath	http://localhost:9005/	Path to the REST Gateway
mvon.UseGateway	true	Indicates that the gateway must be used.
mvon.RemoteHost	192.168.137.102	The servers IP name that is running the D3 Database.
mvon.GatewayType	D3	Connecting to a D3 server
mvon.UserName	dm	The D3 User name to log in with
mvon.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.

5.7 mvBase

```
{
    "folders":[
        {
            "uri": "RestFS:/",
            "name": "Account – mvBase",
        }
    ],

    "settings": {
        “mvon.RestPath”: “http://localhost:9005/”,
        "mvon.UseGateway": true,
        "mvon.GatewayType": "mvBase",
        "mvon.RemoteHost": "192.168.1.2",
        "mvon.UserName": "MyUserName ",
        "mvon.AccountPassword": "MyPassword",
        "files.associations": {"*":"mvon"}
    }
}
```

Setting		Description
mvon.RestPath	http://localhost:9005/	Path to the REST Gateway
mvon.UseGateway	true	Indicates that the gateway must be used.
mvon.RemoteHost	192.168.137.2	The servers IP name that is running mvBase.
mvon.GatewayType	mvBase	Connecting to a mvBase server
mvon.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.

5.8 MVON#

```
{
    "folders":[
        {
            "uri": "RestFS:/",
            "name": "Account – MVON#",
        }
    ],

    "settings": {
        “mvon.RestPath”: “http://192.168.1.2/mvonrest”,
        "mvon.UseGateway": false,
        "mvon.UserName": "MyUserName ",
        "mvon.Password": "MyPassword",
        "mvon.Account: "Netbasic",
        "mvon.RemoteDebug": true,
        "files.associations": {"*":"mvon"}
    }
}
```

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

5.9 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 typicalls 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": {
        "mvon.RestPath": "http://localhost/mvonrest",
        "files.associations": {"*":"mvon"}
    }
}
```

5.10 Additional MultiValue Basic Developer Settings

The following settings are availabel to customise your VSCODE MultiValue Basic Developer experience.

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

6 MV Developer Features

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

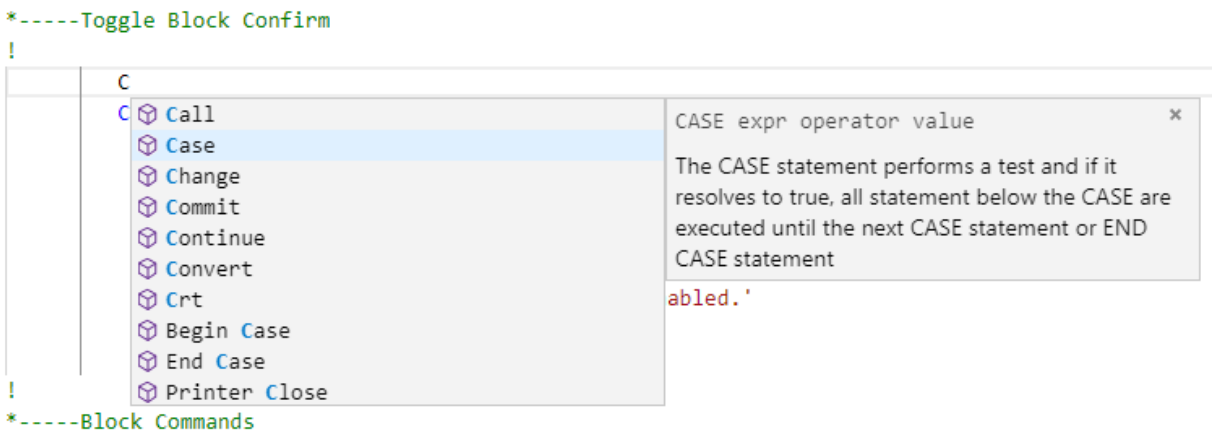
6.1 Syntax Highlighting

```
!
*-----Insert Text
!
         CASE UPCASE(ANS) = 'I' OR UPCASE(ANS) = 'IB' OR UPCASE(ANS) MATCHES
"'I '0X" OR UPCASE(ANS) MATCHES "'IB '0X"
            GOSUB 1030
!
*-----Toggle Block Confirm
!
         CASE UPCASE(ANS) = 'BLOCK'
            IF BLOCK THEN
               BLOCK = FALSE
               CRT 'BLOCK operation verification = disabled.'
            END ELSE
               BLOCK = TRUE
               CRT 'BLOCK operation verification = enabled.'
            END
            GOSUB 1000
```

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

6.2 Intellisense

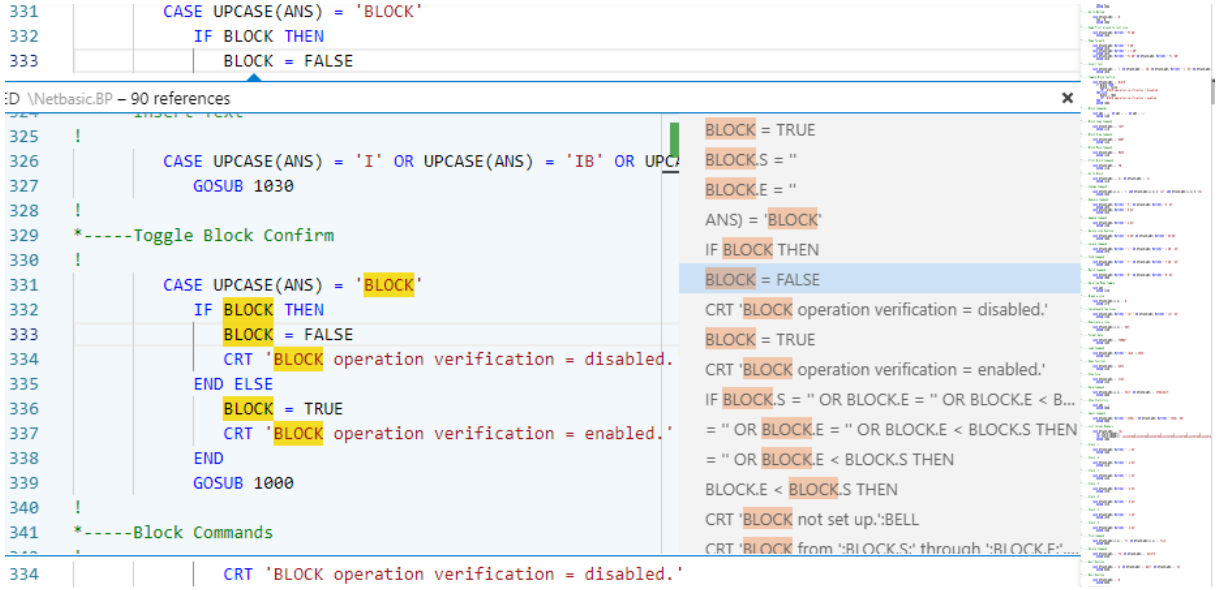
As you type your program, you will be prompted with available statements and functions including the sytax and description.



6.3 Find All References

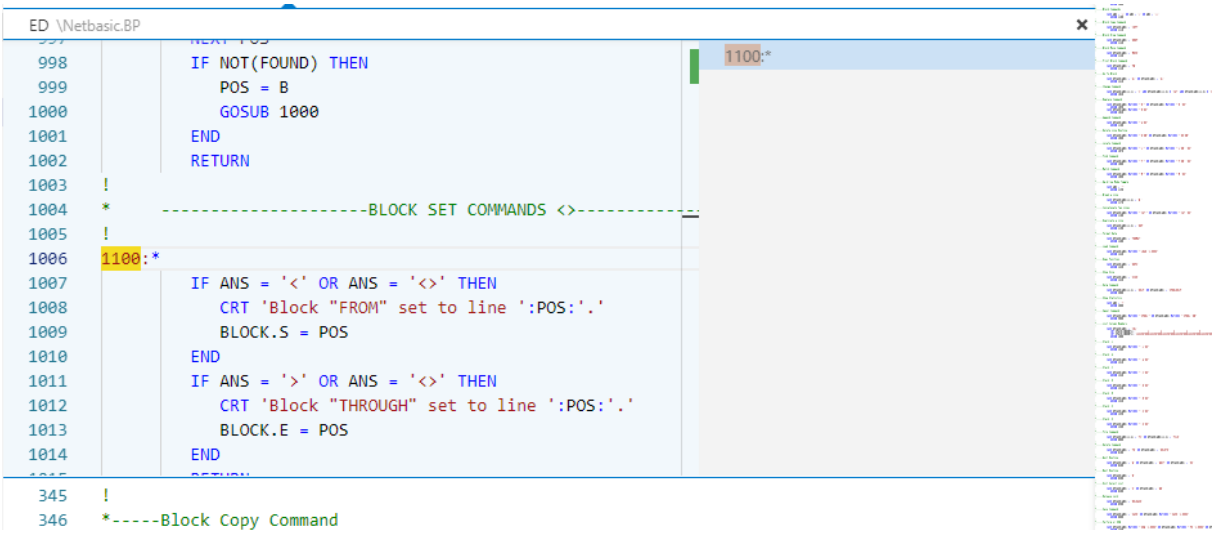
You can find all references to a word in your program by **right clicking** on a word and selecting **Find All Refrences** from the menu.

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.

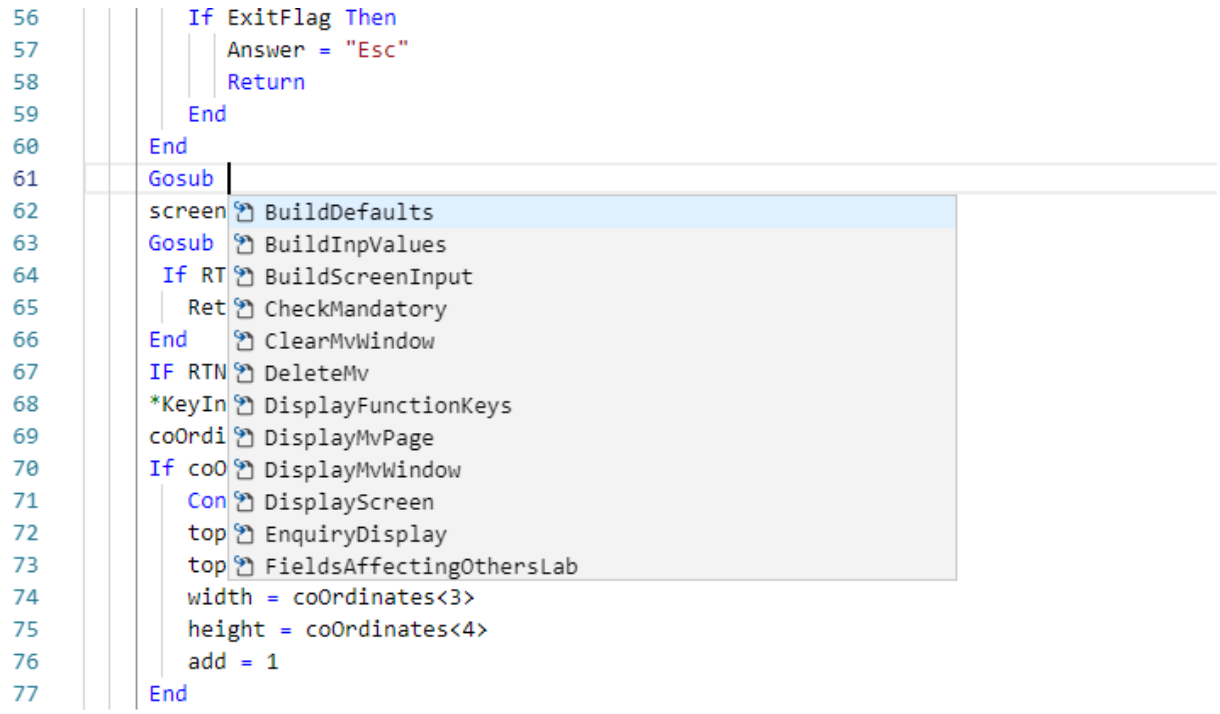
If you **right click** on a internal or external subroutine name and select **Peek Definition**, a window appears showing the internal or external subroutine.



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

6.5 Internal Subroutine lookup

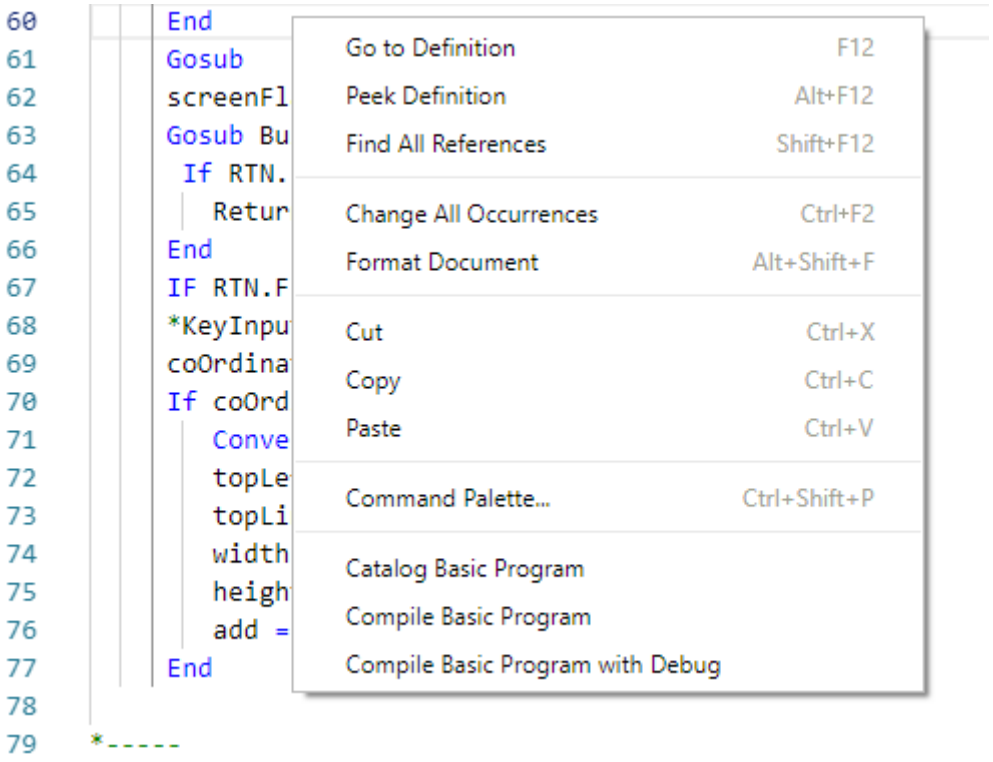
Pressing **"Ctl-space"** after the word GOTO, GOSUB or GO TO, will allow you to select from defined internal subroutines in your program.



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 occus.

6.7 Formatting Programs

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