|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Electronics Corporation of India Limited, PO.ECIL, HYDERABAD - 500 062.** | | **Doc Ref.** | **UNCL/ECIL/ASD/RTS/EC-WB** | | | |
| **Version No.** | 1.0.0 | **Date** | | 22/06/2020 |
| **Copy No.** |  | **No. of Pages** | | 82 |
| **Group** | CNID | **Project** | | **UNCL-ECSCADA-WEB-HMI** |
| **Document**  **Classification** | Secret  Confidential  Restricted  Unrestricted | | | |
| **User Manuel**  **OF ECSCADA – WEB APPLICATION** | | | | | | |
|  | **Name & Designation** | | | | **Signature** | |
| **Prepared By** | Swetha K, Software Engineer, Unical | | | |  | |
| Navya, Software Engineer, Unical | | | |  | |
| **Reviewed By** | Aparna P, Project Leader, Unical | | | |  | |
| L Rajesh, Technical Manager, SCADA R&D, ECIL | | | |  | |
| **Approved By** | Anil Pamidipalli, Delivery Manager, Unical | | | |  | |
| Mohammed Jamaluddin, Senior Managaer, SCADA R&D, ECIL | | | |  | |
| **Issue Authorized By** | Prashant Arra, Director, Unical | | | |  | |
|  | | | |  | |
| **Electronics Corporation of India Limited (ECIL)**  **COPYRIGHT:** Copyright in this work is vested in Electronics Corporation of India Limited (ECIL), Hyderabad. All information contained in this document is the sole property of Electronics Corporation of India Limited. Reproduction of this material in whole or in part is forbidden without the written permission of ECIL. | | | | | | |

**VERSION SHEET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **Version No** | **Date** | **Remarks** | **Changed By** |
| **1.** | **1.0.0** |  | **Initial Version** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[**VERSION SHEET** i](#_Toc91631406)

[Table of Contents ii](#_Toc91631407)

[1. INTRODUCTION 1](#_Toc91631408)

[1.1. PURPOSE 1](#_Toc91631409)

[1.2. SCOPE 1](#_Toc91631410)

[1.3. ASSUMPTIONS AND TOLERANCES 1](#_Toc91631411)

[1.4. ACRONYMS and ABBREVIATIONS 1](#_Toc91631412)

[1.5. REFERENCE DOCUMENTS 1](#_Toc91631413)

[2. WEB APPLICATION OVERVIEW 2](#_Toc91631414)

[3. Deployment And Installation 4](#_Toc91631415)

[3.1 server 4](#_Toc91631416)

[3.1.1 Software Requirements 4](#_Toc91631417)

[3.1.2 Installation Procedure 5](#_Toc91631418)

[3.1.2.1 Installation Process for SSMS 2018 5](#_Toc91631419)

[3.1.2.2 Installation process for Visual studio 2019 12](#_Toc91631420)

[3.1.2.3 Installation for visual studio Code 2019 18](#_Toc91631421)

[3.1.2.3 Installation Process for React Packages 22](#_Toc91631422)

[3.2 Client 23](#_Toc91631423)

[3.2.1 Software Requirements 23](#_Toc91631424)

[3.2.2 Installation Procedure 23](#_Toc91631425)

[3.2.2.1 Installation Process for SSMS 2018 23](#_Toc91631426)

[3.2.2.2 Installation process for Visual studio 2019 23](#_Toc91631427)

[3.2.2.3 Installation for visual studio Code 2019 23](#_Toc91631428)

[3.2.2.4 Installation Process for React Packages 23](#_Toc91631429)

[4. Configurations 26](#_Toc91631430)

[4.1 APIs Configuration 26](#_Toc91631431)

[4.2 XML File Configuration 26](#_Toc91631432)

[4.3 Reports Configuration 26](#_Toc91631433)

[4.4 Batch Jobs Configuration 26](#_Toc91631434)

[4.4.1 Ssis package execution process 26](#_Toc91631435)

[4.4.2 Creating sql job 26](#_Toc91631436)

[5. Application Build Process 35](#_Toc91631437)

[5.1 Web server Build Process 35](#_Toc91631438)

[5.2 Web Client Build Process 35](#_Toc91631439)

[5.3 Web APIs Build Process 35](#_Toc91631440)

[5.4 Batch Jobs Run Process 40](#_Toc91631441)

# INTRODUCTION

## PURPOSE

The purpose of this document is to present a detailed description of the designs of the SCADA HMI web application solution. Firstly, this document is intended for the programming group, to use the designs as guidelines to implement the project. Equally, this document is also for the team‟s instructor, as it fulfils one of the requirements of the project. Lastly, this document could be used for designers who try to upgrade or modify the present design of the SCADA HMI system.

## SCOPE

This document gives a detailed description of the software architecture of the inventory system. It specifies the structure and design of some of the modules discussed in the SRS. It also displays some of the use cases that had transformed into sequential and activity diagrams. The class diagrams show how the programming team would implement the specific module.

## ASSUMPTIONS AND TOLERANCES

1. IT Infrastructure / Data Centre specific activities
2. IT Hardware related issues or providing any hardware
3. Any Software maintenance other than the deployed Software during support phase
4. Other activities not mentioned explicitly in in-scope

## ACRONYMS and ABBREVIATIONS

NIL

## REFERENCE DOCUMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Document No.** | **Document Title** | **Prepared by** |
| 1 | ECIL/PUR/U-6905/D/5014 | Scope document – Purchase order | ECIL-CNID |

# WEB APPLICATION OVERVIEW

ECSCADA is indigenously developed by ECIL Hyderabad with SCADA package for monitoring and controlling various process plants and industries. ECSCADA contains desktop-based HMI/MMI with graphics editor tool. Minimal functional ECSCADA web server is also available, and data shall be transferred to web server from ECSCADA main server. Unical must design and develop the ECSCADA web application which will be running on all browsers.

The proposed SCADA Webserver application shall be developed as client -server architecture and details of each System is mentioned below.

**Server Module:** The server module shall run on Webserver. This module shall collect the SCADA data and monitored parameters as metadata structure. This module shall be used to transforming the designer model forms into webpages using XML / JSON Schemas. The alarm related data and trend values shall also be captured.

Unical Identified the below mentioned functional modules for the implementation of ECSCADA Webserver application.

1. User Management Module.
2. Project Management Module
3. Model Transformation Module
4. Project Configuration Module
5. Alarms Module
6. Events Module
7. Components Module
8. Communication Module
9. Security Module
10. Reports Module

**Client Module:** The Client module shall be intended to be used by SCADA personal to access SCADA data on their webs. This shall facilitate the user to access data from their web devices. SCADA Web application should provide access control systems on-the-go and interact with real time SCADA devices of power plant monitoring system to increase the efficiency and downtime data using PDAs / Web from anywhere at any time. Web phone application shall request the Web Server every minute for the recent values to be sent. These values shall be stored in the Web Database further analysis and display purpose.

Unical Identified the below mentioned functional modules for the implementation of ECSCADA Webserver application.

1. User Authentication Module.
2. Model Display Module
3. Data Visualization Module (Trends)
4. Alarm Management Module
5. Events Module
6. Components Module
7. Communication Module
8. Security Module
9. Timer Module
10. Settings / Configuration Module
11. Reports Module
12. Help Module

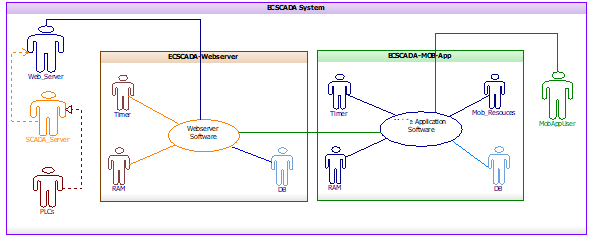


Figure 1: Context diagram of ECSCADA Web Application System

# Deployment And Installation

ECSCADA HMI web application implemented in the below mentioned technologies.

1. UI Development - React Js
2. APIs Development – ASP.Net MVC, 4.0 and Node Js
3. Database – SQL Server

Server and Clients are supposed to be in the same domain. If they are in the different domain switch/router level settings has to be changed for enabling multicasting among multiple switches/routers.

## server

Server application is provided as an executable. It should be run in a system where data is monitored through web application. This application acts as a source of data for all the available clients. A server is **a** computer program or device that provides a service to another computer program and its user, also known as the client. ... A given application in a computer might function as a client with requests for services from other programs and as a server of requests from other programs.

### Software Requirements

|  |  |  |
| --- | --- | --- |
| **S. No** | **Description** | **Installation Software** |
| 1 | UI Development | Visual studio 2019 & Visual studio Code2019 |
| 2 | APIs Development | Visual studio 2019 |
| 3 | Database – | SSMS 2018 |

### Installation Procedure

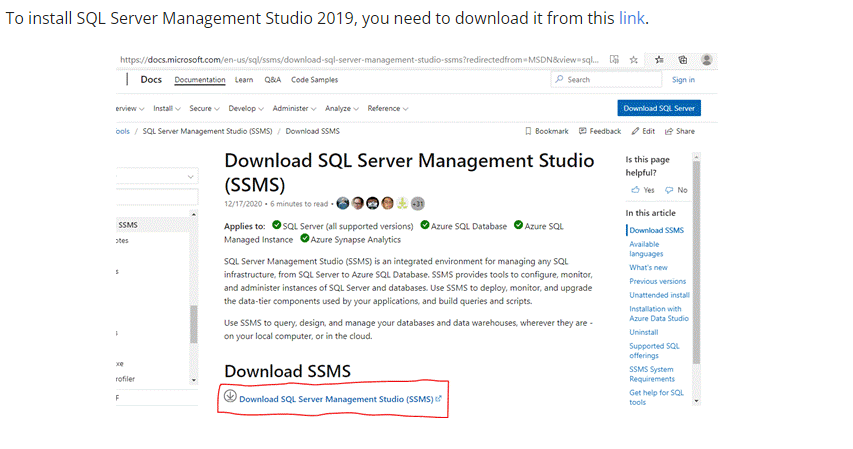
The act or process of making a machine, a service, etc., ready to be used in a certain place the act of installing something. A ceremony in which someone is put in an official or important job. something (such as a piece of equipment) that is put together and made ready for use.

### Installation Process for SSMS 2018

The latest version of SQL Server Management Studio (SSMS) is 18.8 released on December 17, 2020. The SQL Server Management Studio is software for querying, designing, and managing SQL Server on your local computer or in the cloud. It provides you with tools to configure, monitor, and administer SQL Server instances.

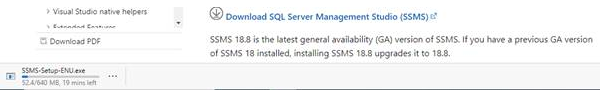
**Step:1**

To install SQL Server Management Studio 2019, you need to follow below steps



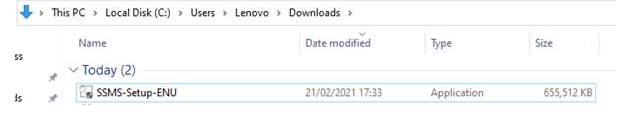
**Step:2**

Click the "Download" button for downloading the SSMS 2019 executable file on the downloaded path.

****

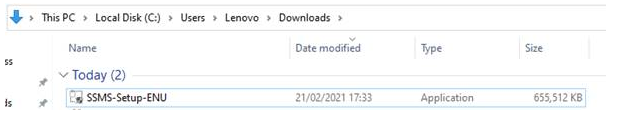
**Step:3**

Open your system download path and find the .exe file. SSMS-Setup-ENU.exe

****

**Step:4**

Double-click the .exe file SSMS-Setup-ENU.exe to starting installing. The installation process of SMSS is straightforward which you just need to follow the screen sequence.



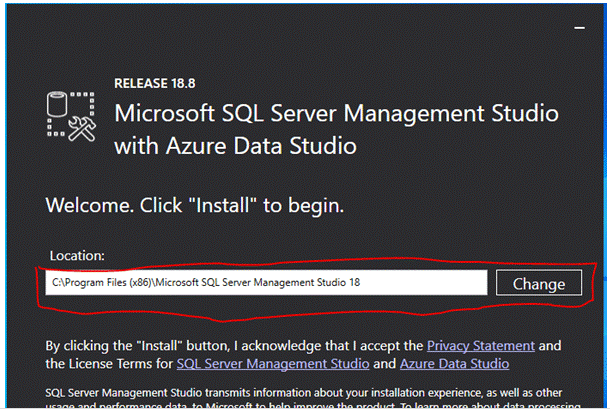
**Step:5**

After double-clicking, the system will ask the permission: “Do you want to allow the following to make a change to this computer? Click yes to continue installing the SQL Server Management Studio 2019, or...

 Click “Yes” on any security prompt.

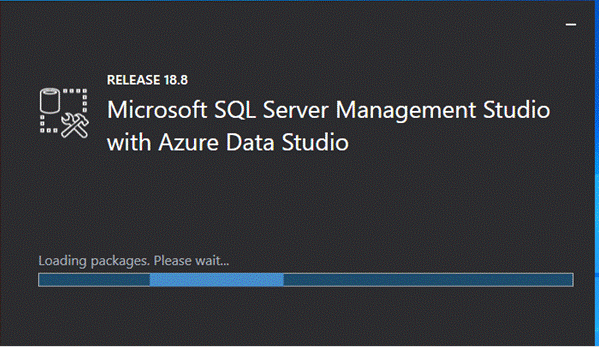
**Step:6**

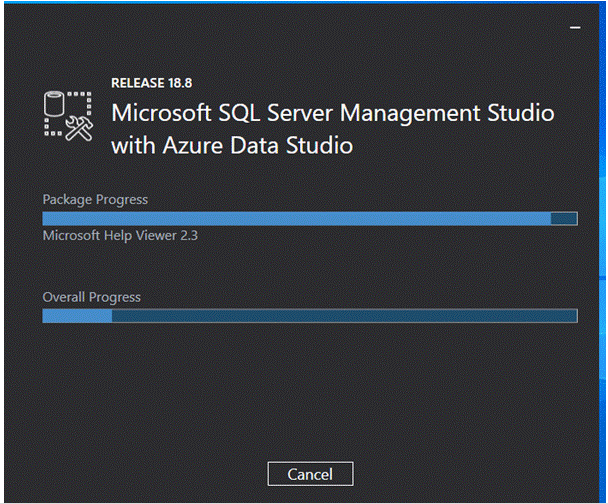
The installation window will be open after giving permission to install. Click the Install button.

****

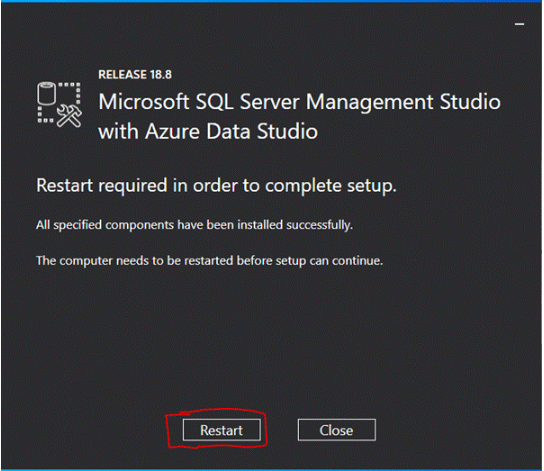
**Step:7**

After loading packages progress bar will be shown. One is Package Progress and Overall Progress. Wait for few minutes while the installer sets up the software.



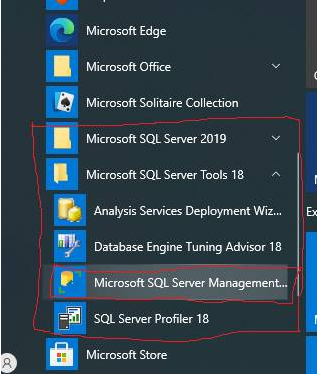
**Step:8**

Installation completed. After completing the installation restart your computer for a complete setup.



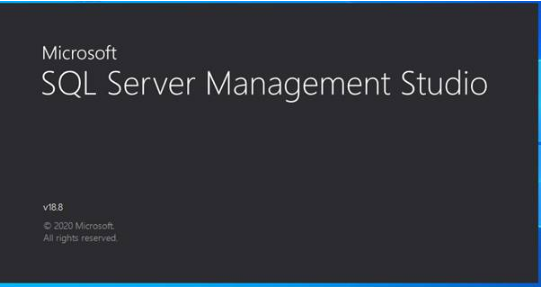
**Step:9**

Go to all programs in your systems, we can see two folders one is Microsoft SQL Server 2019 and another one is Microsoft SQL Server Tool 2018. Under Microsoft SQL Server Tools 18 you can see the Microsoft SQL Server Management Studio 18.

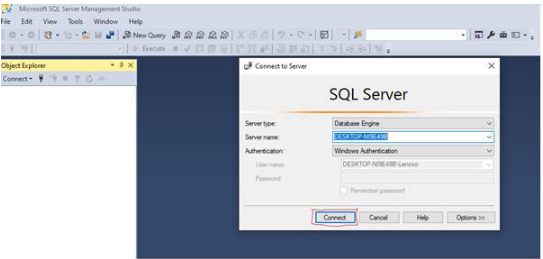
****

**Step:10**

Double-click on SQL Server Management Studio 18 and it will open looks like the below screenshot. The first time opening it will take a few minutes.

****

After opening SQL Server Management Studio 2018, we can see that it looks like the below screenshot.

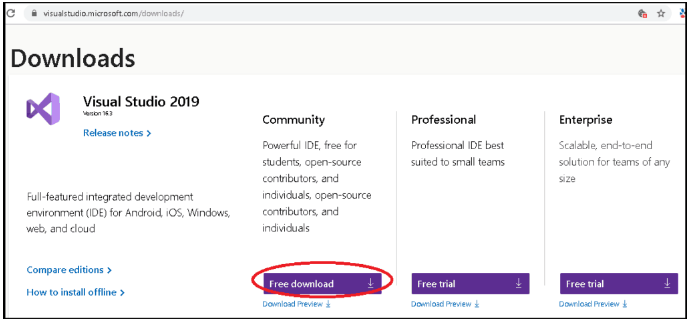
****

### Installation process for Visual studio 2019

The download and installation process is very simple

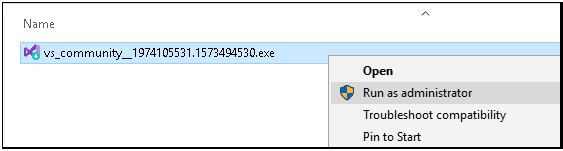
**Step:1**

Open Browser and search for visual studio 2019 and open like the below picture



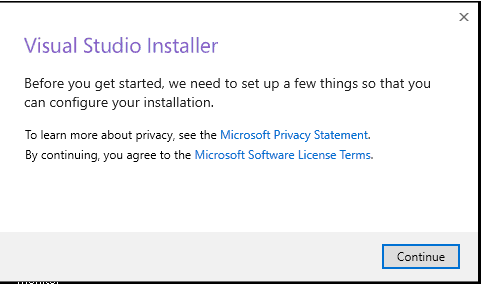
**Step:2**

Right-click and select the Run as administrator on the downloaded file to begin the installation

You will be asked for permission to continue. Click on Yes to continue with the installation.

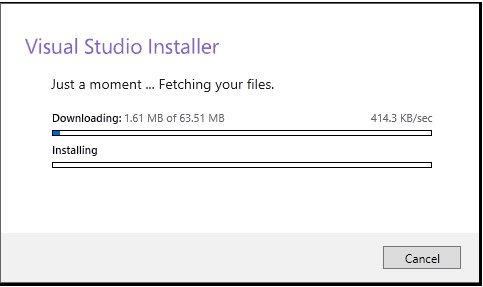
**Step:3**

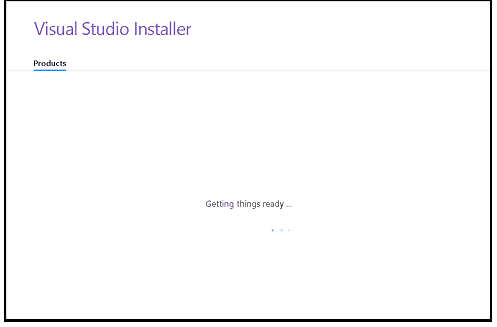
Next, you will be asked to accept the software License terms and Privacy Statement. Click on continue to go ahead with the installation.

****

**Step:4**

The Installer fetches the required files to install the Visual Studio 2017.  This process will take few minutes

****

**Step:5**

The Installer will ask for the features you wish to install. You have four options here

1. Workloads
2. Individual Components
3. Language Packs
4. Installation Location

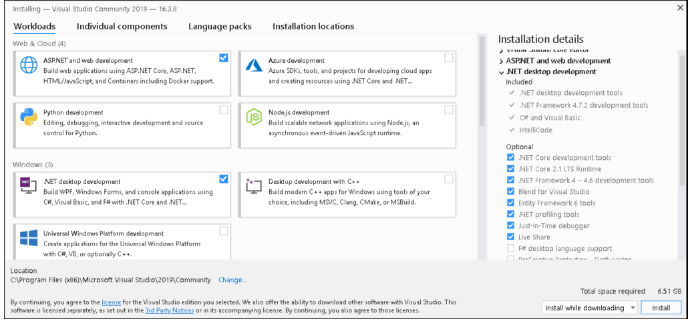
The first option is to select workloads.

A workload is a group of individual components based on the project type. For Example, if you are developing using ASP.NET for web development then you can choose **ASP.**NET and web development workload which will install the following components for you

1. C# and Visual Basic
2. .Net Framework 4.6
3. ASP.NET and web development tools
4. Entity Framework tools
5. Typescript SDK
6. IIS Express
7. NuGet Package Manager

Click here to see the complete [list of workloads in Visual Studio 2019](https://docs.microsoft.com/en-us/visualstudio/install/workload-component-id-vs-enterprise?view=vs-2019)

Select the required Workload. I have chosen **.NET Desktop development** & **ASP.NET and web development**.



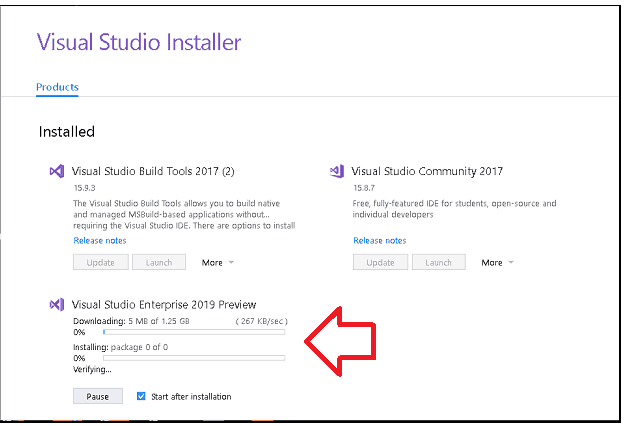
**Step:6**

The Second Option gives an exhaustive list of components that you can install.  Choosing the workloads automatically selects the components that are part of that workload.  You can further customize what you want to install from this option.

Select the required components



Now, click on the Install button to begin the installation of Visual Studio 2019.

****

The installer will now download each component from the internet and starts the installation. This will take a while depending on your internet speed.

Once the installation is complete, you will be presented with the **Installation succeeded** message along with the option to register

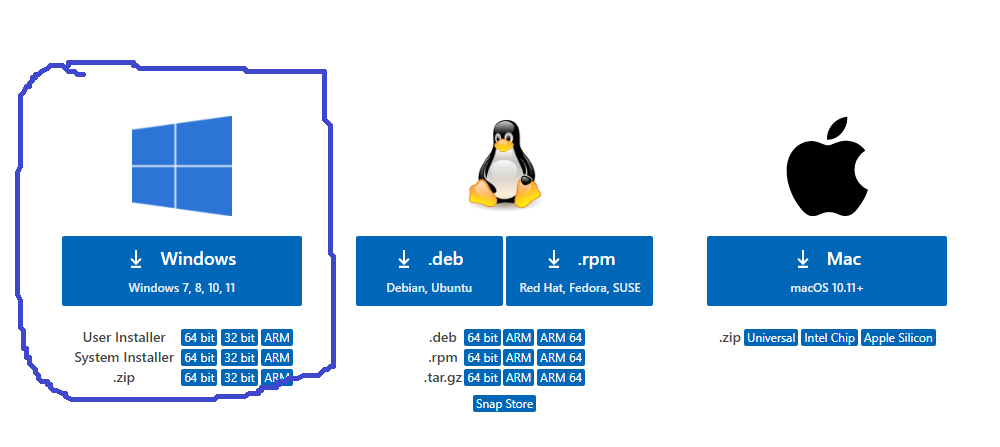
If you are installing Visual Studio for the first time you will be asked to sign in

### Installation for visual studio Code 2019

**Step:1**

First we have to open browse then search for install visual studio code 2019 then below

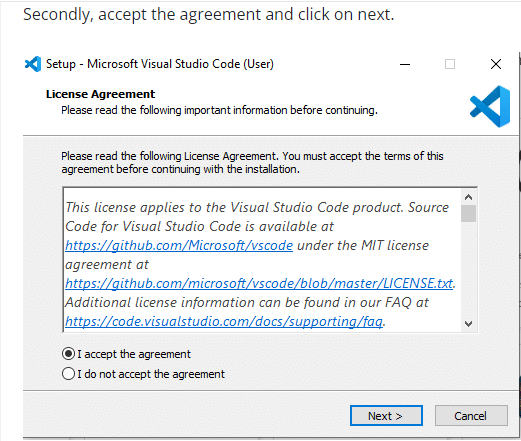
Window will be opened. Then click on Download.

****

The installer will now download.

**Step:2**

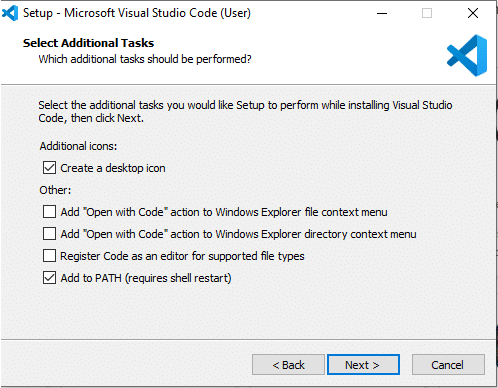
Firstly, download the Visual Studio Code installer for Windows. Once it is downloaded, run the installer *(VSCodeUserSetup-{version}.exe).* It will only take a minute. Secondly, accept the agreement and click on next.



Thirdly, click on ***"create a desktop icon"*** so that it can be accessed from desktop and click

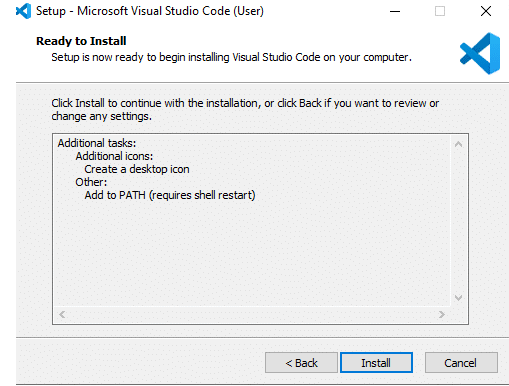
On Next.

**Step:3**

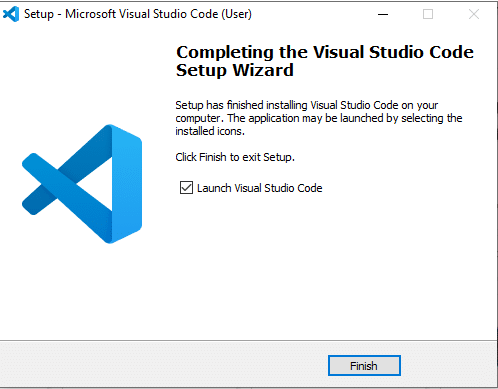
****

After that, click on the install button.

**Step:4**

****

Finally, after installation completes, click on the finish button, and the visual studio code will get open.



### Installation Process for React Packages

|  |  |  |
| --- | --- | --- |
| **S. No** | **Package Name** | **Package Installation Command** |
| 1 | axios | npm install axios |
| 2 | testing-library | npm install testing-library |
| 3 | concurrently | npm install concurrently |
| 4 | cors | npm install cors |
| 5 | express | npm install express |
| 6 | fast-xml-parser | npm install fast-xml-parser |
| 7 | formik | npm install formik |
| 8 | js-base64 | npm install js-base64 |
| 9 | konva | npm install konva |
| 10 | list-react-files | npm install list-react-files |
| 11 | moment | npm install moment |
| 12 | multer | npm install multer |
| 13 | nodemon | npm install nodemon |
| 14 | npm | npm install npm |
| 15 | react | npm install react |
| 16 | react-autocomplete-hint | npm install react-autocomplete-hint |
| 17 | react-bootstrap | npm install react-bootstrap |
| 18 | react-csv | npm install react-csv |
| 19 | react-csv-downloader | npm install react-csv-downloader |
| 20 | react-datepicker | npm install react-datepicker |
| 21 | react-dom | npm install react-dom |
| 22 | react-form-with-constraints | npm install react-form-with-constraints |
| 23 | react-icons | npm install react-icons |
| 24 | react-konva | npm install react-konva |
| 25 | react-paginate | npm install paginate |
| 26 | react-pro-sidebar | npm install react-pro-sidebar |
| 27 | react-redux | npm install react-redux |
| 28 | react-router-dom | npm install react-router-dom |
| 29 | react-scripts | npm install react-scripts |
| 30 | react-select | npm install react-select |
| 31 | react-toastify | npm install react-toastify |
| 32 | react-xml-parser | npm install react-xml-parser |
| 33 | reactstrap | npm install reactstrap |
| 34 | redux | npm install redux |
| 35 | redux-persist | npm install redux-persist |
| 36 | utils | npm install utils |
| 37 | web-vitals | npm install web-vitals |

## Client

Server application is provided as an executable. It should be run in a system where data is monitored through web application. This application acts as a source of data for all the available clients. A server is **a** computer program or device that provides a service to another computer program and its user, also known as the client. ... A given application in a computer might function as a client with requests for services from other programs and as a server of requests from other programs.

### Software Requirements

|  |  |  |
| --- | --- | --- |
| **S. No** | **Description** | **Installation Software** |
| 1 | UI Development | Visual studio 2019 & Visual studio Code2019 |
| 2 | APIs Development | Visual studio 2019 |
| 3 | Database – | SSMS 2018 |

### Installation Procedure

The act or process of making a machine, a service, etc., ready to be used in a certain place the act of installing something. A ceremony in which someone is put in an official or important job. something (such as a piece of equipment) that is put together and made ready for use.

### Installation Process for SSMS 2018

**Refer Section 3.1.2.1**

### Installation process for Visual studio 2019

**Refer Section 3.1.2.2**

### Installation for visual studio Code 2019

**Refer Section 3.1.2.3**

### Installation Process for React Packages

|  |  |  |
| --- | --- | --- |
| **S. No** | **Package Name** | **Package Installation Command** |
| 1 | apexcharts | npm install apexcharts |
| 2 | fortawesome | npm install fortawesome |
| 3 | @testing-library/jest-dom | npm install @testing-library/jest-dom |
| 4 | @uiw/react-color-github | npm install @uiw/react-color-github |
| 5 | @uiw/react-color-sketch | npm install @uiw/react-color-sketch |
| 6 | axios | npm install axios |
| 7 | bootstrap | npm install bootstrap |
| 8 | canvas | npm install canvas |
| 9 | canvasjs | npm install canvasjs |
| 10 | canvasjs-react-charts | npm install canvasjs-react-charts |
| 11 | chart.js | npm install chart.js |
| 12 | chartjs-plugin-zoom | npm install chartjs-plugin-zoom |
| 13 | fast-xml-parser | npm install fast-xml-parser |
| 14 | highcharts | npm install highcharts |
| 15 | highcharts-react-official | npm install highcharts-react-official |
| 16 | js-base64 | npm install js-base64 |
| 17 | konva | npm install konva |
| 18 | list-react-files | npm install list-react-files |
| 19 | moment | npm install moment |
| 20 | mxgraph | npm install mxgraph |
| 21 | react | npm install react |
| 22 | react-advanced-datetimerange-picker | npm install react-advanced-datetimerange-picker |
| 24 | react-apexcharts | npm install react-apexcharts |
| 25 | react-blink-text | npm install react-blink-text |
| 26 | react-bootstrap | npm install react-bootstrap |
| 27 | react-chartjs-2 | npm install react-chartjs-2 |
| 28 | react-color | npm install react-color |
| 29 | react-confirm-alert | npm install react-confirm-alert |
| 30 | react-csv | npm install react-csv |
| 31 | react-csv-downloader | npm install react-csv-downloader |
| 32 | react-datepicker | npm install react-datepicker |
| 33 | react-dates | npm install react-dates |
| 34 | react-datetime | npm install react-datetime |
| 35 | react-datetime-picker | npm install react- datetime-picker |
| 36 | react-datetime-range-picker | npm install react-datetime-range-picker |
| 37 | react-day-picker | npm install react-day-picker |
| 38 | react-dom | npm install react-dom |
| 39 | react-google-charts | npm install react-google-charts |
| 40 | react-hammerjs | npm install react-hammerjs |
| 41 | react-konva | npm install react-konva |
| 42 | react-line-graph | npm install react-line-graph |
| 43 | react-mxgraph | npm install react-mxgraph |
| 44 | react-native | npm install react-native |
| 45 | react-native-web | npm install react-native-web |
| 46 | react-paginate | npm install react-paginate |
| 47 | react-redux | npm install react-redux |
| 48 | react-router-dom | npm install react-router-dom |
| 49 | react-scripts | npm install react-scripts |
| 50 | react-scroll | npm install react-scroll |
| 51 | react-select | npm install react-select |
| 51 | react-simple-maps | npm install react-simple-maps |
| 53 | react-toastify | npm install react-toastify |
| 54 | react-typist | npm install react-typist |
| 55 | react-widgets | npm install react-widgets |
| 56 | react-xml-parser | npm install react-xml-parser |
| 57 | reactjs-popup | npm install reactjs-popup |
| 58 | redux | npm install redux |
| 59 | redux-persist | npm install redux-persist |

# Configurations

## APIs Configuration

API paths are globally configured in utility.jsx.so we can configured from utility.jsx.

## XML File Configuration

Make sure xml files path properly configured in Uncacanvas.jsx.

## Reports Configuration

Make sure report files path properly configured in Report\_shift.js and Report\_daily.js

## Batch Jobs Configuration

### Ssis package process

**Step 1:** Create a simple package, one OLEDB source and Flat file destination.

**Step 2:** **Go to the Flat File connection properties.**

**Step 3: Go to the expression property for the same.**

**Step 4:** **Select the property as connection string and when you open the expression type the below code as it is:**

**For example: let's say my package name is: "ImportDatatoFlatFile"**

**"E:\\unical\\DynamicFlatFiles"+@ [System::Package Name] +""+REPLACE ((DT\_STR, 30, 1252) GETDATE (),":","")**

**"E:\\A-Learnings\\Miscellaneous\\TextFiles\\"+@ [System::Package Name] +""+REPLACE ((DT\_STR, 30, 1252) GETDATE (),":","")**

**For the first time, we need to browse the connection to a flat file, to retrieve the input columns and**

**Their properties, but after the connection manager property change, it's not required.**

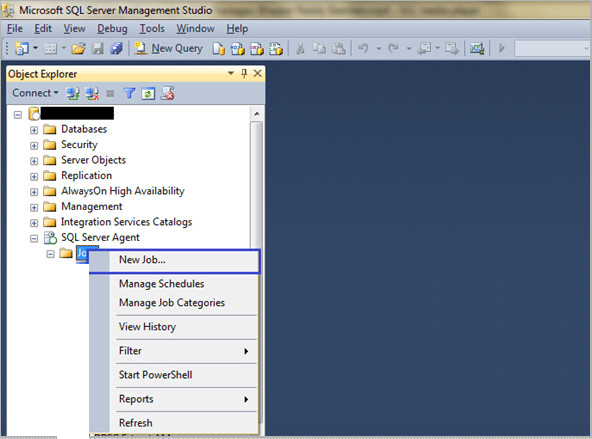
### Creating sql job

**Step: 1**

Follow the below steps to create a new job.

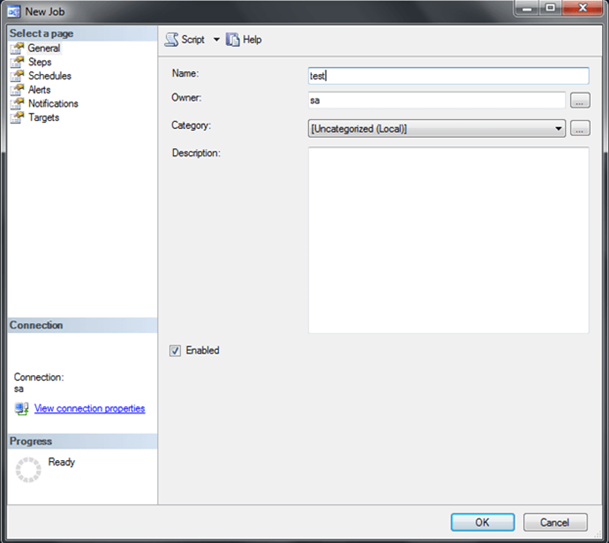
* Open the SQL Server Management Studio on Windows and you should have the “*SQL Server Agent*” to implement the automated jobs.
* Expand the “*SQL Server Agent*” in Object Explorer.
* Right click on the Jobs and select the “New Job…”

You can see the “New Job” window

****

**Step: 2**

Here, you should give a valid name for the new job. And then click on “OK” button.

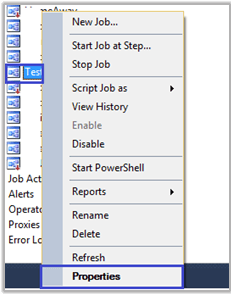


Finally, you can see that the job has been created on the “*SQL Server Agent*” list.

Once this is done, right click on the SQL Agent and refresh it.

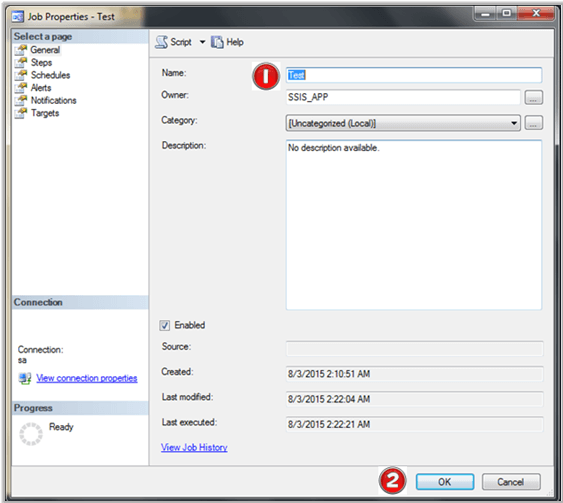
Find the below screenshot for your reference (The job has been created).

Then click on “*Property*” option to set the job configurations.

****

**Step: 3 General Page**

In the general page, you should give a name for Scheduling the job and then click on “OK”.

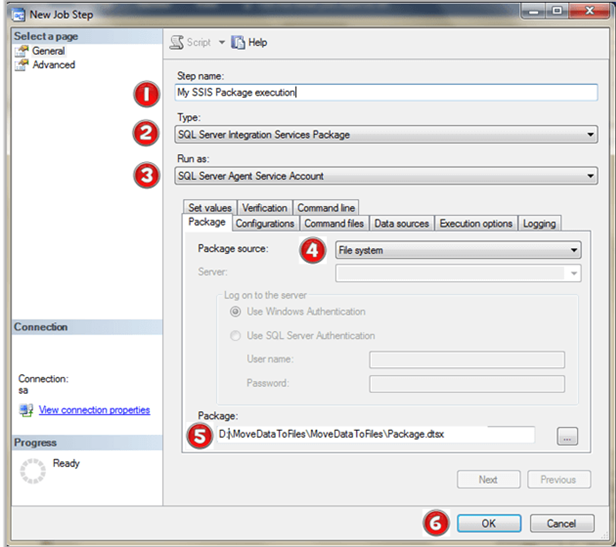


**Step: 4**

Then select the “*Steps*” option from the menu list.

You will see the below screen. Follow the below steps for the JOB configurations.

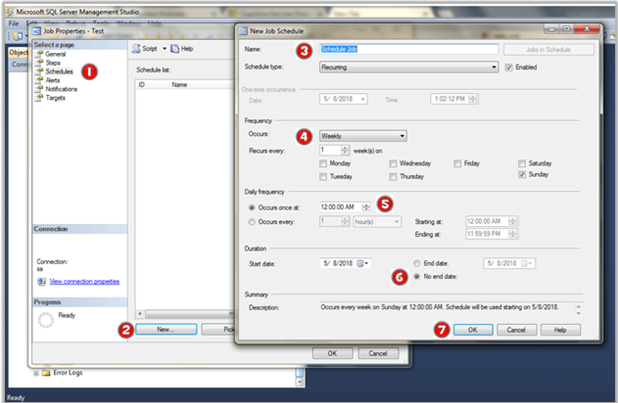
1. Enter the step name
2. The Type should be “SQL Server Integration Services Package”
3. The Run as mode should be “SQL Server Agent Service Account”
4. And change Package Source from SSIS Package to File System.
5. Then select the your SSIS package file, where you have stored locally in your machine/server
6. Then click on “OK”.

****

**Step: 5 - Schedule Page**

Follow the steps to schedule the JOB timing.

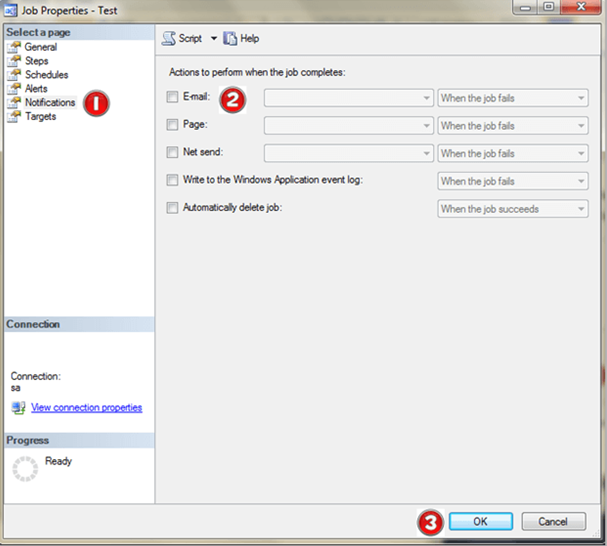
1. Click on “Schedule” option from the menu
2. You can see that “New” button in bottom of the window. Click on it.
3. Enter the name to “Schedule a JOB”
4. Mention the timing that, when the JOB wants to execute Daily, Weekly, Monthly or Yearly.
5. Set the timing to execute your operation AM/PM
6. If you want to continue your JOB till end, set “*No end date*” else set how long the JOB wants to execute.
7. Click on “*OK*”



Now, the created schedule will be displayed on the schedules list.

**Step: 6**

It will send you the email notification alert, when the JOB succeeds or fails.

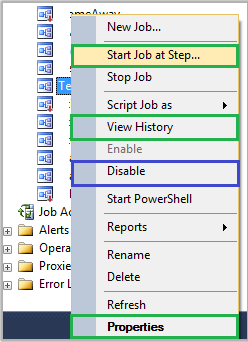


Click on the e-mail option and select the JOB mode. Then click on “OK”.

**Step: 7**

These are the properties to Start/Stop/View/Disable a JOB.

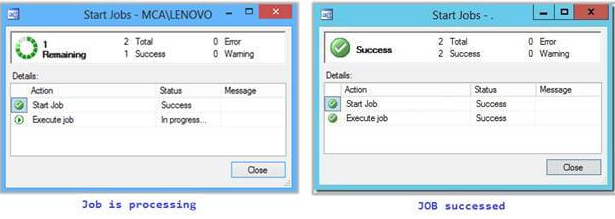
* Start Job at Step - It will execute your JOB
* Stop Job - The Job execution process will be stopped
* View History - You can see the successes/failure history of a JOB
* Enable - To enable the JOB for package execution
* Disable - It will stop the JOB execution.
* Properties - You can do the configuration for the JOB



Right click on the Job name and click the “Start Job at Step”.

The JOB will be starting to execute.

Something like the below screenshot where you can see the updates.



Once this is complete, you can see the changes as per your SSIS logic.

# Application Build Process

The term build may refer to the process by which source code is converted into a stand-alone form that can be run on a computer or to the form itself. One of the most important steps of a software build is the compilation process, where source code files are converted into executable code.

## Web server Build Process

In order to build the application we need to run the following command **npm run build** This creates a build directory inside the root directory, which bundles your React app and minifies it into simple HTML, CSS, and JavaScript files. This build folder serves your app via a simple entry point, index.html, where your entire React app resides. Running your app via a remote server means running this index.html file on the server.

## Web Client Build Process

In order to build the application we need to run the following command **npm run build** This creates a build directory inside the root directory, which bundles your React app and minifies it into simple HTML, CSS, and JavaScript files. This build folder serves your app via a simple entry point, index.html, where your entire React app resides. Running your app via a remote server means running this index.html file on the server.

## Web APIs Build Process

**Step 1:**

Start Visual Studio.

**Step 2:**

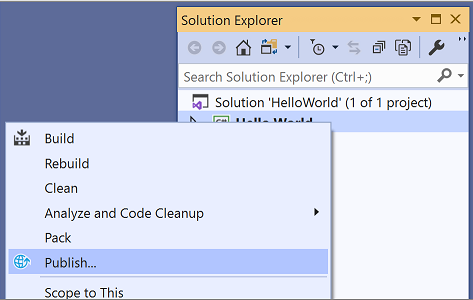
1. Open the project that you created in [Create a .NET API application using Visual Studio](https://docs.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio).

**Step** **3**:

Make sure that Visual Studio is using the Release build configuration. If necessary, change the build configuration setting on the toolbar from **Debug** to **Release**.

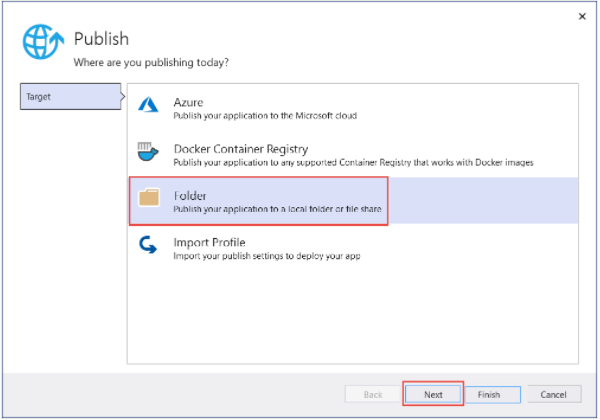
**Step** **4**:

Right-click on the project (not the solution) and select **Publish** from the menu.



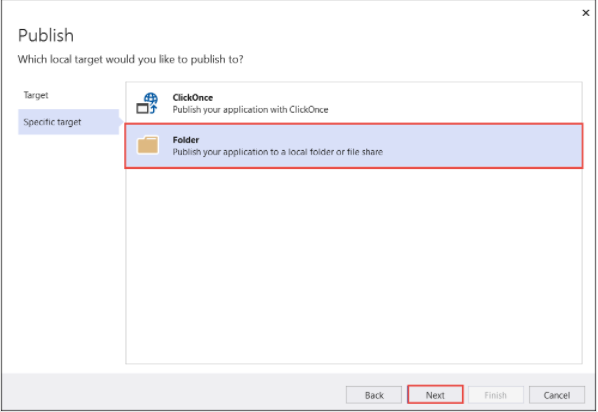
**Step** **5**:

On the **Target** tab of the **Publish** page, select **Folder**, and then select **next**.



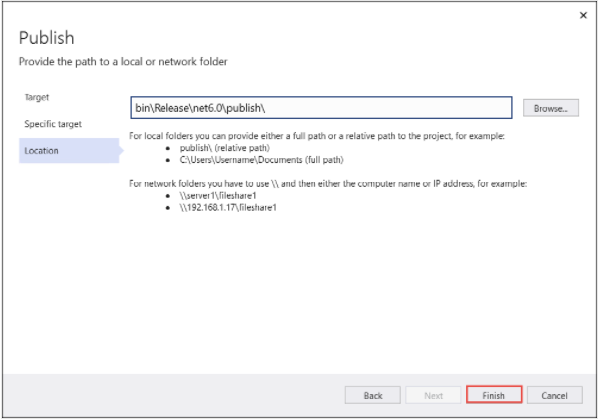
**Step** **6**:

On the **Specific Target** tab of the **Publish** page, select **Folder**, and then select **next**.



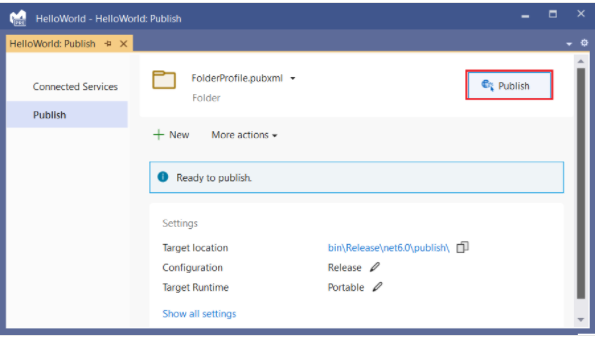
**Step** **7**:

On the **Location** tab of the **Publish** page, select **Finish**.



**Step** **8**:

On the **Publish** tab of the **Publish** window, select **Publish**.



## Batch Jobs Run Process

1. Log on to the Database Server computer with an Administrator account.
2. Start Microsoft SQL Server Management Studio.
3. In the left pane, expand SQL Server Agent > Jobs.
4. Right-click the job you want to start, and then click Start Job at Step.
5. On the Start Jobs window, review any messages.