



EPSON RC+ 8.0 擴展功能 RC+ Extensions 8.0

翻譯版

© Seiko Epson Corporation 2026

Rev.3
TCM269S9172F

目錄

1. 前言	3
1.1 前言	4
1.2 商標	4
1.3 本手冊中的商標註釋	4
1.4 注意	4
1.5 製造商	4
1.6 聯絡資訊	4
1.7 閱讀本手冊之前	5
2. 概述	6
2.1 概述	7
3. Extension 的使用	9
3.1 概述	10
3.2 下載 Extension	10
3.3 安裝 Extension	12
3.4 Extension 的解除安裝	14
4. Extension 的種類	15
4.1 概述	16
5. RC+ 自定義功能的實作	17
5.1 概述	18
5.2 安裝	18
5.3 開始使用	19
5.4 Extension 的發佈	21
5.5 試著建立看看吧	22
6. PC Vision 自定義視覺物件的實作	32
6.1 概述	33
6.2 安裝	33
6.3 開始使用	33
6.4 Extension 發佈	36

1. 前言

1.1 前言

感謝您購買我們的機器人產品。

本手冊包含正確使用RC+ Extensions的必要資訊。

在使用機器人系統之前，請詳閱本手冊及其他相關手冊。

閱讀之後，請妥善保管本手冊以供隨時取用。

所有機器人系統與其選配部件經嚴格的品質控管、測試與檢驗，以確保其符合我們的高效能標準，始能出貨給貴客戶。請注意，若未依本手冊說明的使用條件與產品規格使用本機器人系統，將無法發揮產品的基本性能。

本手冊說明我們可預測的可能危險及後果。務必遵守本手冊的安全注意事項，確保安全及正確地使用機器人系統。

1.2 商標

Microsoft、Windows、Windows、Visual Basic、Visual C++標誌皆為Microsoft Corporation在美國及其他國家的註冊商標或商標。

其他品牌及產品名稱皆為個別擁有者所有之商標或註冊商標。

1.3 本手冊中的商標註釋

Microsoft® Windows® 10作業系統

Microsoft® Windows® 11作業系統

於本手冊中，Windows 10、Windows 11分別表示上述作業系統。某些情況下，Windows意指Windows 10與Windows 11。

1.4 注意

如未獲授權，不得複製或重製本手冊中的任何部分。

本手冊內容如有變更，恕不另行通知。

若您在本手冊中發現任何錯誤或對相關內容有任何意見，請告知我們。

1.5 製造商

SEIKO EPSON CORPORATION

1.6 聯絡資訊

在下列手冊的「供應商」中，記載了詳細的聯絡資訊。

各地區的聯絡資訊有所不同，敬請注意。

「安全手冊 - 聯絡資訊」

亦可從以下網站瀏覽安全手冊。

URL： <https://download.epson.biz/robots/>



1.7 閱讀本手冊之前

本節說明在閱讀本手冊前您應先瞭解的資訊。

關於Epson RC+ 8.0安裝資料夾

Epson RC+ 8.0允許您將安裝資料夾路徑更改為任何位置。本手冊假定Epson RC+ 8.0安裝在C:\EpsonRC80中進行說明。

符號意義

在手冊中的安全注意事項皆以下列符號表示。請確實閱讀各符號的說明。

警告

此符號代表若不正確遵守相關指示，可能會有重傷或死亡的危險。

警告

此符號代表若不正確遵守相關指示，可能會有因觸電導致重傷或死亡的危險。

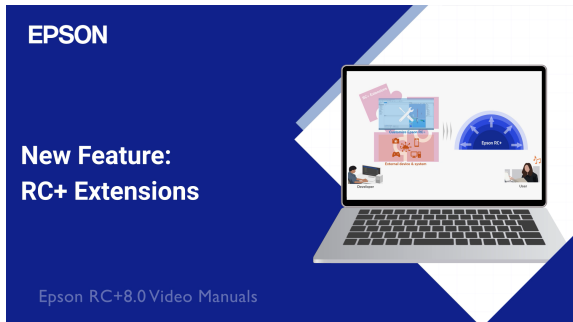
注意

此符號代表若不正確遵守相關指示，可能會有人員受傷或設備及設施受損的危險。

2. 概述

2.1 概述

RC+ Extensions 也有透過影片進行說明。

Title	Link
1. New Feature: RC+ Extensions	

提示

- 觀看影片時，需連接網際網路。
- 會播放音訊。
- 母語顯示請使用 YouTube 的自動翻譯字幕功能。

RC+ Extensions 是一個可依據使用者需求與業務流程，對 Epson RC+ 8.0 進行客製化，並能與外部設備或系統整合，實現彈性擴展的平台。

除了 EPSON 提供的擴展功能（Extension）之外，使用者也可利用 Visual Studio 自行開發專屬的擴展功能（Extension），並安裝於 Epson RC+ 8.0 上使用。

透過此方式，Epson RC+ 8.0 可持續且彈性地進化，所開發的擴展功能（Extension）也能於組織內輕鬆重複利用與分享。

RC+ Extensions 可實現的功能：

- 擴展功能的使用：可安裝並使用 EPSON 提供的擴展功能（Extension），或使用者自行開發的擴展功能（Extension）。
- 擴展功能的開發：可利用 Visual Studio 與 RC+ Extensions SDK 製作專屬的擴展功能（Extension）。

提示

需具備 Epson RC+ 8.0 Ver8.1.3.0 以上版本，以及 Epson RC+ Premium Edition 授權。

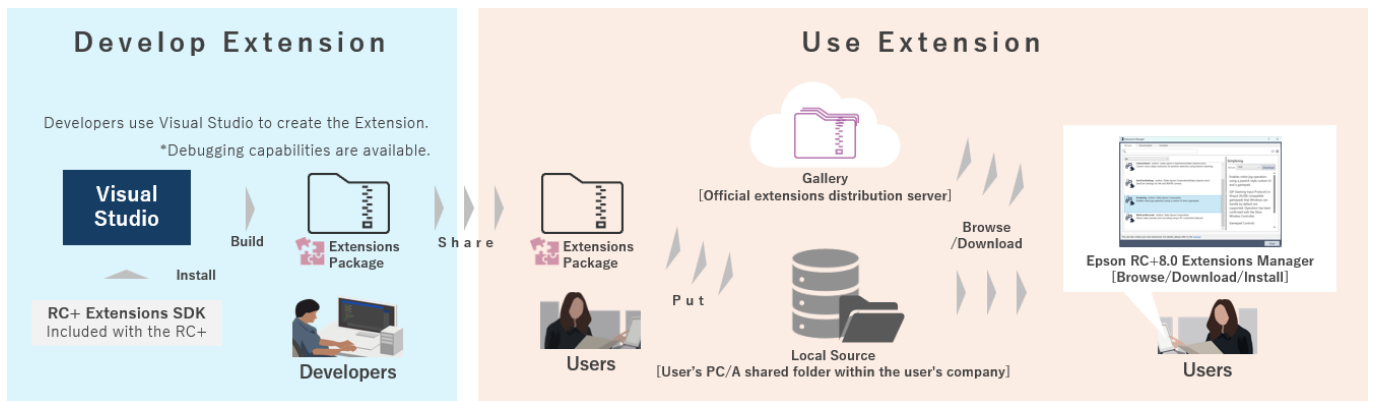
術語說明

術語	說明
RC+ Extensions	用於擴展 Epson RC+ 8.0 的平台。除了 EPSON 提供的擴展功能外，使用者亦可自行開發擴展功能並新增至 Epson RC+ 進行擴展。透過 RC+ Extensions，可依需求彈性客製化 Epson RC+ 的 UI 擴展、外部設備整合、嵌入自定義處理等功能。
Extension	指由 RC+ Extensions 開發，可新增至 Epson RC+ 的擴展功能。以 Visual Studio 與 RC+ Extensions SDK 進行開發，並以 .rcxpkg 格式封裝成套件以供發佈。Extension 安裝於 RC+ 後，可實現 UI／選單新增、SPEL+ 整合、外部設備控制等功能。

術語	說明
擴展功能管理器	Epson RC+ 內建的 Extension 管理介面。可針對 Extension 執行以下操作： *另請參閱 *下載／刪除 *安裝／解除安裝 *啟用／停用 *本機來源資料夾設定 可由 Epson RC+ 主選單的 [擴展]→[擴展功能管理器] 啟動。
Gallery	EPSON 官方提供的 Extension 線上發佈伺服器。於擴展功能管理器的 [瀏覽] 分頁會自動顯示清單，可選擇所需 Extension 進行下載。需連接網際網路。使用者開發的 Extension 無法發佈至 Gallery（使用者發佈請透過本機來源）。
本機來源	指用於存放使用者或組織管理的 Extension 套件（.rcxpkg）的電腦本機資料夾或網路共用資料夾。於擴展功能管理器的「本機來源資料夾設定」註冊資料夾後，該資料夾內的 Extension 會於 [瀏覽] 分頁顯示清單，可如同 Gallery 一樣進行下載與安裝。

下圖說明 RC+ Extensions 的使用方式與開發流程。

- 使用者：可透過 Epson RC+ 的擴展功能管理器，從 EPSON 公開的 Gallery 瀏覽、下載及安裝擴展功能（Extension）。指定本機來源後，也可同樣使用開發者製作的擴展功能（Extension）。
- 開發者：可利用 Visual Studio 與 RC+ Extensions SDK 製作擴展功能（Extension），並封裝成套件以供分享。



在 RC+ Extensions 中，可使用與開發下列類型的擴展功能。各類型的目標功能與開發方式皆有所不同。

- 1. RC+ 自定義：**自定義 Epson RC+ 的 UI 及處理流程
 - 可於 Epson RC+ 新增自定義操作畫面。此外，亦可結合 SDK 提供的標準功能與外部軟硬體，實現新功能。
- 2. PC Vision 自定義視覺物件：**自定義視覺物件
 - 可於 VisionGuide 的 PC Vision 功能中新增自定義視覺物件。因此，可嵌入自定義的影像處理或檢測邏輯。

Epson RC+ 的活用方式有多種。可依據需求選用下列功能。

功能	主要用途
RC+ Extensions	用於新增 Epson RC+ 的畫面與功能，並客製化 UI 及工作流程。亦可將開發的擴展功能封裝後於組織內重複利用與分享。
RC+ Library Builder	用於將 SPEL+ 程式庫化以利重複利用。
RC+ API	用於從外部應用程式透過 Epson RC+ 控制機器人（外部操作 RC+）。

3. Extension 的使用

3.1 概述

若要在 Epson RC+ 中使用 Extension，請於擴展功能管理器中執行下列操作。

- 下載 Extension
- 安裝 Extension
- 啟用 Extension

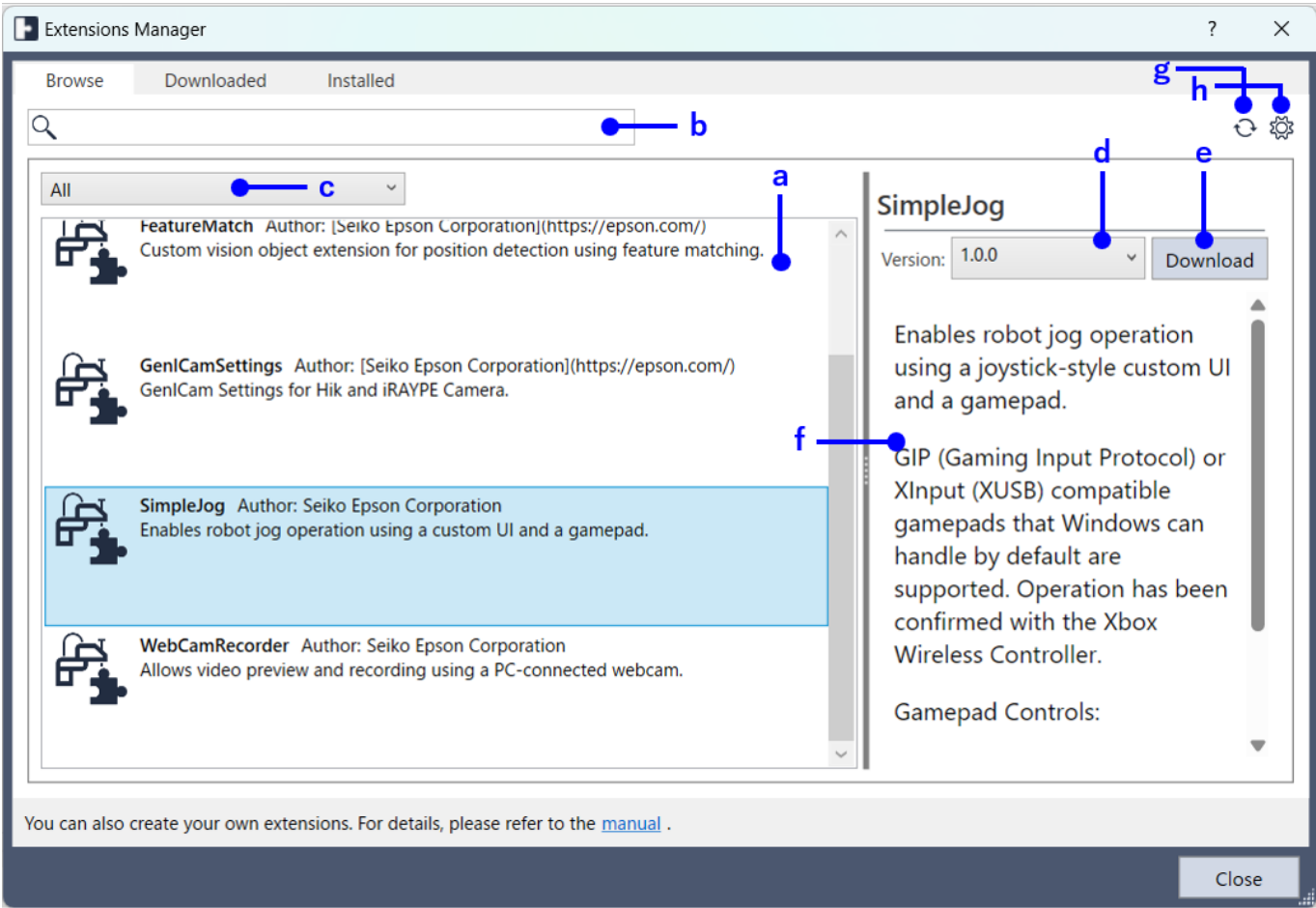
 提示

安裝 Extension 後，預設即為啟用狀態。

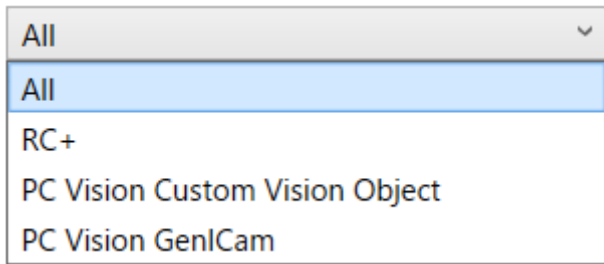
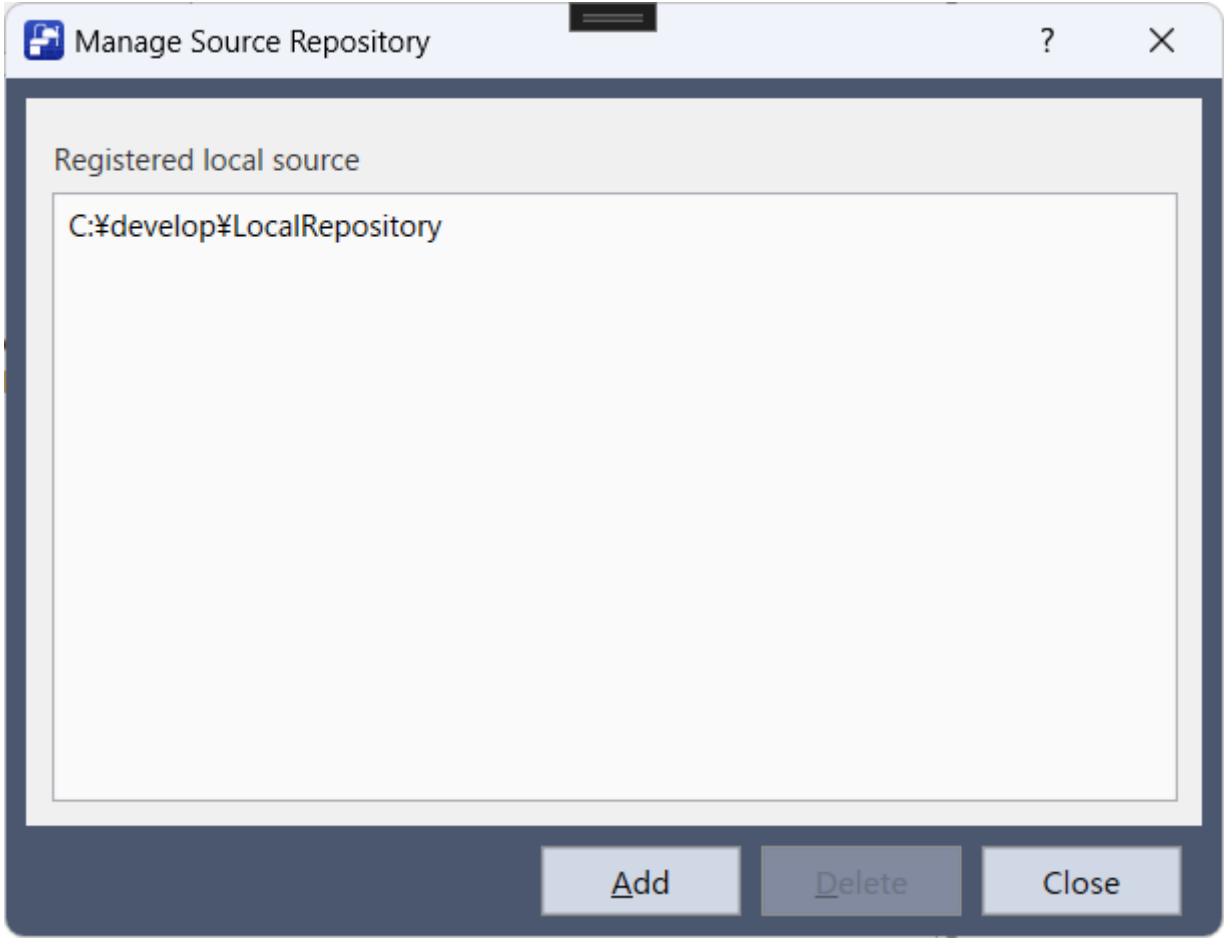
擴展功能管理器可由 [擴展] 選單中選取 [擴展功能管理器] 啟動。

3.2 下載 Extension

Extension 的下載作業，請於擴展功能管理器的 [瀏覽] 分頁中執行。
請選取欲下載的 Extension 及其版本，然後按下 [下載] 按鈕。

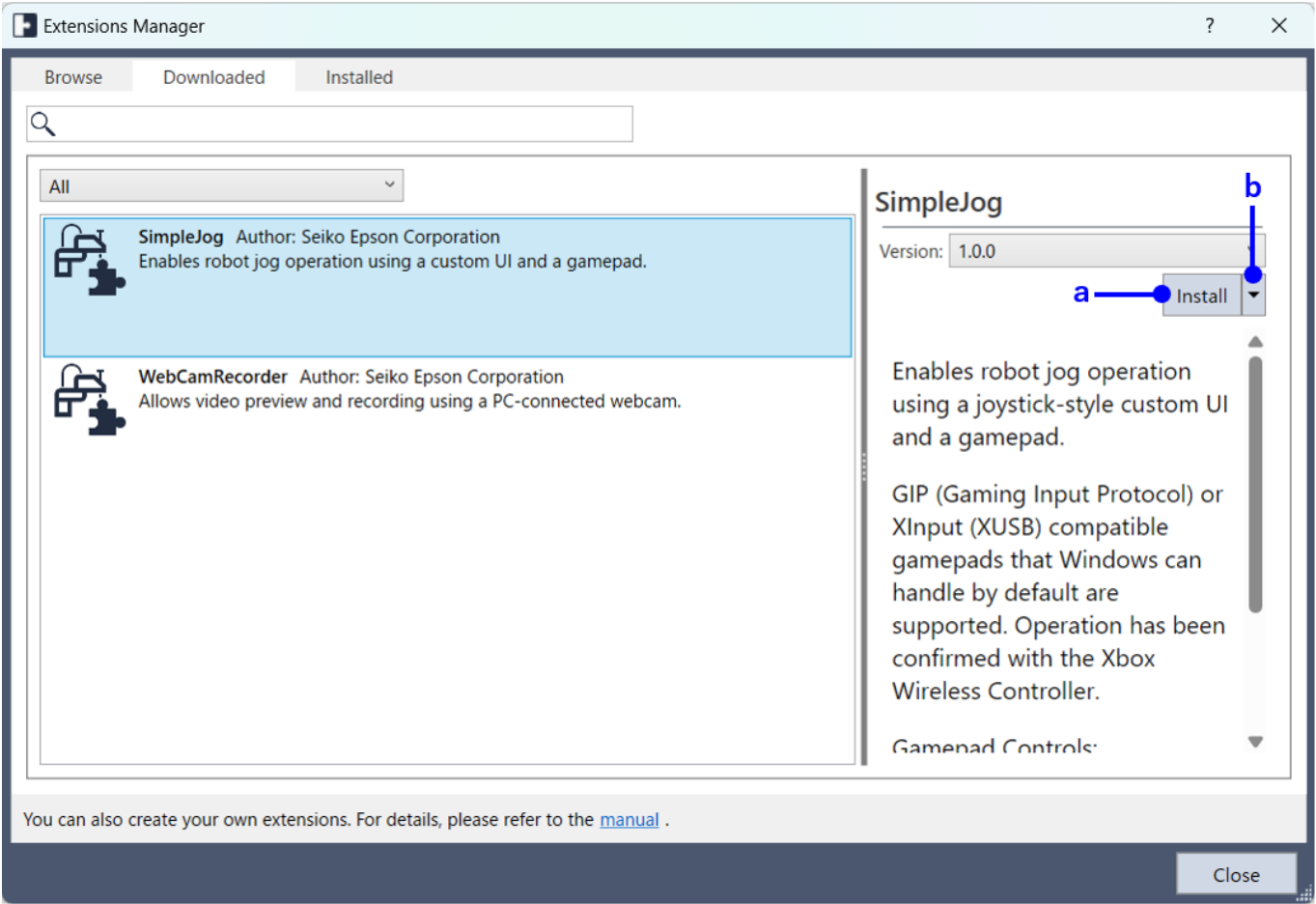


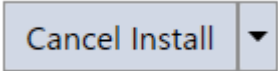
符號	說明
a	將顯示下列 Extension。 - 由 EPSON 一般公開的 Extension - 配置於本機來源的 Extension

符號	說明
b	包含所輸入文字的 Extension 會顯示於「a」。 若未輸入任何內容，則會顯示所有 Extension。
c	請選擇要在「a」中顯示的 Extension 類型。 
d	請選擇 Extension 的版本。
e	將下載於「a」中所選 Extension 的「d」中所選版本。 已下載的 Extension 會顯示於 [已下載] 分頁。
f	會顯示所選 Extension 的說明。
g	更新「a」的顯示內容。
h	請設定本機來源的資料夾。 將 Extension 的套件檔案放置於此處所設定的資料夾後，即會顯示於 [瀏覽] 分頁。 

3.3 安裝 Extension

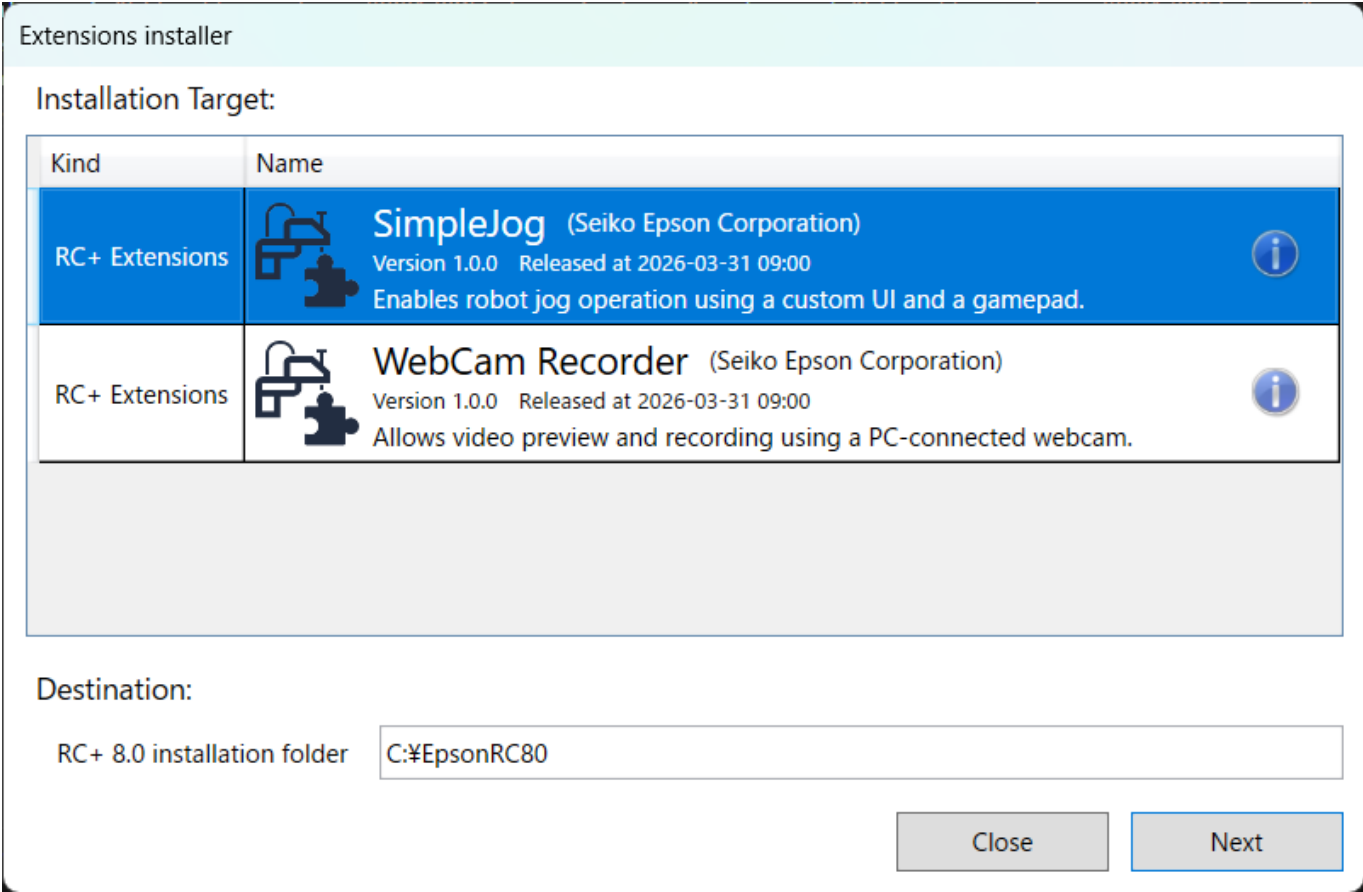
Extension 的安裝需於擴展功能管理器的 [已下載] 分頁中執行。
請選擇欲安裝的 Extension 及其版本，然後按下 [安裝] 按鈕。
結束 Epson RC+ 時，會啟動 [擴展功能安裝程式]，並執行 Extension 的安裝作業。



符號	說明
a	安裝 Extension。 按下按鈕後，按鈕顯示會變更為 [取消安裝]。再次按下按鈕，即可取消安裝。 
b	將顯示以下選單。  - 安裝：與 a 相同。 - 刪除：可刪除已下載的 Extension。

擴展功能安裝程式

結束 Epson RC+ 時，會啟動擴展功能安裝程式。請按下 [下一步] 按鈕以進行安裝。

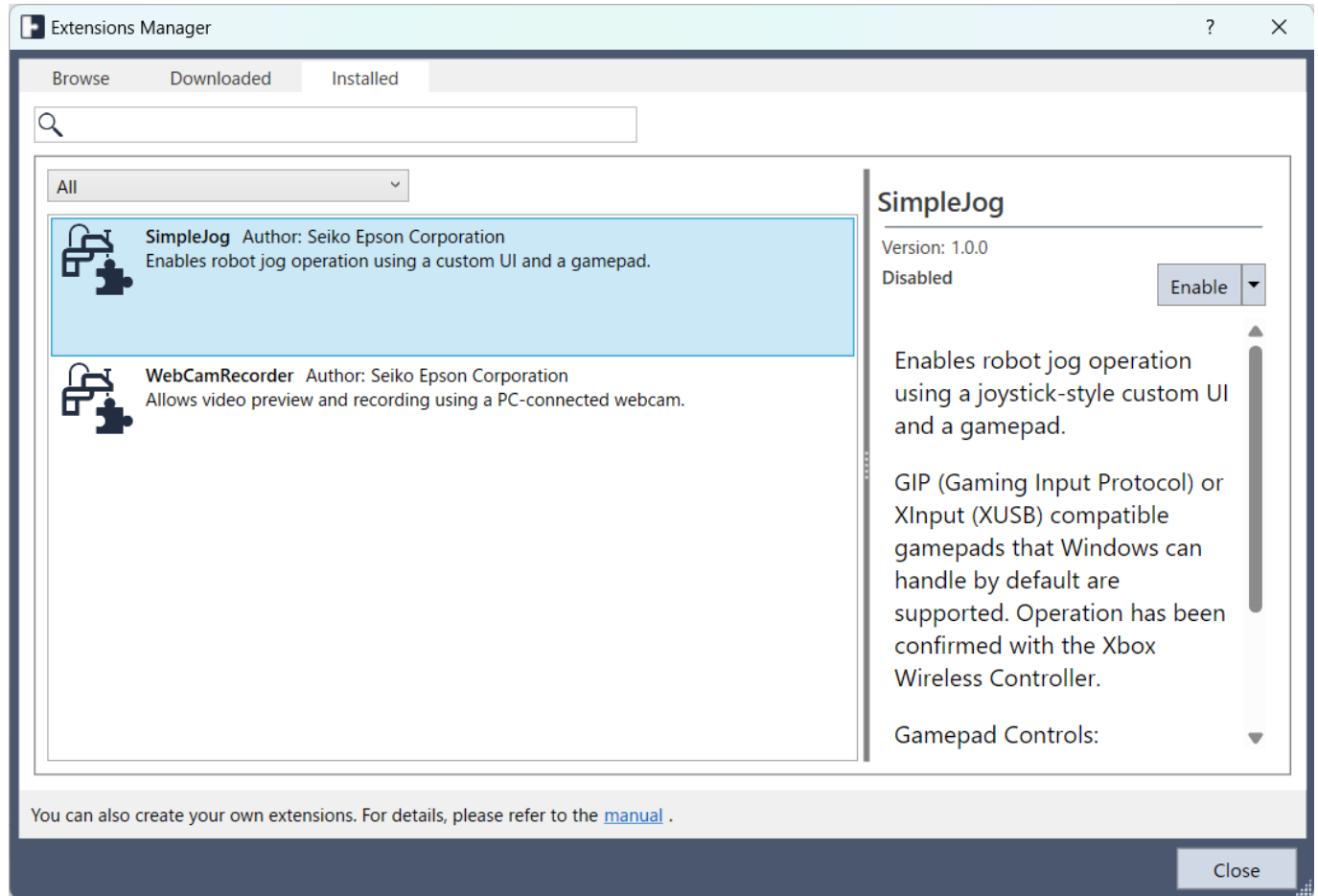


Extension 的啟用與停用

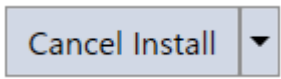
Extension 的啟用需於擴展功能管理器的 [已安裝] 分頁中執行。

請選擇欲啟用的 Extension，然後按下 [啟用] 按鈕。

結束 Epson RC+ 時，Extension 會被啟用，並可於下次啟動時使用。



按下 [啟用] 按鈕後，按鈕顯示會變更為 [取消啟用]。再次按下按鈕，即可取消啟用。



Extension 的停用亦採用相同步驟。

若需暫時停用所有 Extension，請於啟動 Epson RC+ 時指定以下命令列選項。

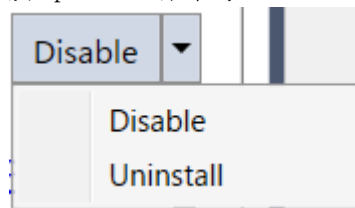
```
erc80.EXE /DISABLE_EXT
```

3.4 Extension 的解除安裝

Extension 的解除安裝需於擴展功能管理器的 [已安裝] 分頁中執行。

請選取欲解除安裝的 Extension，點選 [啟用]/[停用] 按鈕旁的 [▼]，然後按一下解除安裝選單。

於 Epson RC+ 結束時，Extension 會被解除安裝。



※若執行上述步驟後仍無法解除安裝 Extension，請先將 Epson RC+ 更新為最新版本之後，再重新執行一次 Extension 的解除安裝操作。

4. Extension 的種類

4.1 概述

在 RC+ Extensions 中，可以開發以下擴展功能。各類型的目標功能與開發方式皆有所不同。

Extension 的種類	主要使用語言 / 技術	概述
RC+ 自定義： 可自定義 Epson RC+ 的 UI 及處理流程	C#, WPF, XAML	可於 Epson RC+ 新增自定義操作畫面。此外，透過 SDK 所提供的標準功能，並與外部軟體或硬體整合，可實現新的功能。
PC Vision 自定義視覺物件： 可自定義視覺物件	C++, (影像處理知識)	可於 VisionGuide 的 PC Vision 功能中新增自定義視覺物件。因此，可嵌入自定義的影像處理或檢測邏輯。

提示

需具備 Epson RC+ 8.0 Ver8.1.3.0 以上版本，以及 Epson RC+ Premium Edition 授權。

開發流程如下：

1. 開發環境準備：請參閱 [\[安裝\]](#)
2. RC+ Extensions 專案建立：請參閱 [\[開始使用\]](#)
3. 實作、除錯：請參閱 GitHub 上提供的完整原始碼。
<https://github.com/Epson-Robots/rcplus-extensions>
4. 封裝、共享：請參閱 [\[Extension 的發佈\]](#)

下節將介紹各類型的開發方式。

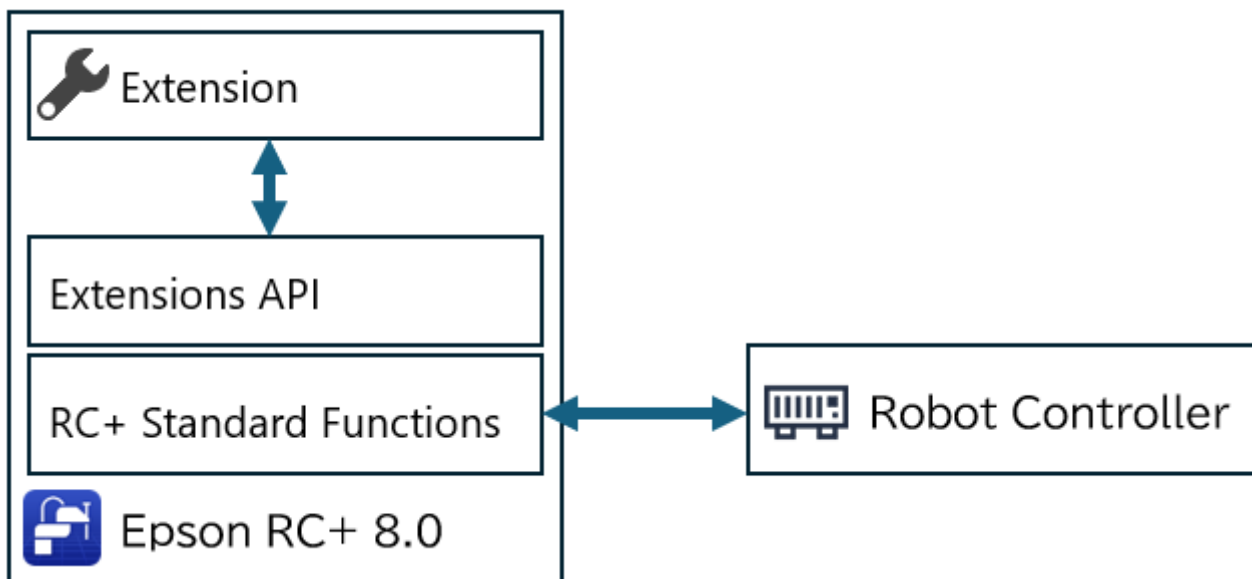
5. RC+ 自定義功能的實作

5.1 概述

RC+ 自定義功能可讓使用者針對 Epson RC+ 的標準功能進行擴展，以滿足特定需求。可實現以下擴展功能。

- **UI 擴展**：可在 Epson RC+ 的畫面中新增自定義視窗。
- **標準功能呼叫**：可呼叫 Epson RC+ 的標準功能，例如點位編輯、控制器連線等。
- **SPEL+ 整合**：可透過 Declare 敘述，從 SPEL+ 程式呼叫 Extension 功能。
- **Web 內容整合**：利用 C# WPF 的 WebView2，在 Epson RC+ 上顯示網頁。此外，Extension 與 Web 之間可進行雙向訊息通訊。

系統結構如下所示。



Extension 於 RC+ 主程式行程內（In-Process）執行。Extension 透過 RC+ Extensions SDK 的 Extensions API 公開介面存取 RC+ 標準功能，因此設計上無法直接存取 RC+ 內部。如此可確保 RC+ 標準功能的一致性，同時安全地實現 UI 擴展與功能整合。

※ GitHub 上已提供範例程式與 API 的說明。關於程式的實作方式等，亦請參考此資源。

<https://github.com/Epson-Robots/rcplus-extensions>

5.2 安裝

開始開發前，需進行以下準備作業。

1. 請安裝 Epson RC+ 8.0。
 - 版本：8.1.3.0 以上
 - 必要授權：Epson RC+ Premium Edition
 - 安裝路徑可自定義。本手冊假設安裝於 C:\EpsonRC80。
2. 請安裝 Visual Studio。
 - 版本：Visual Studio 2022 17.14 以上（包含 Enterprise、Professional 或 Community 版本）
 - 必要元件：.NET 桌面開發、C++ 桌面開發
3. 請安裝 .NET 8 SDK。
 - 版本：8.0.307 以上

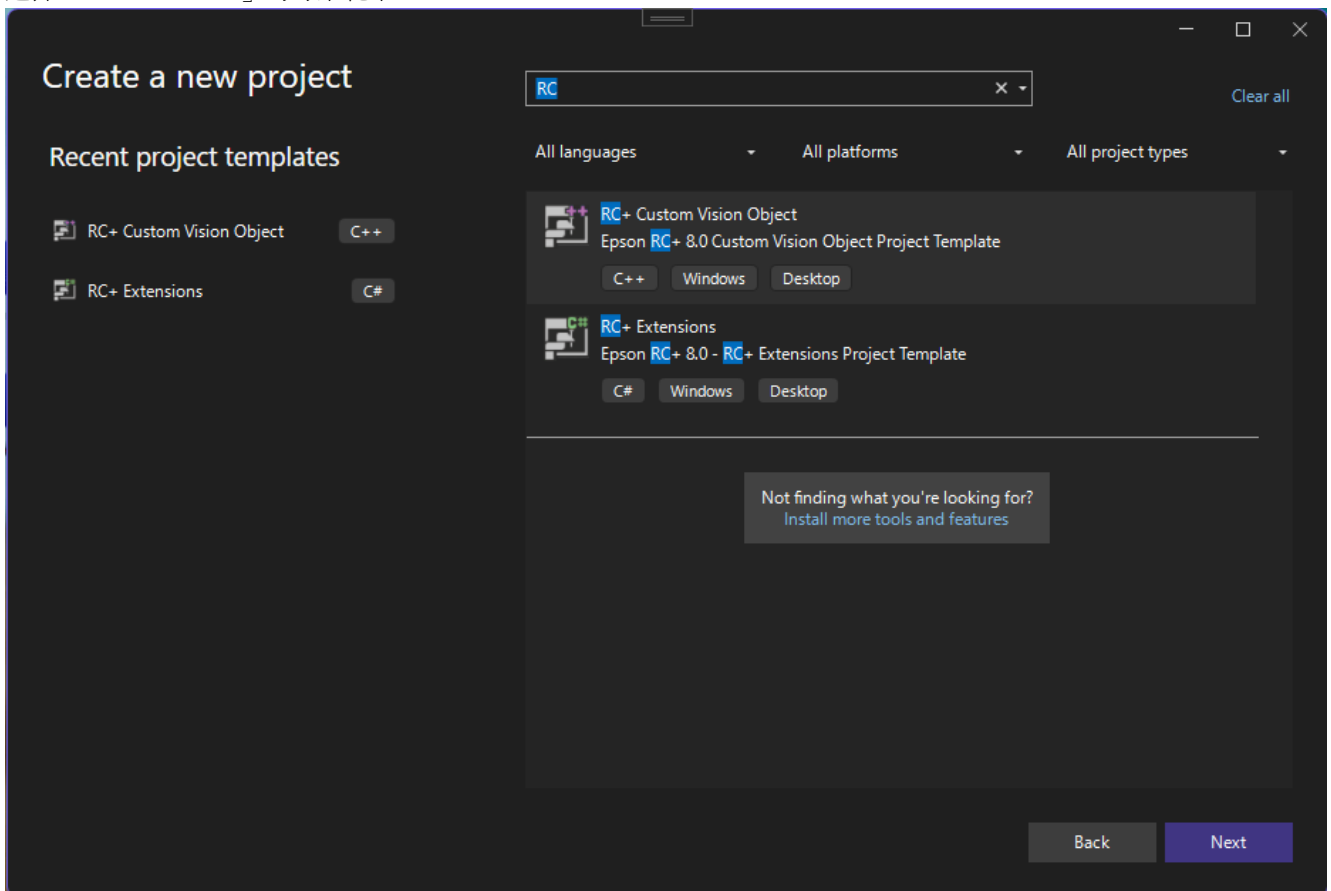
4. 請將擴展功能（VSIX）安裝至 Visual Studio。

- 請按兩下 C:\EpsonRC80\Extensions\SDKs\ExtensionsGenerator.vsix。VSIX 安裝程式將會啟動，請依照畫面指示完成安裝。
※使用 VSIX 需要 Premium Edition。啟動以 Premium Edition 認證的 RC+ 後，VSIX 會自動展開至上述路徑。
- 此擴展功能會在 Visual Studio 中註冊 RC+ Extensions 專用的「專案範本」及「檔案範本」。後續步驟將會使用這些範本來建立專案。

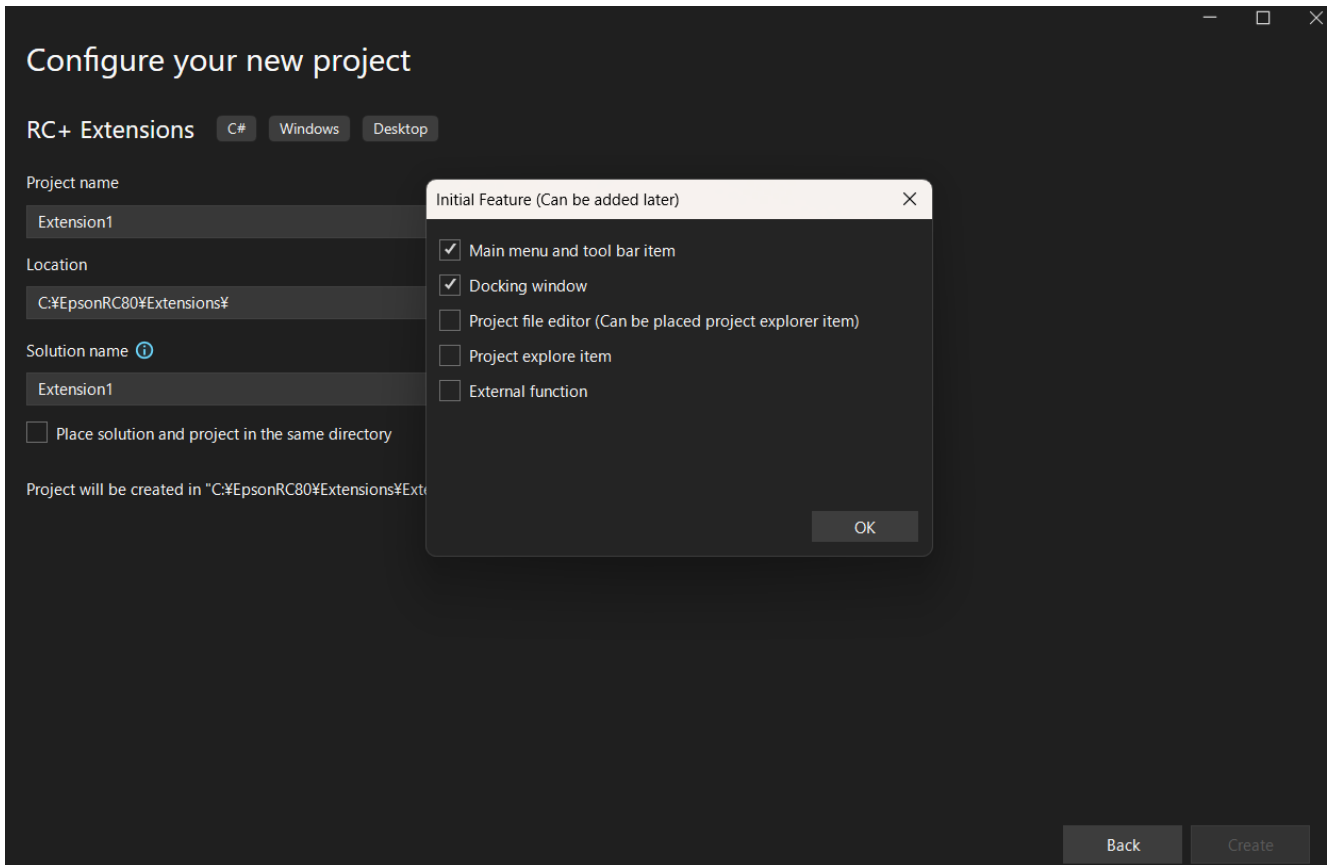
5.3 開始使用

請依照以下步驟建立 RC+ Extensions 專案。

1. 於 Visual Studio 中選擇「建立新專案」。
2. 選擇「RC+ Extensions」的專案範本。



3. 在「Initial Feature」視窗中選擇要使用的功能。



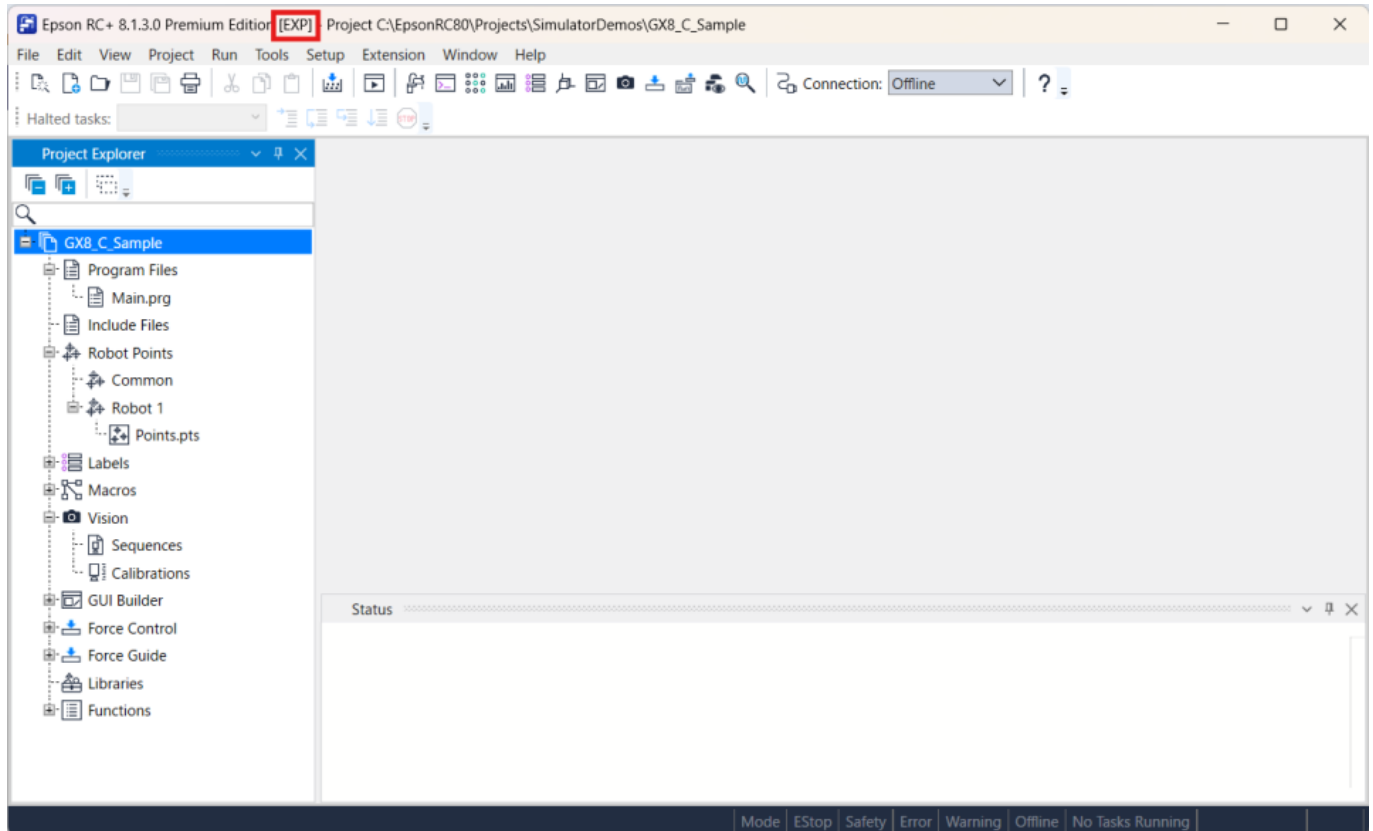
- **Main menu and tool bar item**：可於 Epson RC+ 主視窗的功能表及工具列新增 Extension 專用的選單。
- **Docking window**：可於 Epson RC+ 主視窗新增 Extension 專用的停駐視窗。
- **Project file editor**：可於專案管理下新增 Extension 專用的自定義檔案。
- **Project explore item**：可於專案總管中新增 Extension 專用的項目。
- **External function**：可自 SPEL+ 透過 Declare 陳述式呼叫 Extension 專用的處理程序。

提示

日後如有需要，可另外新增所需功能的檔案範本。於方案總管上按右鍵，選擇「新增」-「新項目」，並以「RC+」為關鍵字搜尋後新增。

建立完成的專案可直接進行建置，並於安裝至 RC+ 後進行偵錯。

於選單中選擇「偵錯」-「開始偵錯」，即可建置 Extension，完成後會啟動 Epson RC+。Epson RC+ 啟動後，請確認所選功能（視窗或選單）已正確顯示。主視窗標題會顯示 [EXP]，以表示 Extension 處於偵錯狀態。



RC+ 啟動時，如已有其他 RC+ 執行個體，請先將其結束。

GitHub 上提供有 API 參考文件及範例程式。請參考程式實作方法的詳細說明。請參閱以下 URL。

<https://github.com/Epson-Robots/rcplus-extensions>

5.4 Extension 的發佈

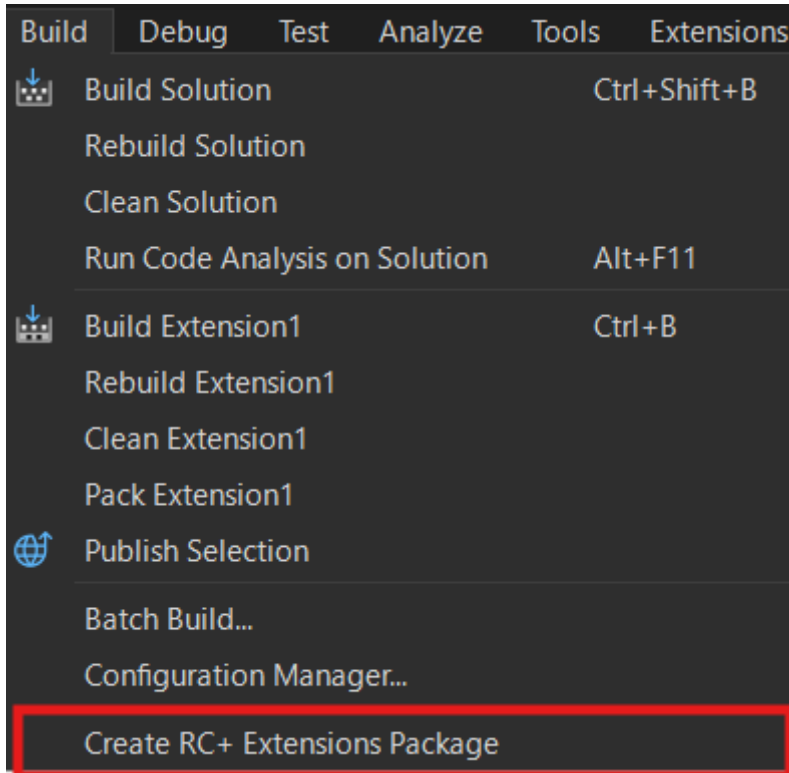
若要在其他環境中使用所開發的 Extension，需將其封裝成套件。在 RC+ Extensions 中，可透過 Visual Studio 的建置選單輕鬆建立套件。

請依下列步驟進行套件封裝。

1. 於 Visual Studio 開啟專案，選擇選單 [建置] - [建置方案]。
 - 建置順利完成後，專案根目錄下的 `bin` 資料夾會產生 DLL 等產出檔案。

2. 選擇選單 [建置] - [Create RC+ Extensions Package]。

- 如以下圖所示，Visual Studio 的建置選單中已新增專用項目。



- 執行後，專案根目錄下的 `bin\Release` 或 `bin\Debug` 資料夾會產生 `.rcxpkg` 檔案。
- `.rcxpkg` 檔案為可於 Epson RC+ 擴展功能管理器安裝的格式。

所建立的 `.rcxpkg` 可依下列步驟安裝至 Epson RC+。

1. 請將 `.rcxpkg` 檔案放置於電腦內任意資料夾或組織內的共用資料夾。
2. 啟動 Epson RC+，於 [擴展] 選單中選擇 [擴展功能管理器]。
3. 於畫面右上方的本機來源設定按鈕，將剛才放置的資料夾路徑新增為本機來源。如此一來，於擴展功能管理器的瀏覽分頁中即可顯示所建立的 Extension，並可進行下載與安裝。
 - 詳細內容請參閱 [Extension 的使用]。

<補充說明>

進行 Extension 的衍生開發時，需要修改作為 Extension 識別碼使用的 ID(包含 GUID)，以避免與其他 Extension 重複。Extension 的 ID 包含於以下位置，且皆需要替換為新的 ID：

- PackageItems/manifest.rcxmanifest 內的 id
(例) id: SimpleJog_d2525b11-6852-4164-bba0-1a0469afa2d1
- 在 Main.CommonId 中定義的 ID 字串
專案檔案(.csproj)的

若這些 ID 與其他 Extension 相同，可能會導致無法顯示於擴展功能管理器、或發生 Extension 誤識別等問題。用於 ID 的 GUID，可透過 Visual Studio 的 [工具] - [建立 GUID] 產生。

5.5 試著建立看看吧

在 Extension (RC+ 自定義) 中，提供了可將自定義功能整合至 Epson RC+ 的機制 (擴展點)，例如新增選單項目、管理專案檔案、顯示停駐視窗等。

這些功能是透過將 Extensions API 所提供的擴展點介面，以 .NET 內建的擴展框架 Managed Extensibility Framework

(MEF) 的 Export 形式實作來運作。

建立 Extension 專案時，可以將欲使用的擴展點作為 Initial Feature 進行選擇，專案建立後亦可作為新項目新增。

此處將說明擴展點實作方法的一例，並說明從建立 RC+ Extensions 專案到在 Epson RC+ 中使用為止的流程。

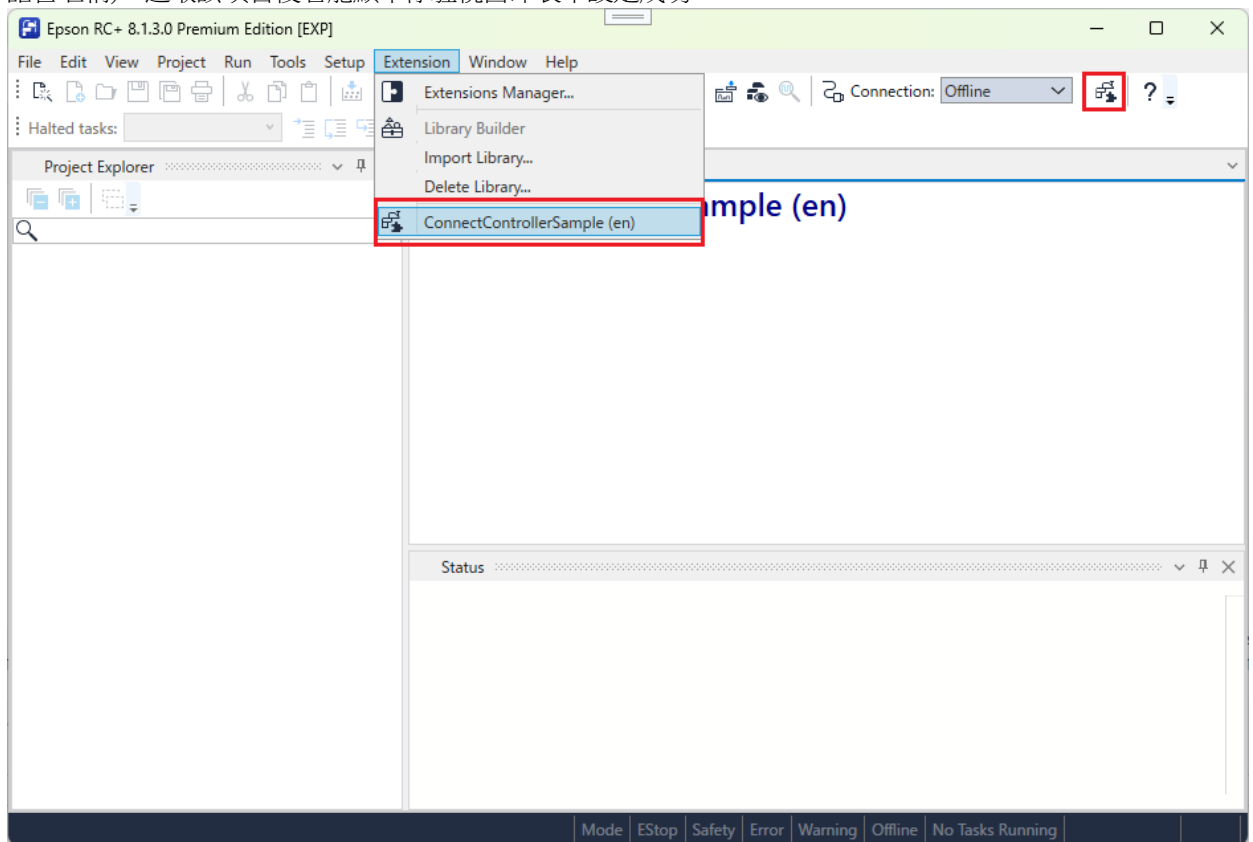
以下步驟中，將在 Extension 的停駐視窗配置按鈕，並建立使用 Extension API 的介面進行控制器連線/切斷的範例。

1. 請依照「開始使用」的步驟，建立 Extension 的新專案。

- 名稱設為 ConnectControllerSample。
- 初始功能請勾選 **Main menu and tool bar item**、以及 **Docking window**。
- 於 ARM64 版 Windows 上，請將組態設為 x64。

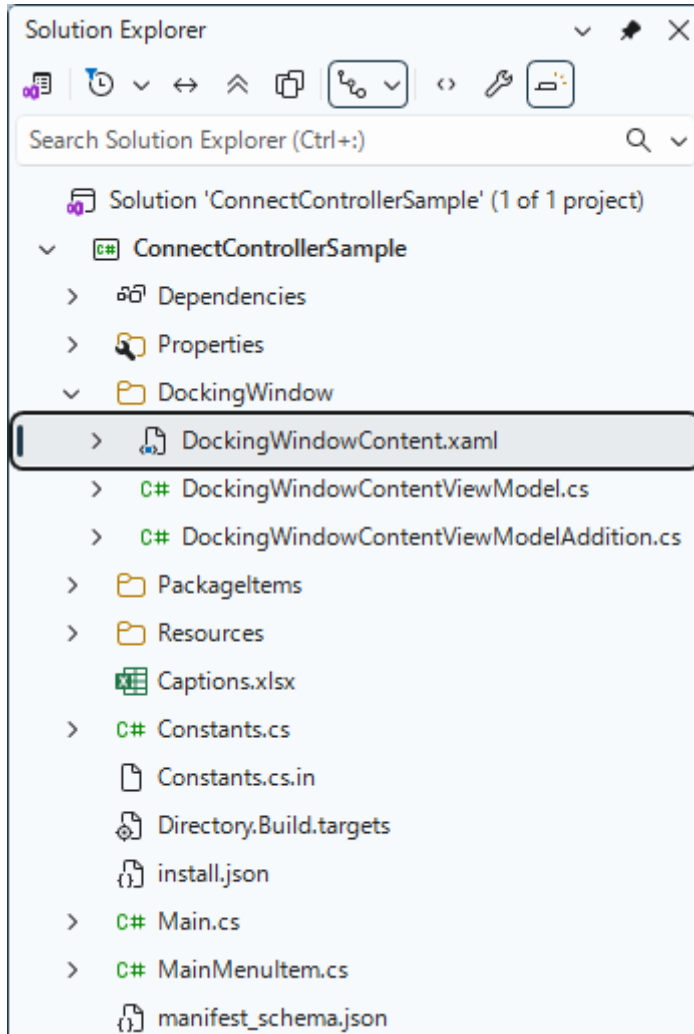
2. 請於 Visual Studio 上進行建置與除錯，確認功能可正常運作。

- 於 Epson RC+ 主選單的擴展標籤下的選單項目中會新增 ConnectControllerSample (xx) (xx)(xx 為顯示語言名稱)，選取該項目後若能顯示停駐視窗即表示設定成功。



3. 範本確認完成後，請結束 Epson RC+。

4. 於 Visual Studio 上編輯 DockingWindow 資料夾下的 DockingWindowContent.xaml 檔案。



- 在自動產生並配置的 TextBlock 下方 Grid 內，新增控制器連線的 Button、以及控制器切斷的 Button。
- 請留意 TextBlock 的 Text。其綁定至 Captions[WindowHeader].Value。於 Extension 中，欲依 Epson RC+ 顯示語言進行在地化的字串，請記述於 Captions.xlsx。詳細內容將於後述說明，透過使用 Captions.xlsx 的 symbol 欄定義名稱(本例為 WindowHeader)，並同樣進行繫結，即可依 Epson RC+ 的顯示語言進行在地化。

```
<UserControl
x:Class="ConnectControllerSample.DockingWindow.DockingWindowContent"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:ConnectControllerSample.DockingWindow"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800">

    <UserControl.DataContext>
        <local:DockingWindowContentViewModel />
    </UserControl.DataContext>

    <DockPanel
        Background="White"
        LastChildFill="True">
```

```
<TextBlock
    Margin="3,0,0,0"
    DockPanel.Dock="Top"
    FontSize="24"
    FontWeight="Bold"
    Foreground="DarkBlue"
    Text="{Binding Captions[WindowHeader].Value}" />

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>

    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto" />
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>

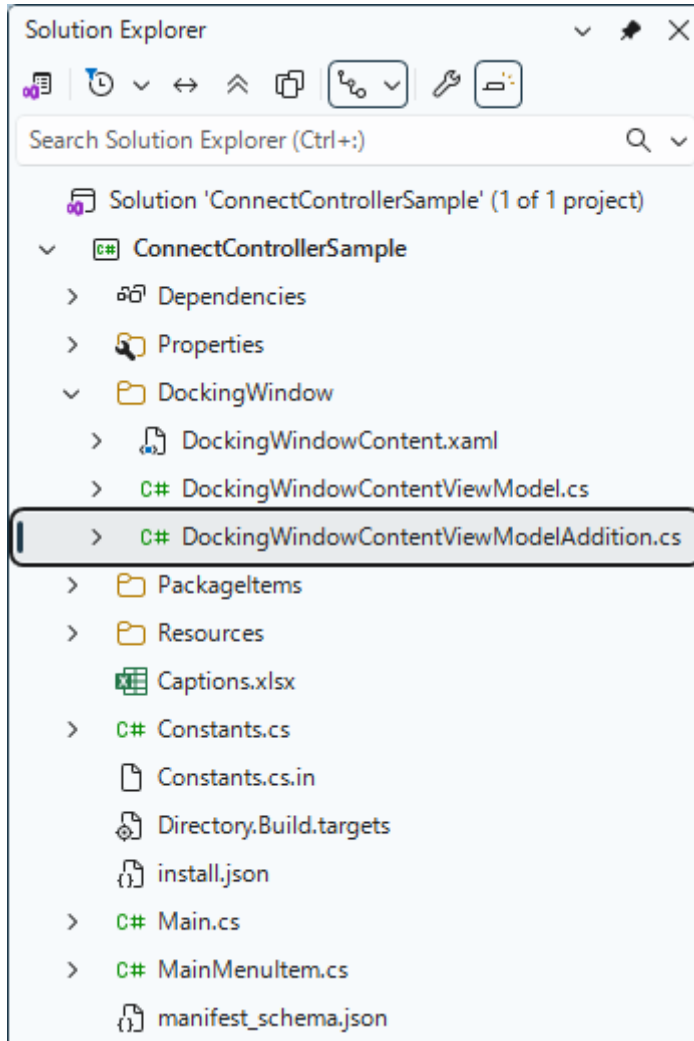
    <Button
        x:Name="ConnectButton"
        Grid.Row="0"
        Grid.Column="0"
        Margin="4"
        HorizontalAlignment="Left"
        VerticalAlignment="Top"
        Command="{Binding ConnectCommand}"
        Content="Connect Controller" />

    <Button
        x:Name="DisconnectButton"
        Grid.Row="0"
        Grid.Column="1"
        Margin="4"
        HorizontalAlignment="Left"
        VerticalAlignment="Top"
        Command="{Binding DisconnectCommand}"
        Content="Disconnect Controller" />
</Grid>

</DockPanel>

</UserControl>
```

5. 請編輯 DockingWindow 資料夾下的 DockingWindowContentViewModelAddition.cs 檔案。



- 在 .xaml 中，新增繫結至控制器連線與切斷 Button 的 ConnectCommand、以及 DisconnectCommand 命令。
- 透過參照 IRCXControllerConnectionAPI(控制器連線 API)的 IsOnline 屬性，即可確認與機器人控制器的連線狀態。IsOnline 於連線建立時為 true，斷線時為 false，其他（如連線建立中等中間狀態）為 null。
- 與機器人控制器的連線，會呼叫 IRCXControllerConnectionAPI 的 ConnectControllerAsync 方法。與機器人控制器的切斷，會呼叫控制器連線 API 的 DisconnectControllerAsync 方法。
- 訊息方塊的顯示，會呼叫 IRCXWindowAPI(視窗 API)的 ShowMessageBox 方法。
- 在建構函式中，透過 Main 類別的 GetAPI 方法，從 MEF 的容器取得 IRCXControllerConnectionAPI、以及 IRCXWindowAPI 的實例。

(前略)

```
namespace ConnectControllerSample.DockingWindow
{
    /// <summary>
    /// Extension : Docking Window (Specific Part)
    /// </summary>
    internal partial class DockingWindowContentViewModel
    {
        /// <summary>
        /// Connect controller command
        /// </summary>
        public AsyncReactiveCommand ConnectCommand { get; } = new
        AsyncReactiveCommand();
    }
}
```

```

    /// <summary>
    /// Disconnect controller command
    /// </summary>
    public AsyncReactiveCommand DisconnectCommand { get; } = new
AsyncReactiveCommand();

    /// <summary>
    /// Window-related API
    /// </summary>
    private IRCXWindowAPI _windowAPI;

    /// <summary>
    /// Controller Connection API
    /// </summary>
    private IRCXControllerConnectionAPI _connectionAPI;

    /// <summary>
    /// Constructor
    /// </summary>
    public DockingWindowContentViewModel()
    {
        _windowAPI = Main.GetAPI<IRCXWindowAPI>();
        _connectionAPI = Main.GetAPI<IRCXControllerConnectionAPI>();

        ConnectCommand.Subscribe(OnConnect);
        DisconnectCommand.Subscribe(OnDisconect);
    }

    private async Task OnConnect()
    {
        if (_connectionAPI.IsOnline == false)
        {
            var result = await _connectionAPI.ConnectControllerAsync();

            _windowAPI.ShowMessageBox(
                new RCXCaption(Main.CommonId, Caption.ExtensionName),
                new RCXCaption(result ? "Controller connection succeed." :
"Controller connection failed."),
                IRCXWindowAPI.ButtonType.OK,
                IRCXWindowAPI.IconType.Information);
        }
        else
        {
            _windowAPI.ShowMessageBox(
                new RCXCaption(Main.CommonId, Caption.ExtensionName),
                new RCXCaption("The controller cannot be connected."),
                IRCXWindowAPI.ButtonType.OK,
                IRCXWindowAPI.IconType.Warning);
        }
    }

    private async Task OnDisconect()
    {

```

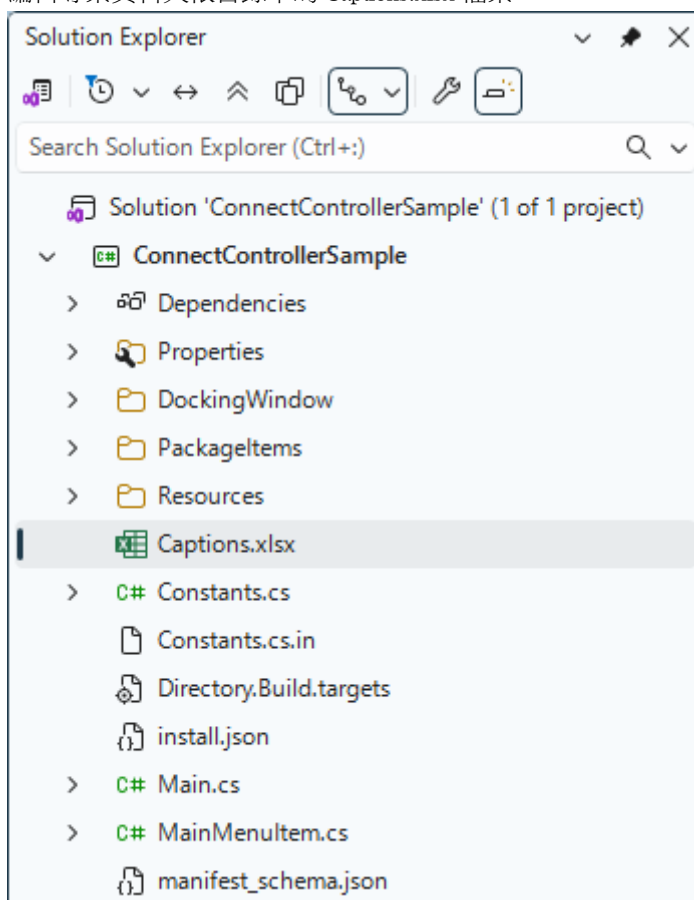
```

        if (_connectionAPI.IsOnline == true)
        {
            await _connectionAPI.DisconnectControllerAsync();
        }
        else
        {
            _windowAPI.ShowMessageBox(
                new RCXCaption(Main.CommonId, Caption.ExtensionName),
                new RCXCaption("The controller cannot be disconnected."),
                IRCXWindowAPI.ButtonType.OK,
                IRCXWindowAPI.IconType.Warning);
        }
    }

    /// <inheritdoc />
    public Task WindowCreated()
    {
        return Task.CompletedTask;
    }
}

```

6. 編輯專案資料夾根目錄下的 Captions.xlsx 檔案。

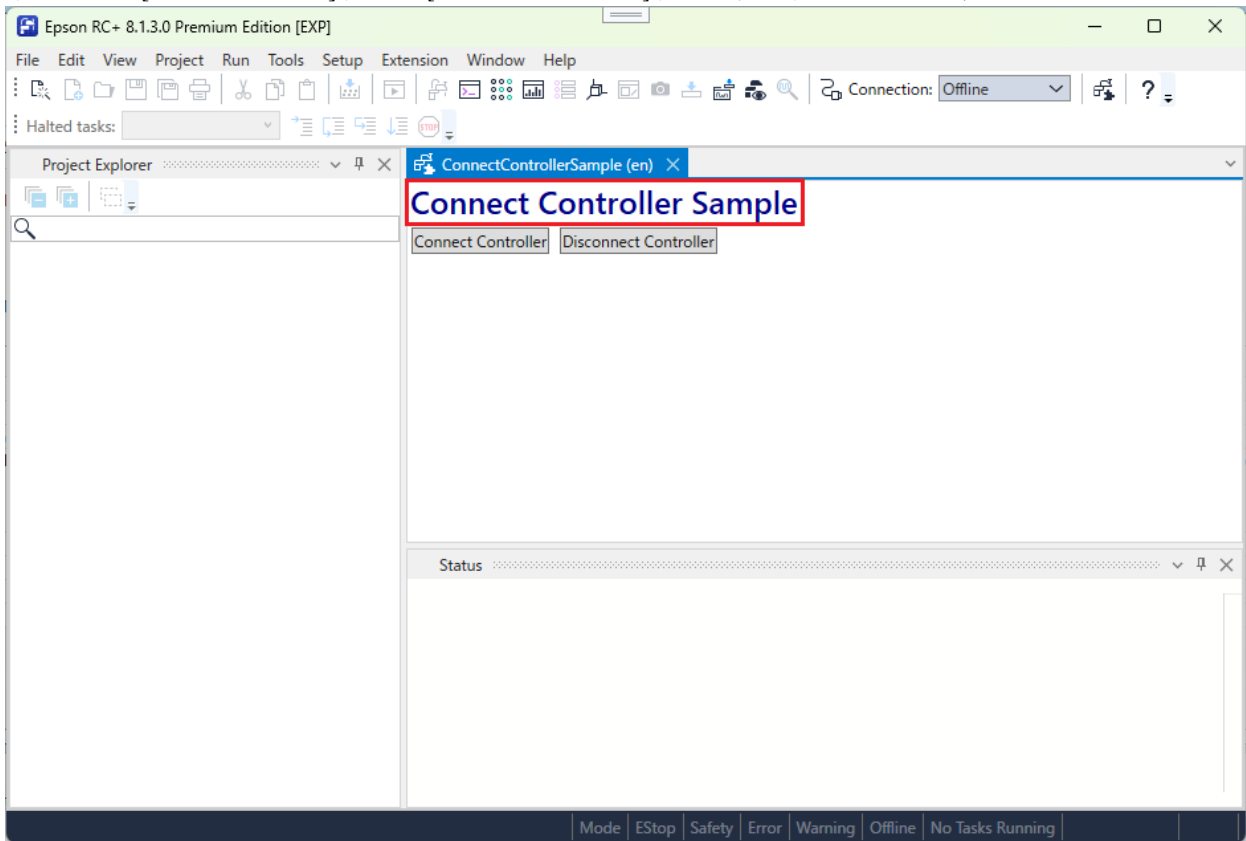


- 修改 symbol 欄為 WindowHeader 的行中各語言的標題。

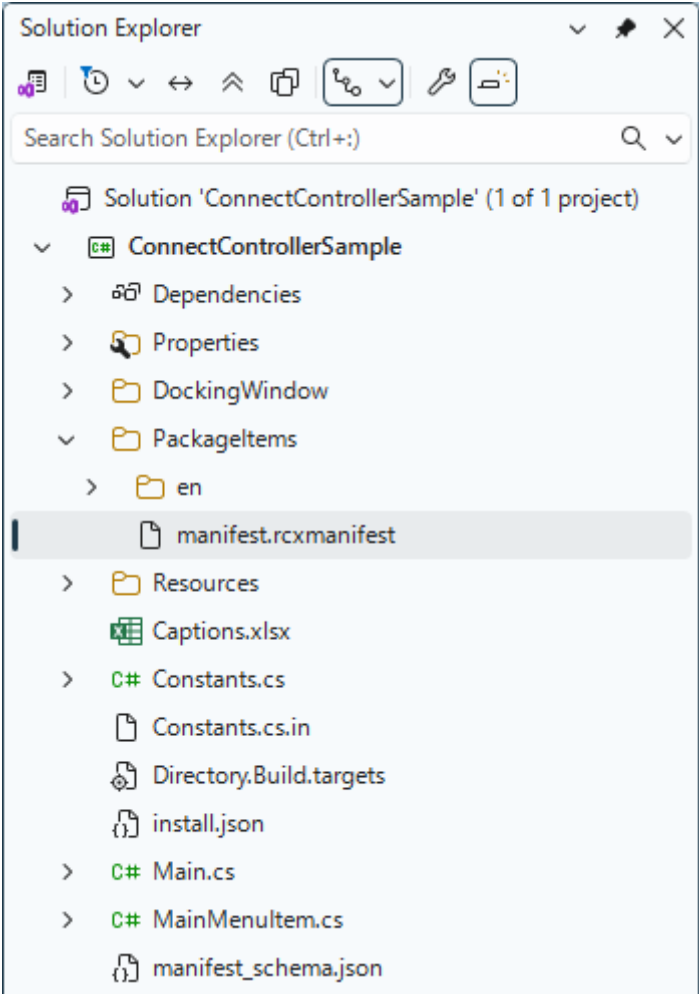
	A	B	C	D	E	F	G	H	I	J	K
1	ID	description	symbol	DEL	English English_en	Japanese Japanese_ja	German German_de	French French_fr	Spanish Spanish_es	SimplifiedChinese Chinese(Simplified)_zh-Hans	TraditionalChinese Chinese(Traditional)_zh-Hant
2											
3											
4	0	Extension Name	ExtensionName		ConnectControllerSample (en)	ConnectControllerSample (ja)	ConnectControllerSample (de)	ConnectControllerSample (fr)	ConnectControllerSample (es)	ConnectControllerSample (zh-Hans)	ConnectControllerSample (zh-Hant)
5	1	Main Menu & ToolBar Item	MainMenu		ConnectControllerSample (en)	ConnectControllerSample (ja)	ConnectControllerSample (de)	ConnectControllerSample (fr)	ConnectControllerSample (es)	ConnectControllerSample (zh-Hans)	ConnectControllerSample (zh-Hant)
6											
7											
8	100	Project Explorer Category	PECategory		ConnectControllerSample (en)	ConnectControllerSample (ja)	ConnectControllerSample (de)	ConnectControllerSample (fr)	ConnectControllerSample (es)	ConnectControllerSample (zh-Hans)	ConnectControllerSample (zh-Hant)
9	110	Project Explorer Item	PEItem		ConnectControllerSample (en) Item	ConnectControllerSample (ja) Item	ConnectControllerSample (de) Item	ConnectControllerSample (fr) Item	ConnectControllerSample (es) Item	ConnectControllerSample (zh-Hans) Item	ConnectControllerSample (zh-Hant) Item
10	120	Project Explorer Item Menu	PEItemMenu		ConnectControllerSample (en) Item Command	ConnectControllerSample (ja) Item Command	ConnectControllerSample (de) Item Command	ConnectControllerSample (fr) Item Command	ConnectControllerSample (es) Item Command	ConnectControllerSample (zh-Hans) Item Command	ConnectControllerSample (zh-Hant) Item Command
11											
12	200	Project File Category	FileCategory		ConnectControllerSample (en)	ConnectControllerSample (ja)	ConnectControllerSample (de)	ConnectControllerSample (fr)	ConnectControllerSample (es)	ConnectControllerSample (zh-Hans)	ConnectControllerSample (zh-Hant)
13	201	Project File Type Name	FileTypeName		ConnectControllerSample (en) File	ConnectControllerSample (ja) File	ConnectControllerSample (de) File	ConnectControllerSample (fr) File	ConnectControllerSample (es) File	ConnectControllerSample (zh-Hans) File	ConnectControllerSample (zh-Hant) File
14											
15	300	Project File Editor Header (UI Sample)	EditorHeader		ConnectControllerSample (en)	ConnectControllerSample (ja)	ConnectControllerSample (de)	ConnectControllerSample (fr)	ConnectControllerSample (es)	ConnectControllerSample (zh-Hans)	ConnectControllerSample (zh-Hant)
16											
17	400	Docking Window Title	WindowTitle		ConnectControllerSample (en)	ConnectControllerSample (ja)	ConnectControllerSample (de)	ConnectControllerSample (fr)	ConnectControllerSample (es)	ConnectControllerSample (zh-Hans)	ConnectControllerSample (zh-Hant)
18	401	Docking Window Header (UI Sample)	WindowHeader		Connect Controller Sample コントローラ接続サンプル	Connect Controller Beispiel	Connect Controller Sample Beispiel	Exemple de contrôleur de connexion	Ejemplo de controlador de conexión	连接控制器示例	连接控制器範例

7. 請重新建置與除錯。

- 在啟動的 Epson RC+ 中，點擊已開發 Extension 的選單或工具列圖示，顯示所建立的 DockingWindow。
- 確認於畫面上方顯示在對應目前 Epson RC+ 語言的 Captions.xlsx 中所設置的標題。
- 確認可透過 [Connect Controller] 按鈕與 [Disconnect Controller] 按鈕進行控制器連線與切斷。



8. 在 Visual Studio 上編輯 PackageItems 資料夾中的 manifest.rcxmanifest。



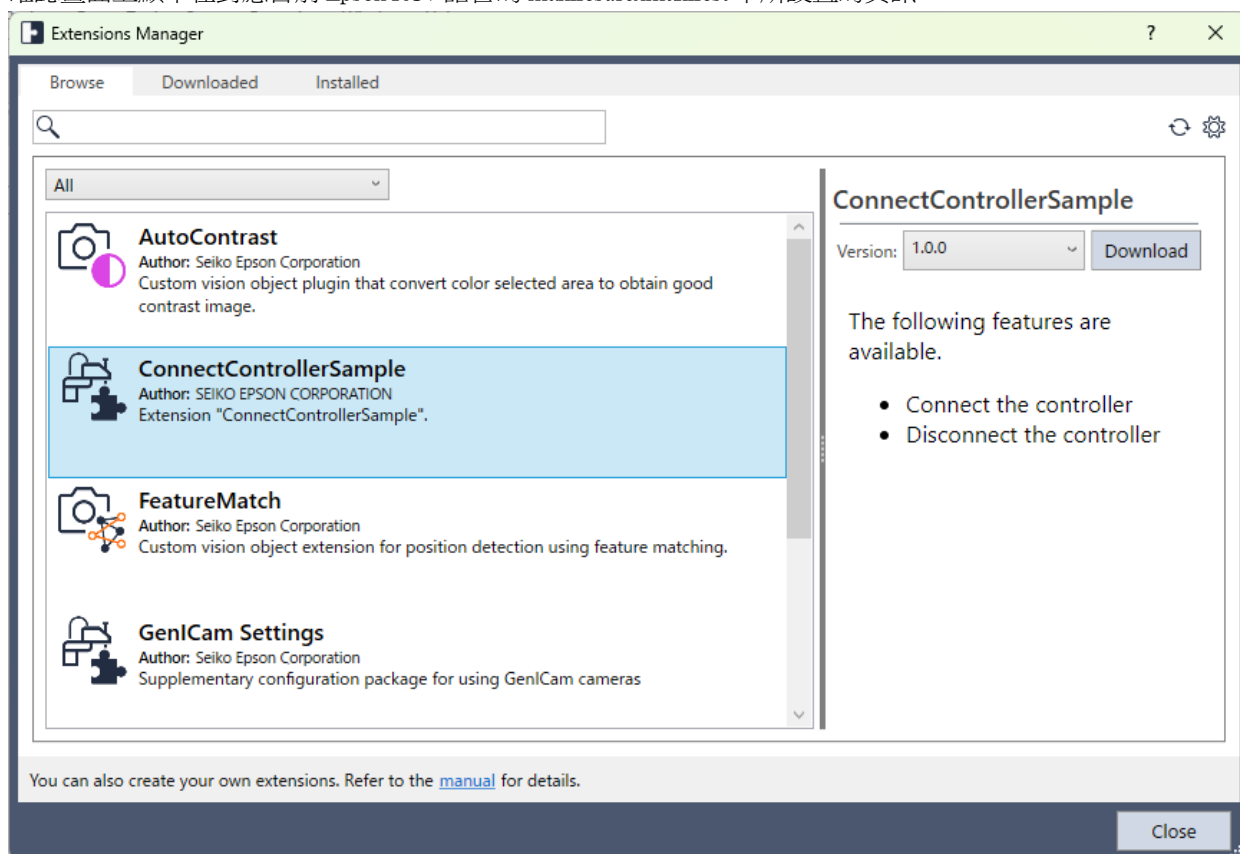
- 修改 Vendor Info, Extension Name, Summary, Description 行中各語言的字串。

Validity: Valid
Category: RC+ Extensions
Id: ConnectControllerSample - fc3c3051-642d-4b45-8c1e-c4ba811663a8 New Guid
Vendor: com.example (Reverse FQDN expected)
Version: 1.0.0
Released At: 2025/04/09 00:00:00 (UTC)

	en	ja	de	fr	es	zh-Hans	zh-Hant
Vendor Info	SEIKO EPSON CORPORATION	SEIKO EPSON CORPORATION	SEIKO EPSON CORPORATION	SEIKO EPSON CORPORATION	SEIKO EPSON CORPORATION	SEIKO EPSON CORPORATION	SEIKO EPSON CORPORATION
Extension Name	ConnectControllerSample	ConnectControllerSample	ConnectControllerSample	ConnectControllerSample	ConnectControllerSample	ConnectControllerSample	ConnectControllerSample
Summary	Extension "ConnectControllerSample".	拡張機能 "ConnectControllerSample".	Erweiterung "ConnectControllerSample".	Extension "ConnectControllerSample".	Extensión "ConnectControllerSample".	擴展功能 "ConnectControllerSample".	擴展功能 "ConnectControllerSample".
Description	The following features are available. - Connect the controller - Disconnect the controller	以下の機能を利用可能です。 - コントローラーを接続する - コントローラーを切断する	Folgende Funktionen stehen zur Verfügung: - Controller verbinden - Controller trennen	Les fonctions suivantes sont disponibles: - Connecter un contrôleur - Déconnecter un contrôleur	Las siguientes funciones están disponibles: - Conectar un controlador - Desconectar un controlador	可使用以下功能： - 連接控制器 - 斷開控制器連接	可使用以下功能： - 連接控制器 - 斷開控制器連接
Image File	Image.png						
Release Note File							
EULA File							
OSS Notice File							

9. 依照 [Extension 的發布] 的步驟，在擴展功能管理器中顯示已開發的 Extension。

- 確認畫面上顯示在對應目前 Epson RC+ 語言的 manifest.rcxmanifest 中所設置的資訊。



10. 依照 [Extension 的使用] 步驟，啟用 Extension，並確認可使用已開發的功能。

6. PC Vision 自定義視覺物件的實作

6.1 概述

所謂「自定義視覺物件 (Custom Vision Object)」是指可作為擴展功能安裝於 RC+80 Vision Guide 系統中的自定義化影像辨識物件。

自 Epson RC+ 8.0 Ver8.1.3.0 版本起，PC Vision 支援使用自定義視覺物件 (Custom Vision Object)。

透過使用自定義視覺物件，能夠將自定義的影像處理功能整合至 RC+80 Vision Guide 系統中。在自定義視覺物件中，可以實作如 ImageOp 這類的影像轉換處理，或如 Geometric 這類的位置偵測處理。亦可利用 OpenCV 等外部函式庫，實現現有物件所不具備的影像處理功能。

本章將說明從自定義視覺物件的實作到套件製作的開發流程。

※關於開發完成之自定義視覺物件的使用方式，請參閱「RC+ Extensions 的使用」或「Vision Guide 8.0 軟體篇 自定義視覺物件」。

※ GitHub 上已提供範例程式與 API 的說明。關於程式的實作方式等，亦請參考此資源。

<https://github.com/Epson-Robots/rcplus-extensions>

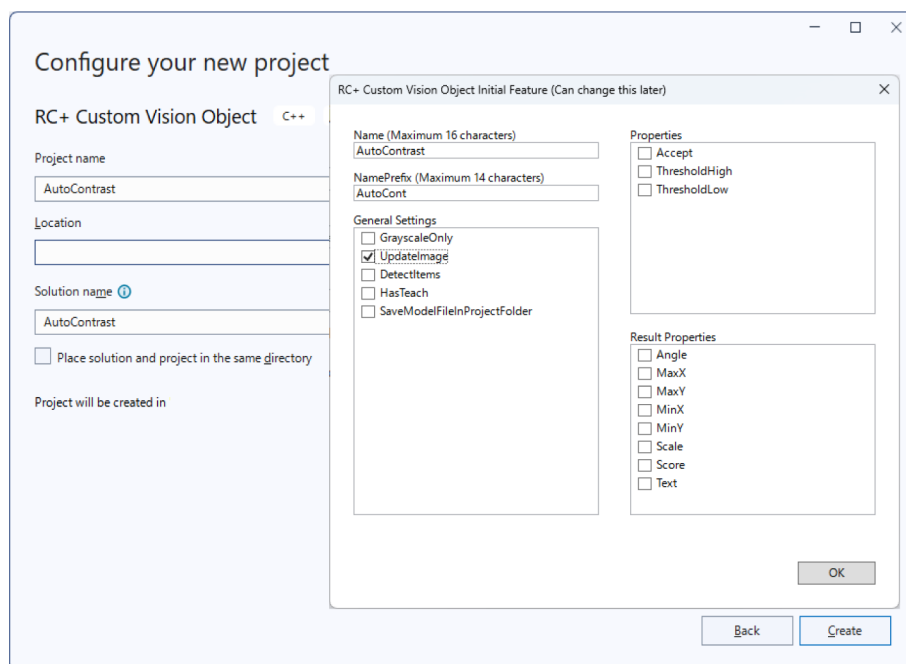
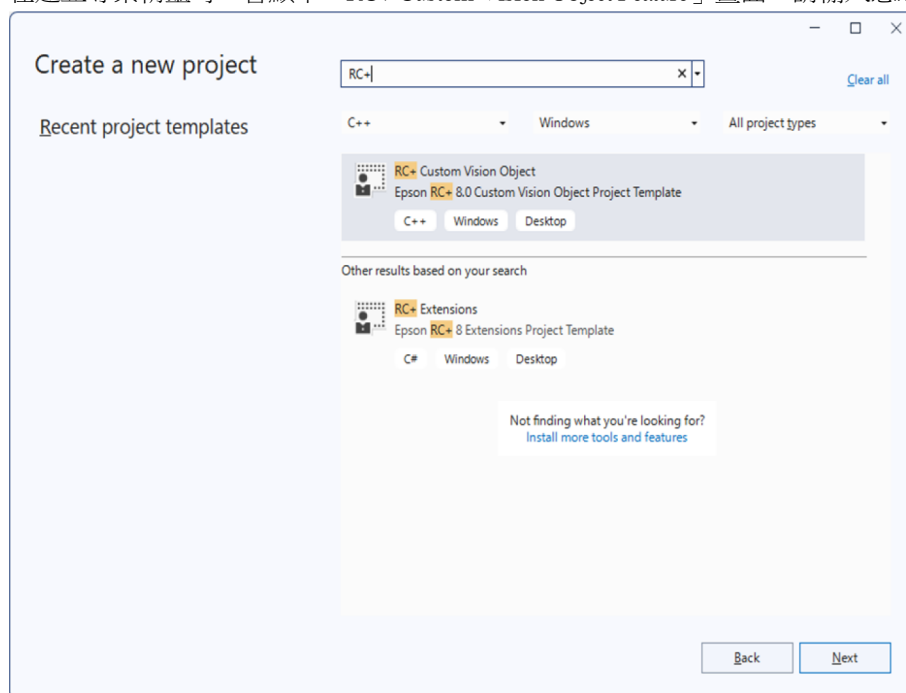
6.2 安裝

擴展功能為 Windows DLL，需以 Visual C++ 進行開發，因此必須安裝 VisualStudio。請事先安裝 RC+ Extensions 共通的 VSIX。詳細內容請參閱[5.2 安裝]。

6.3 開始使用

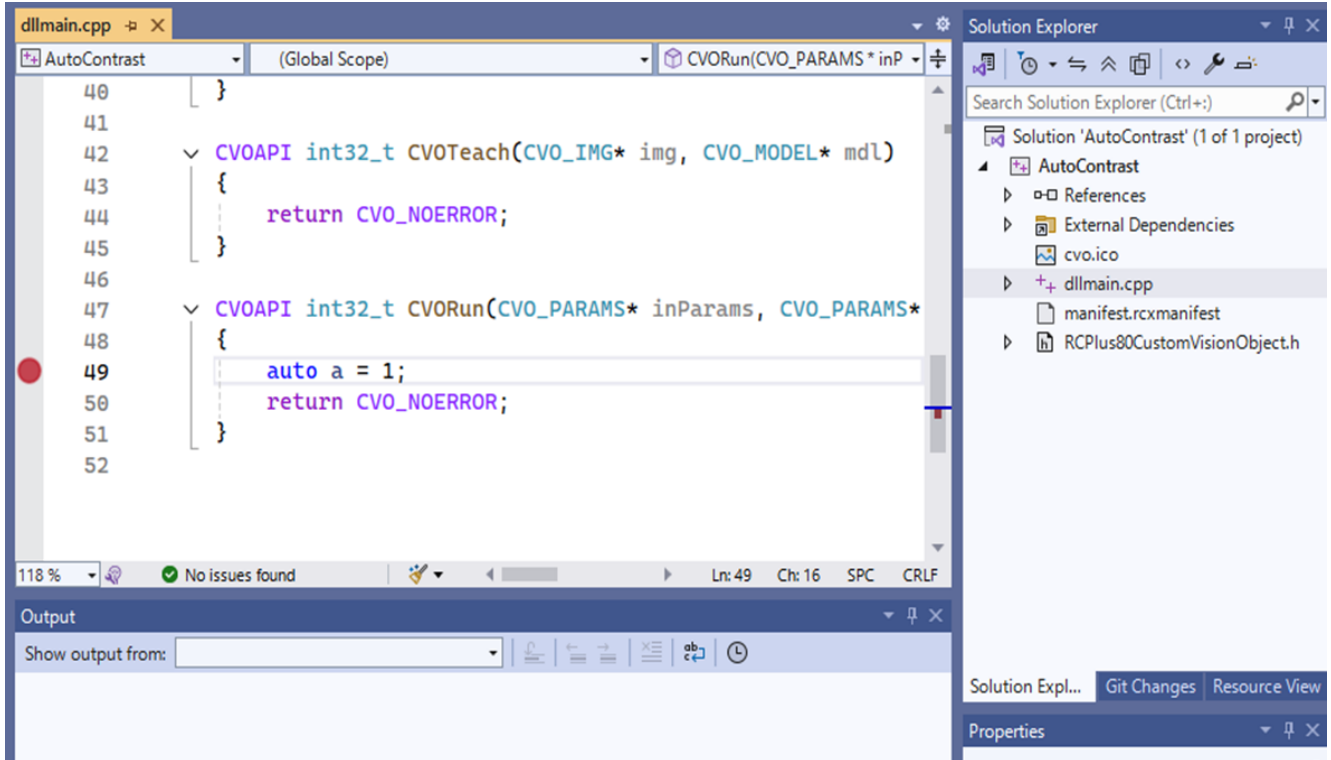
RC+ Extensions 共通的 VSIX 中包含了 Visual Studio 的自定義視覺物件專案範本「RC+ Custom Vision Object」。
請使用此範本建立 Visual Studio 專案。

在建立專案精靈時，會顯示「RC+ Custom Vision Object Feature」畫面，請輸入您的自定義視覺物件的設定檔資訊。



建立專案後，請於 dllmain.cpp 進行實作。

由於此檔案已預先實作必要的 API（匯出的函式），請於函式內部進行實作。



請將目標專案設為啟動專案，並以偵錯模式執行。

將會啟動用於開發自定義視覺物件的工具，您可以確認自定義視覺物件的運作情形。此外，也可以透過設定中斷點來進行偵錯執行。



6.4 Extension 發佈

請對目標專案執行 Release 組建。在方案資料夾下會建立 Packages 資料夾，於此資料夾下會產生副檔名為 rcxpkg 的自定義視覺物件套件。透過擴展功能管理器安裝此套件後，您的自定義視覺物件即可於 RC+ 中使用。