# How to get information of running apps and processes in Universal Windows Platform apps

This sample demonstrates how to get information of running apps and processes in Universal Windows Platform apps.

**Sample prerequisites**

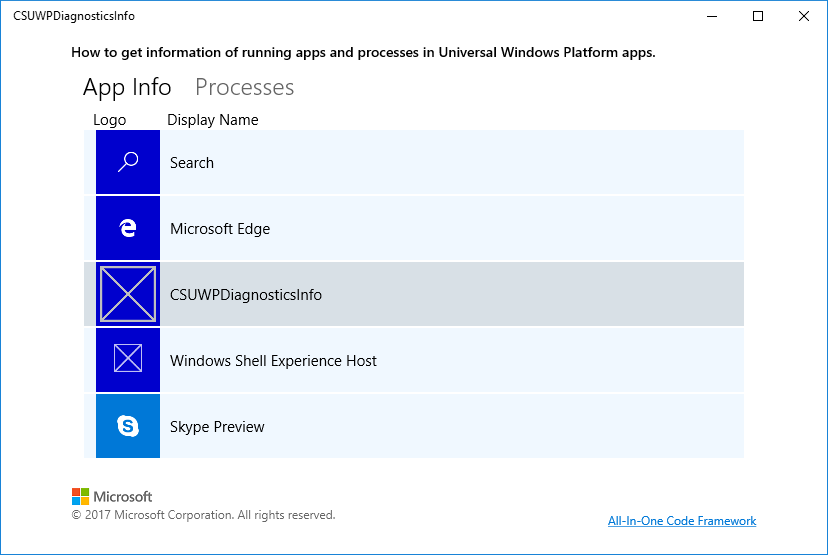
* Visual Studio 2017 or above [[Visual Studio Home Page](https://www.visualstudio.com/)]
* [Windows 10 Creators Update](https://support.microsoft.com/en-us/instantanswers/d4efb316-79f0-1aa1-9ef3-dcada78f3fa0/get-the-windows-10-creators-update)

## Building the sample

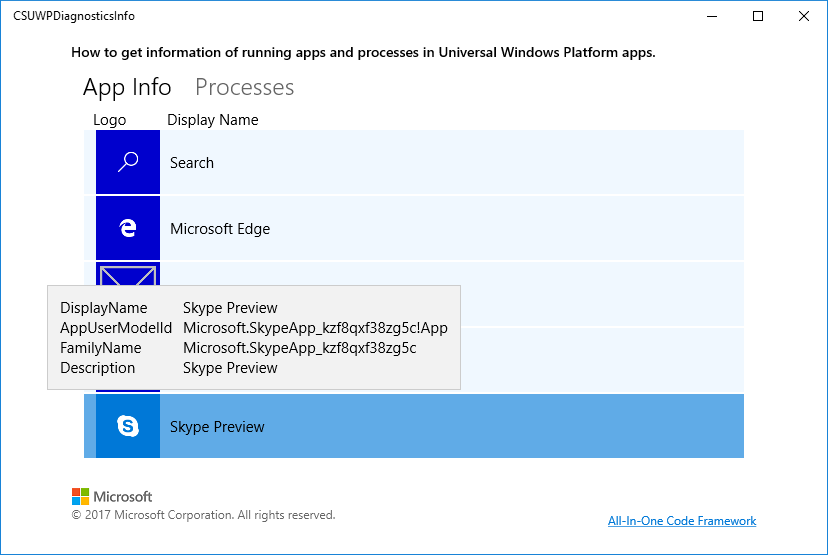
* Open the sample solution “UWPDiagnosticsInfo.sln” using Visual Studio
* Right click the project “UWPDiagnosticsInfo” and select Restore Packages
* Press **F6** Key or select **Build -> Build Solution** from the menu to build the sample.

## Running the sample

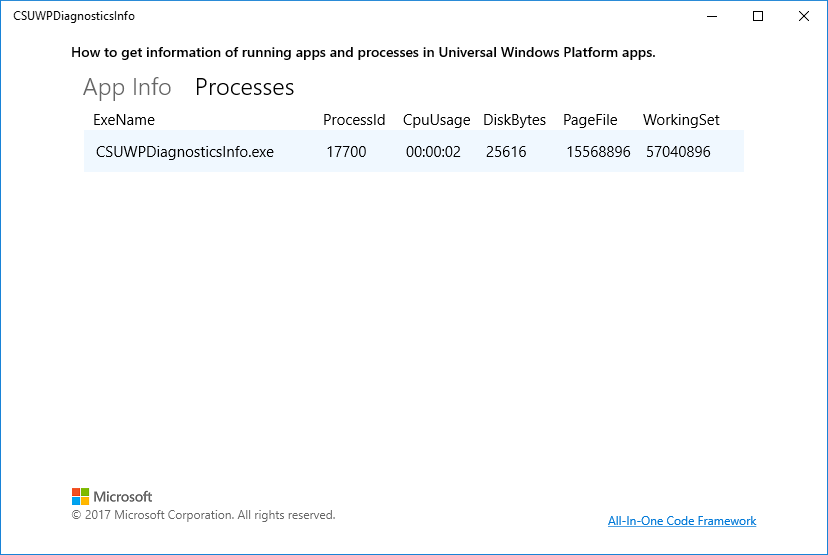
* Open the sample solution using Visual Studio, select **Local Machine** in the tool bar then press F5 Key or select Debug -> Start Debugging from the menu.



* Click on the item to see the details.



* Click on the pivot item **Processes** to see the processes (accessible to the caller)



* There is only one process in the same **AppContainer** with the running app.

**Using the code**

MainPage:

private void LoadProcesses()

{

mainViewModel.ProcessList.Clear();

List<ProcessDiagnosticInfo> processList = ProcessDiagnosticInfo.GetForProcesses().ToList();

processList.ForEach(o => mainViewModel.ProcessList.Add(new ProcessInfoModel(o)));

}

private async void LoadAppInfo()

{

mainViewModel.AppInfoList.Clear();

IList<AppDiagnosticInfo> list = await AppDiagnosticInfo.RequestInfoAsync();

list.ToList().ForEach(o => mainViewModel.AppInfoList.Add( new AppInfoModel(o.AppInfo)));

}

AppInfoModel:

public AppInfoModel(AppInfo appInfo)

{

AppUserModelId = appInfo.AppUserModelId;

DisplayName = appInfo.DisplayInfo.DisplayName;

Description = appInfo.DisplayInfo.Description;

PackageFamilyName = appInfo.PackageFamilyName;

RandomAccessStreamReference logoStream = appInfo.DisplayInfo.GetLogo(new Windows.Foundation.Size(64, 64));

SetLogo(logoStream);

}

private async void SetLogo(RandomAccessStreamReference logoStream)

{

IRandomAccessStreamWithContentType logoContent = await logoStream.OpenReadAsync();

BitmapImage bitmap = new BitmapImage();

await bitmap.SetSourceAsync(logoContent);

LogoImage = bitmap;

}

ProcessInfoModel

public ProcessInfoModel(ProcessDiagnosticInfo process)

{

ProcessCpuUsageReport cpuReport = process.CpuUsage.GetReport();

if (cpuReport != null)

{

TimeSpan cpuUsageTime = cpuReport.KernelTime + cpuReport.UserTime;

CpuUsageTime = string.Format("{0:hh\\:mm\\:ss}", cpuUsageTime);

}

ProcessDiskUsageReport diskReport = process.DiskUsage.GetReport();

if (diskReport != null)

{

DiskBytesCount = diskReport.BytesReadCount + diskReport.BytesWrittenCount;

}

ProcessMemoryUsageReport memoryReport = process.MemoryUsage.GetReport();

if (memoryReport != null)

{

PageFileSize = memoryReport.PageFileSizeInBytes;

WorkingSetSize = memoryReport.WorkingSetSizeInBytes;

}

ProcessId = process.ProcessId;

ExeName = process.ExecutableFileName;

}

**More information**

[ProcessDiagnosticInfo](https://docs.microsoft.com/en-us/uwp/api/windows.system.diagnostics.processdiagnosticinfo)

[AppDiagnosticInfo](https://docs.microsoft.com/en-us/uwp/api/windows.system.appdiagnosticinfo)

[Pivot control](https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.controls.pivot)