Intel Unite®

SDK

Reference Design for Static Instances

Guide

August 2019

Revision 1.0

Intel Confidential

Revision History

|  |  |  |
| --- | --- | --- |
| Revision | Description | Date |
| 1.0 | Initial release | August 2019 |
|  |  |  |
|  |  |  |
|  |  |  |

**Legal Disclaimers and Copyrights**

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel, the Intel logo, Intel® Core™ vPro™ processor family are trademarks of Intel Corporation in the United States and/or other countries.

\*Other names and brands may be claimed as the property of others

Contents

[1 Terminology 4](#_Toc19702271)

[2 UC Requirements 5](#_Toc19702272)

[2.1 Display WCs 5](#_Toc19702273)

[2.1.1 MVVM files 5](#_Toc19702274)

[2.1.2 Implement Factory Strategy 8](#_Toc19702275)

[2.1.4 Allocate HubView.Type 9](#_Toc19702276)

Figures

[Figure 2 Add View 6](#_Toc19702277)

[Figure 3 View Design Example 7](#_Toc19702278)

Code

[Code 1 WCsDisplay View Model 6](#_Toc19702279)

[Code 2 WCsView xaml 6](#_Toc19702280)

[Code 3 WCsDisplay.cs 7](#_Toc19702281)

[Code 4 enum HubView.Type 8](#_Toc19702282)

[Code 5 WCsDisplayFactory 9](#_Toc19702283)

[Code 6 Allocate HubView Example 9](#_Toc19702284)

# Terminology

Table 1. Terminology

| Term | Description |
| --- | --- |
| UI | User Interface |
| WPF | Windows\* Presentation Foundation |
| Hub | Intel® Core™ vPro™ processor family-based PC running the Intel Unite Hub application |
| Client | Device used to connect to the hub |
| Server | Device that manages access and configuration of Hub and Client Devices |
| Partial Background View | Area on the background layer where applications can reside |
| WC | Web Conference |
| UC | Unified Communications |
| UC User | Unified Communication User |

## Static Instances

Static Instances of classes, objects, and other data allows access to anywhere within a plugin.

### Dispatcher, RuntimeContext and Logging

To access these static objects our test application will have to set these values as seen below.

1. In the **Static** folder, create a class named UnitePluginConfig.cs.

Code - Static Instances example

|  |
| --- |
| public static Dispatcher CurrentUiDispatcher { get; set; }    public static IHubModuleRuntimeContext RuntimeContext { get; set; }  public static Func<FrameworkElement, MarshalNativeHandleContract> CreateContract { get; set; }  public static class Log  {  public static void CriticalMessage(string source, string message)  {  RuntimeContext.LogManager.LogMessage(ModuleConstants.ModuleInfo.Id, LogLevel.Critical, source, message);  }  public static void DebugMessage(string source, string message)  {  RuntimeContext.LogManager.LogMessage(ModuleConstants.ModuleInfo.Id, LogLevel.Debug, source, message);  }  } |

The Log class provides an easier means to log to the SQL database. The three main instances are used by Unite Plugins to access various functions and methods. It is very important that the correct runtime context is set, which is a **IHubModuleRuntimeContext**.

Code - RuntimeContext static instance

|  |
| --- |
| public override void Load()  {  UnitePluginConfig.RuntimeContext = RuntimeContext;  } |

Code - Test App Dispatcher instance

|  |
| --- |
| HubFeatureModuleBase module = new PluginModuleHandler(mainWindow.DataContext as HubModuleRuntimeContext)  {  CurrentUiDispatcher = Current.Dispatcher  }; |

Typically, the Contract is created in the HubViewBase.cs in the Allocate method.

Code - Contract instance

|  |
| --- |
| public override void Allocate()  {  ...  UnitePluginConfig.Contract = CreateContract(HubView);  ...  } |