This document describes how an application developer can start with a UWC Reference Implementation on EII and add simple Video Analytics usage to the base UWC implementation.

The context is a for a PoC implementation and the base version of UWC for this document is UWC1.6.1.

Follow the steps as mentioned in User guide [https://open-edge-insights.github.io/uwc-docs/Pages/page\_04.html or ReadMe.md](https://open-edge-insights.github.io/uwc-docs/Pages/page_04.html%20or%20ReadMe.md) to clone the repos as needed (Includes git clone commands).

Also, can refer the Readme at <https://github.com/open-edge-insights/uwc/blob/feature/ngk%2FUWC_Video_usecase__POC/README.md> to provision/build using UWC scripts.

**Repos to be used for use-case 9**:

Git clone commands for rest of the repos (not documented in User-guide, which are specific to this particular Developer’s use case).:

VI repo-> git clone <https://github.com/open-edge-insights/video-ingestion.git> IEdgeInsights/VideoIngestion.

VA repo -> git clone <https://github.com/open-edge-insights/video-analytics.git> IEdgeInsights/VideoAnalytics

ImageStore repo-> git clone <https://github.com/open-edge-insights/video-imagestore.git> IEdgeInsights/ImageStore

WebVisualizer repo-> git clone <https://github.com/open-edge-insights/video-web-visualizer.git>

IEdgeInsights/WebVisualizer

Post cloning, apply the below patches in respective repos:

VI –> VideoIngestion\_cfg.patch

VA -> VideoAnalytics\_cfg.patch

WebVisualizer -> WebVisualizer\_cfg.patch

Imagestore -> ImageStore\_cfg.patch.

ZmqBroker -> ZmqBroker\_cfg.patch.

All these patches can be found in the “Developer-patches” folder in [https://github.com/open-edge-insights/uwc/tree/feature/ngk/UWC\_Video\_usecase\_\_POC/Developer\_patches](https://github.com/open-edge-insights/uwc/tree/feature/ngk/UWC_Video_usecase__POC/exon_patches).

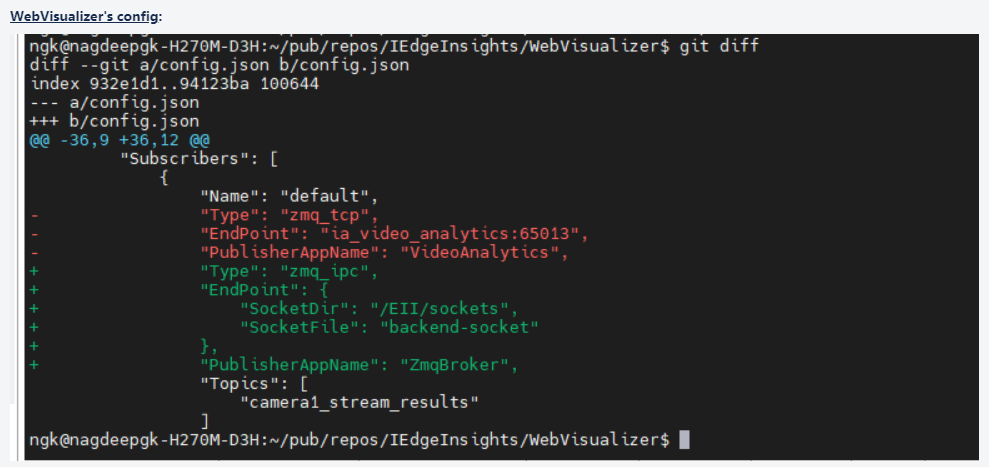
While running “02 provision\_UWC.sh" script, please select the use cases 9, 10, 11 which are specifically for Developer’s requirement for integrating UWC middleware with video use cases. Note: only the branch <https://github.com/open-edge-insights/uwc/tree/feature/ngk/UWC_Video_usecase__POC> contains the code changes for this use case. Hence, let’s keep note to not select other branches.

1. Use case 9:
2. Serves as a reference "working" use case containing sample VI(Video-Ingestion)/VA(Video-Analytics) which supplies the video frames to be stored in the ImageStore database i.e MinIO. To visually view the image content that is being stored into the MinIO DB, a ready-made “WebVisializer” has also been provided.
3. Post applying the respective git patches, refer the config.json of each service which contains the interfaces to be used for connecting to the services.
4. For example, in this use case, VI -> VA -> ZmqBroker -> ImageStore/WebVisualizer.
5. Here in the text in red, the VA publishes images to the ZmqBroker on topic “camera1\_stream\_results”. ImageStore/WebVisualizer subscribes to the images on the published topic “camera1\_stream\_results”.
6. So the config.json of VA, ZmqBroker & Imagestore/WebVisualizer are point of interest here.
7. This can be used as a reference code. Developer can replace the VI/VA services with their video services to make the video use case work with UWC/EII bus.









Pre-requisite to port Developer’s Video services to be compatible on EII bus. Developer can replace their Video-Service instead of VI->VA. The other requirements include:

* Any service which works on EII must be complaint with EII config manager. So, make the new Developer video service complaint with the EII Config manager as according to <https://github.com/open-edge-insights/eii-core/blob/master/common/libs/ConfigMgr/README.md#interfaces>.
* Also refer to the ZmqBroker ReadMe to change broker interfaces <https://github.com/open-edge-insights/eii-zmq-broker/blob/master/README.md> .

1. Use case 10: This use case is a placeholder use case. Running the basic UWC services, [PLACEHOLDER FOR DEVELOPER-TO-ADD-VIDEO-USE-CASES], ImageStore, Sample publisher with Telegraf/InfuxDB.
2. Make sure to add the Developer’s video service in the uwc recipe file “uwc-uwc-pipeline-basic-VI-VA-imagestore-Ref.yml” in

<https://github.com/open-edge-insights/uwc/tree/feature/ngk/UWC_Video_usecase__POC/uwc_recipes> to include the Developer’s video service entry (after having ported non-EII msg bus compliant service to EII msg bus as explained in the pre-requisite section).

1. As a reference one can use the use case 9’s recipe file “ uwc-uwc-pipeline-basic-VI-VA-imagestore-Ref.yml” in <https://github.com/open-edge-insights/uwc/tree/feature/ngk/UWC_Video_usecase__POC/uwc_recipes>.
2. This use case also support, time series data sing Telegraf/INfluxDB and a sample EII message bus publisher. (included in the recipe file). These services which serve as a sample/reference can be modified as per the custom need.
3. Use case 11: This use case is a placeholder use case (subset of the above, which doesn’t have the time series use cases like Telegraf/INfluxDB). Running the basic UWC services, [DEVELOPER-TO-ADD-VIDEO-USE-CASES] & ImageStore.
4. Make sure to add the Developer’s video service in the uwc recipe file “uwc-pipeline-basic-imagestore-Ref.yml” in

<https://github.com/open-edge-insights/uwc/tree/feature/ngk/UWC_Video_usecase__POC/uwc_recipes> to include the Developer’s video service entry (after having ported non-EII msg bus compliant service to EII msg bus as explained in the pre-requisite section).

1. As a reference one can use the use case 9’s recipe file “ uwc-uwc-pipeline-basic-VI-VA-imagestore-Ref.yml” in <https://github.com/open-edge-insights/uwc/tree/feature/ngk/UWC_Video_usecase__POC/uwc_recipes>