This is the facet as documented in Part 84. As you can see, we only mandated SKS PubSub Key Management and SKS PubSub Model Push.



**Problem #1**

The SKS PubSub Key Management CU says this:



SKS PubSub Key Management is mandatory, but the SecurityGroupType is pulled in through SKS PubSub Model which is optional.



I think we have two options to fix this issue; create a new CU that specifically calls out support for the SecurityGroupType, or just make PubSub Model SKS mandatory. Making PubSub Model SKS mandatory then has the effect of bringing in one other type (SecurityGoupFolderType) that we did not want to require. However in looking at the types, instances of SecurityGroupType probably belong in folders of SecurityGoupFolderType.

**Problem #2**

During our discussions on SKS behavior we agreed that we did not want to require controllers to act as a general purpose SKS for the whole network. It is analogous to having a CM on the controller that works to establish its own connections, but not all connections. We made SKS PubSub Model Push mandatory thinking that this was the correct CU to specify this behavior. The CU states the following:



Unfortunately, the PubSubPushTargetFolderType specifies the AddPushTarget and RemovePushTarget methods as mandatory. This effectively would open the controller to be a general SKS if exposed.



My proposed solution here is to convince the PubSub WG to change these to Optional. That would allow controllers to support the folder type without exposing the methods.

**Problem #3**

The last problem deals with other text in SKS PubSub Model Push. It states that the SKS will invoke the SetSecurityKeys Method on the target. It is not clear to me that we have required support for that method anywhere. The Method is part of the PublishSubscribeType as optional.



There is no PubSub Model Base CU in the tool, so it is not clear how this type is pulled in. It is clear that this type should be supported by any implementation that supports PubSub, but I am not sure we have that covered. And since the method is optional, we would need a separate CU for that anyway. My proposed solution is to write a new CU for support of the SetSecurityKeys and this would be required on all UAFX implementations.