Understanding how tag changes at the Edge affect MQTT Engine

This document describes the various tag actions at an Edge device and the action required to represent the Edge tags correctly at MQTT Engine.

> Where two actions are described, the action highlighted in bold is the one required to represent the tags correctly at MQTT Engine.

> To use the sample tags provided to test the various scenarios, install MQTT Distributor, MQTT Engine and MQTT Transmission. Configure a Transmitter with Tag Provider "default", an empty Tag Path and leave the SparkplugIDs as empty. Import the SampleTags.json and refresh Transmission.

Non UDT Tags at Edge

Action at Edge	Result	Action	Result
Create new group level folder with no tags *Add G2*	On add: No change at Engine	**No action required**	No folder created at Engine
		Refresh Transmission	No folder created at Engine
Create new edge level with no tags *Add G2/E2*	On add: No change at Engine	**Refresh Transmission**	Folder path is created at Engine
Create new device level folder with no tags *Add G2/E2/D2*	On add: No change at Engine	**Refresh Transmission**	Folder path is created at Engine
Create new folder under the device level with no tags *Add G2/E2/D2/New Folder*	On add: No change at Engine	**Refresh Transmission**	No folder path created at Engine
Add a tag to folder *Add G1/E1/D1/Tag2 as type Integer*	On add: No change at Engine	**Refresh Transmission**	Tag is created under tag folder path at Engine
Delete a tag *Delete G1/E1/D1/Tag2*	On delete: Engine will mark tag as Bad_Stale	**Delete tag at Engine**	Tag deleted at Engine
Delete tag and add tag with the same name *Delete G1/E1/D1/Tag1* *Add G1/E1/D1/Tag1 as type Boolean*	On delete: Engine will mark tag as Bad_Stale On add: No change as Engine	**Refresh Transmission**	Tag will be configured at Engine under tag folder path with any new properties and GOOD quality
Delete tag and create a folder which has the same name as the tag. Add a tag to this folder. *Delete G1/E1/D1/Tag1* *Add G1/E1/D1/Tag1/Tag2*	On delete: Engine will mark tag as Bad_Stale On add: No change at Engine	**Delete tag at Engine before a Transmission refresh is performed**	New tag will be created under new tag folder path at Engine
		Refresh Transmission without deleting tag at Engine	Error similar to below logged, tag will not be deleted and no tag path folder will be created Error processing edit for tag path '[MQTT Engine]Edge Nodes/My MQTT Group/Edge Node 0964bf/PLC 1/New Tag/New Tag1': Bad_Unsupported("The target path '[MQTT Engine]Edge Nodes/My MQTT Group/Edge Node 0964bf/PLC 1/New Tag' cannot accept children tags.")
Delete tag and parent folder and add a tag with same name as the parent folder *Delete G1/E1/D1/Tag1/Tag2* *Add G1/E1/D1/Tag1*	On delete: Engine will mark tag as Bad_Stale On add: No change at Engine	**Refresh Transmission**	Tag and folder will be configured at Engine

UDT Tags at Edge with Transmission Convert UDTs enabled

> ⚠ This will convert UDT members to normal tags before publishing. Tags representing the UDT member will retain their member path prefixed by the UDT Instance name.

Action at Edge	Result	Action	Result
Add UDT Definition and UDT Instance *Create TestUDT1 with tag "New Tag" as type Integer with value = 10* *Add G1/E1/D1/NewInstance1 as TestUDT1*	On add: No change at Engine	**Refresh Transmission**	Folder will be created named as the UDT Instance and UDT member tag will be created under tag folder at Engine
Add new member tag to UDT Definition *Add "New Tag 1" as type Boolean with value = True to TESTUDT1*	On add: No change at Engine	**Refresh Transmission**	New UDT member tag created under folder named as UDT Instance at Engine
Delete member tag from UDT Definition *Delete "New Tag 1" from TESTUDT1*	On delete: Engine will mark the tag as Bad_NotFound	Refresh Transmission	No change at Engine
		Delete tag at Engine	Tag deleted at Engine
Delete a member tag from a UDT Definition and add one with the same name in the same hierarchy position *Delete "New Tag" from TESTUDT1* *Add "New Tag" as type Boolean with value = True*	On delete: Engine will mark the tag as Bad_NotFound On add: No change at Engine	**Refresh Transmission**	UDT member tag will be configured with new properties under folder named as UDT Instance at Engine
Add a Child UDT definition to a UDT Definition *Add TestUDT/t4Instance as type T4*	On add: No change at Engine	**Refresh Transmission**	New folder and member tags created at Engine named as the new Child UDT instance under the folder named as UDT Instance
Delete a Child UDT definition from a UDT definition *Delete TestUDT/t4Instance*	On delete: Engine will mark all tags under the Child UDT folder as Bad_NotFound	**Delete tags at Engine**	Tags deleted at Engine
Delete a Child UDT definition from a UDT definition and add one in the same hierarchy position with the same name and tag folder structure *Delete TestUDT/t3Instance with type T3* *Add TestUDT/t3Instance as type T3*	On delete: Engine will mark all tags under the Child UDT folder as Bad_NotFound On add: Engine will mark all the tags under the Child UDT folder as Good quality	**No action required**	
Delete a Child UDT definition from a UDT definition and add one in the same hierarchy position with the same name and a different tag folder structure *Delete TestUDT/t3Instance with type T3* *Add TestUDT/t3Instance as type T4*	On delete: Engine will mark all tags under the Child UDT folder as Bad_NotFound On add: No change at Engine	**Delete tags at Engine before Transmission refresh is performed**	New folder and member tags are created at Engine following the hierarchy of the Child UDT instance
		Refresh Transmission without deleting tags at Engine	New folder and member tags are created at Engine following the hierarchy of the Child UDT instance. Original tags are left with a quality of Bad_NotFound
Delete a child UDT definition member tag (this deletes the Child UDT definition from the parent UDT definition) *Delete TestUDT/t3Instance/t3Tag1*	On delete: Engine will mark all tags under the Child UDT folder as Bad_NotFound	**Delete tags at Engine**	Tags deleted
Delete a child UDT definition member tag (this deletes the Child UDT definition from the parent UDT definition) and recreate with the same name and tag structure *Delete TestUDT/t3Instance with type T3* *Add Test/UDT/t3Instance as type T3*	On delete: Engine will mark all tags under the Child UDT folder as Bad_NotFound On add: engine will mark all tags under the Child UDT folder as Good quality	**No action required**	
Delete a child UDT definition member tag (this deletes the Child UDT definition from the parent UDT definition) and recreate with the same name and different tag structure *Delete TestUDT/t3Instance with type T3* *Add TestUDT/t3Instance as type T2*	On delete: Engine will mark all tags under the Child UDT folder as Bad_NotFound On add: No change at Engine	**Delete tags at Engine before Transmission refresh is performed**	New folder and members tags are created at Engine following the hierarchy of the Child UDT instance.

		Refresh Transmission without deleting tags at Engine	New folder and member tags are created at Engine following the hierarchy of the Child UDT instance. Original tags are left with a quality of Bad_NotFound

UDT Tags at Edge with Transmission Convert UDTs disabled

> Changes made at an Edge device to UDT Definitions are not propagated through the system. You will see that any tag instances of that UDT at the Edge device will reflect the changes, but the recorded UDT Definition at MQTT Engine will not change nor will tag instances for other Edge devices using that UDT Definition.

Publish UDT Definitions Enabled

With the Publish UDT Definitions set to True, the UDT Definitions will be included in NBIRTH messages where each UDT Definition includes an "md5" metric. For each named UDT Definition, MQTT Engine will compare the MD5 sum received in the NBIRTH with the MD5 sum of the UDT Definition stored to detect differences and identify collisions. If a collision is detected, the UDT Definition received in the NBIRTH will be ignored and a warning will be logged from the com.cirruslink.mqtt.engine.gateway.sparkplug.SparkplugBPayloadHandler logger similar to:

W	SparkplugBPayloadHandler	08Mar2024 10:20:43	UDT definition collision detected for TestUDT. Set log level to 'TRACE' for details.

Turning this logger to TRACE will show the UDT definition at MQTT Engine and also the UDT definition published in the NBIRTH message. The offending edge node can be identified from the NBIRTH topic.

T	SparkplugBPayloadHandler	08Mar2024 10:30:06	Metric received on topic spBv1.0/G1/NBIRTH/E1: Metric [name=TestUDT, alias=null, timestamp=null, dataType=Template, isHistorical=null, isTransient=null, metaData=null, properties=null, value=Template [version=null, templateRef=null, isDefinition=true, metrics=[Metric [name=New Tag, alias=null, timestamp=null, dataType=Int32, isHistorical=null, isTransient=null, metaData=null, properties=null, value=10, isNull=false], Metric [name=New Tag 1, alias=null, timestamp=null, dataType=Boolean, isHistorical=null, isTransient=null, metaData=null, properties=null, value=true, isNull=false], Metric [name=New Tag 2, alias=null, timestamp=null, dataType=Int32, isHistorical=null, isTransient=null, metaData=null, properties=null, value=20, isNull=false], Metric [name=t3Instance, alias=null, timestamp=null, dataType=Template, isHistorical=null, isTransient=null, metaData=null, properties=null, value=null, isNull=true]], parameters=null], isNull=false]
T	SparkplugBPayloadHandler	08Mar2024 10:30:06	Metric on MQTT Engine side: Metric [name=TestUDT, alias=null, timestamp=null, dataType=Template, isHistorical=null, isTransient=null, metaData=null, properties=null, value=Template [version=null, templateRef=null, isDefinition=true, metrics=[Metric [name=New Tag, alias=null, timestamp=null, dataType=Int32, isHistorical=null, isTransient=null, metaData=null, properties=null, value=10, isNull=false], Metric [name=New Tag 1, alias=null, timestamp=null, dataType=Boolean, isHistorical=null, isTransient=null, metaData=null, properties=null, value=true, isNull=false], Metric [name=t3Instance, alias=null, timestamp=null, dataType=Template, isHistorical=null, isTransient=null, metaData=null, properties=null, value=null, isNull=true]], parameters=null], isNull=false]
W	SparkplugBPayloadHandler	08Mar2024 10:30:06	UDT definition collision detected for TestUDT. Set log level to 'TRACE' for details.
T	SparkplugBPayloadHandler	08Mar2024 10:30:06	MD5 of the TestUDT metric (received on topic spBv1.0/G1/NBIRTH/E1): c9ca6bd917b12f3cbddd80bd81d74ffc
T	SparkplugBPayloadHandler	08Mar2024 10:30:06	MD5 of the TestUDT metric (calculated) on the MQTT Engine side: 620dab50d6cc129920c21fd32d390e86
T	SparkplugBPayloadHandler	08Mar2024 10:30:06	MD5 of the TestUDT metric (from metadata) on the MQTT Engine side: 620dab50d6cc129920c21fd32d390e86
D	SparkplugBPayloadHandler	08Mar2024 10:30:06	UDT Definition TestUDT already exists, validating
D	SparkplugBPayloadHandler	08Mar2024 10:30:06	New UDT Definition TestUDT

Publish UDT Definitions Disabled

With Publish UDT Definitions set to False, the UDT Definitions will not be included in NBIRTH messages and MQTT Engine will not be able to detect differences and identify collisions.

MQTT Engine will attempt to process the incoming NBIRTH or DBIRTH messages causing errors similar to the one shown below:

E provider	08Mar2024 10:46:37	Error processing edit for tag path '[MQTT Engine]Edge Nodes/G1/E1/D1/Instance1/New Tag 2': Bad_Unsupported("The target path '[MQTT Engine]Edge Nodes/G1/E1/D1/Instance1' does not have item 'New Tag 2' for overrides, and cannot accept children tags.")
D SparkplugPayloadHandler	08Mar2024 10:46:37	Got Sparkplug message: spBv1.0/G1/DBIRTH/E1/D1

Action at Edge	Result	Action	Result
Add a UDT Instance *Add G1/E1/Instance2 with type T3*	On add: No change at Engine	**Transmission Refresh**	Instance configured at Engine
Delete member tag of UDT Instance (this will delete the UDT Instance) *Delete G1/E1/D1 /Instance1/New Tag*	On delete: MQTT Engine marks all tags for the Instance as Bad_Stale	**Delete tags at Engine**	Tags deleted at Engine
		Refresh Transmission without deleting tags at Engine	No change at Engine
Delete UDT Instance *Delete G1/E1/D1 /Instance1 with type TestUDT*	On delete: MQTT Engine marks all tags for the Instance as Bad_Stale	**Delete tags at Engine**	Tags deleted at Engine
		Refresh Transmission without deleting tags at Engine	No change at Engine
Delete UDT Instance and add with the same name and UDT type *Delete G1/E1/D1 /Instance1 with type TestUDT* *Add G1/E1/D1 /Instance1 with type TestUDT*	On delete: MQTT Engine marks all tags for the Instance as Bad_Stale On add: MQTT Engine will mark all tags for the Instance as Good quality	**No action required**	
Delete UDT Instance and add with the same name and different UDT type *Delete G1/E1/D1 /Instance1 with type TestUDT* *Add G1/E1/D1 /Instance1 with type T2*	On delete: MQTT Engine marks all tags for the Instance as Bad_Stale On add: No change at Engine	**Delete tags at Engine before Transmission refresh is performed**	Instance configured at Engine
		Refresh Transmission without deleting tags at Engine	Instance tags show Bad_Stale at Engine Various errors from tags.management.provider for new tags in definition structure Error processing edit for tag path '[MQTT Engine]Edge Nodes/G1/E1/D1/Instance1 /t3folder/t3Inst/t3Tag1': Bad_Unsupported("The target path '[MQTT Engine]Edge Nodes /G1/E1/D1/Instance1/t3folder/t3Inst' cannot accept children tags.") Error processing edit for tag path '[MQTT Engine]Edge Nodes/G1/E1/D1/Instance1 /t2Tag1': Bad_Unsupported("The target path '[MQTT Engine]Edge Nodes/G1/E1/D1 /Instance1' does not have item 't2Tag1' for overrides, and cannot accept children tags.")