

Make Available Steps to get parameters

STEPS to get all the parameterIds and fix the make available

1) Run the script to get all parameter Ids

```
--NOTE: This script will try to get all the top level device event parameter Ids and child dependency device event parameter Ids
DECLARE @systemName NVARCHAR(50) = 'MobileDevice.Hino'; --<-- Specify the Device you want to work with

USE DeviceConfiguration;
DECLARE @libId INT = -1; --Default Library

-- DEVICE
DECLARE @deviceKey INT = (SELECT DeviceKey FROM definition.Devices WHERE SYSTEMNAME = @systemName)

-- Add the above as a devicekey and then add all child devices
DECLARE @deviceKeys TABLE (
    deviceKey INT
);

INSERT INTO @deviceKeys VALUES(@deviceKey);
INSERT INTO @deviceKeys (deviceKey)
(
    SELECT DISTINCT dp.ChildDeviceKey
    FROM definition.DeviceDependencies dp WITH (NOLOCK)
    INNER JOIN library.Devices lpd
    ON dp.ParentDeviceKey = lpd.DeviceKey
    AND dp.DependencyType != 3 --Excluded
    AND lpd.LibraryKey = @libId
    AND dp.ParentDeviceKey = @deviceKey
)

--      Return Parameters (13)
SELECT DISTINCT ParameterId FROM definition.Parameters
    WHERE ParameterKey IN
        (SELECT ReturnParameterKey FROM definition.Events WITH (NOLOCK)
            WHERE EventKey IN
                (SELECT EventKey FROM [definition].[DeviceEventDependencies] ddep WITH (NOLOCK)
                    INNER JOIN @deviceKeys dk ON dk.deviceKey = ddep.deviceKey))
    OR ParameterKey IN (
--      Event Condition Parameters ()
        SELECT DISTINCT ParameterKey FROM library.EventConditions WITH (NOLOCK)
            WHERE LibraryKey IN (-1)
            AND EventKey IN
                (SELECT EventKey FROM [definition].[DeviceEventDependencies] ddep WITH (NOLOCK)
                    INNER JOIN @deviceKeys dk ON dk.deviceKey = ddep.deviceKey))
```

2) Get Parameter Examples

- Find an eg. for each parameter in DevicesData.xml
- copy them to the <parameters> section of this device
- replace the Ids making use of FMOnlineDB.dbo.GenerateBigId()
- Check this in
- Build
- Deploy

3) Try to make the Device available (EMPTY NEW ORG)

- Check for the parameterId exception

```
EXCEPTION_! All required parameters for this event have not been satisfied - ParameterId:
```

- Find which peripheral device is associated with it in order to add all of it's parameter ids
- Look for this in the log

```
Make Required Event Available for definition device " ...
```

- OR go one by one
- OR debug the code here... and look for all definitionDeviceIdsForNewDevicesAdded Ids and then look for all their events > conditions > parameters

```
foreach (long definitionDeviceId in definitionDeviceIdsForNewDevicesAdded)
{
    DmXLogger.Log($"Make Required Event Available for definition device {definitionDeviceId} with orgId : {orgId}", LogLevel.Debug);
    MakeRequiredEventsAvailalble(
        authenticationToken, ...
```

Use this to find potential parameter ids from log error

```
DECLARE @deviceKeys TABLE (
    deviceKey INT
);
INSERT INTO @deviceKeys (deviceKey)
```

```
(
    SELECT DeviceKey
    FROM definition.Devices
    WHERE DeviceID IN (5695480675736568992, -9084522846345424310, -2231271351705479521, -2677973217952751564) --<<<-- replace with potential
deviceids from log just before error
)

--      Return Parameters (13)
SELECT DISTINCT ParameterId FROM definition.Parameters
    WHERE ParameterKey IN
        (SELECT ReturnParameterKey FROM definition.Events WITH (NOLOCK)
            WHERE EventKey IN
                (SELECT EventKey FROM [definition].[DeviceEventDependencies] ddep WITH (NOLOCK)
                    INNER JOIN @deviceKeys dk ON dk.deviceKey = ddep.deviceKey))
    OR ParameterKey IN (
--      Event Condition Parameters ()
        SELECT DISTINCT ParameterKey FROM library.EventConditions WITH (NOLOCK)
            WHERE LibraryKey IN (-1)
            AND EventKey IN
                (SELECT EventKey FROM [definition].[DeviceEventDependencies] ddep WITH (NOLOCK)
                    INNER JOIN @deviceKeys dk ON dk.deviceKey = ddep.deviceKey))
```

4) A good Log to use for the above, without code debugging

Log

https://app.logz.io/#/dashboard/osd/discover/?\_a=(columns:!(message,AppName),filters:!(('\$state':(store:appState),meta:(alias:!\n,disabled:!f,index:'logzioCustomerIndex\*',key:Env,negate:!f,params:(query:DEV),type:phrase),query:(match\_phrase:(Env:DEV))),('\$state':(store:appState),meta:(alias:!\n,disabled:!f,index:'logzioCustomerIndex\*',key:message,negate:!f,params:!(oem,'Required%20parameter','All%20required%20parameters%20for%20this%20event%20have%20not%20been%20satisfied','Make%20Required%20Event%20Available%20for%20definition%20device','All%20required%20parameters%20for%20this%20event%20have%20not%20been%20satisfied%20-%20ParameterId'),type:phrases,value:'oem,%20Required%20parameter,%20All%20required%20parameters%20for%20this%20event%20have%20not%20been%20satisfied,%20Make%20Required%20Event%20Available%20for%20definition%20device,%20All%20required%20parameters%20for%20this%20event%20have%20not%20been%20satisfied%20-%20ParameterId'),query:(bool:(minimum\_should\_match:1,should:!(match\_phrase:(message:oem)),match\_phrase:(message:'Required%20parameter')),match\_phrase:(message:'All%20required%20parameters%20for%20this%20event%20have%20not%20been%20satisfied')),match\_phrase:(message:'Make%20Required%20Event%20Available%20for%20definition%20device')),match\_phrase:(message:'All%20required%20parameters%20for%20this%20event%20have%20not%20been%20satisfied%20-%20ParameterId'))))))),index:'logzioCustomerIndex\*',interval:auto,query:(language:lucene,query:''),sort:!(('@timestamp',desc))&\_g=(filters:!(,refreshInterval:(pause:!t,value:0),time:(from:'2023-05-08T09:00:00.000Z',to:now))&accountIds=true&switchToAccountId=157279