DATABAR production job auto-grouping - Integration Tests - DATABAR existing grouping check

Description

IMPORTANT

Before starting this work, please read the description of the epic VIS-3787: DATABAR In-Production batches will be grouped automatically AWAITING TRIAGE to get a full understanding of the background to the changes being made

This work follows directly on from VIS-4055.

Existing Integration Tests

Take a look at the class **Vision.Api.DotNet.Tests.Integration.Builder.IBARProduction.Production.CreateProductionJobTests.cs**.

These tests are used to test the process of creating production jobs.

To begin with, take a look at the existing test methods in the class to familarise yourself with the basic structure of the tests, and the helper methods they are using.

Task Details

This task involves writing five new integration tests to test the functionality implemented in the previous tasks in this epic.

These tests will specifically focus on **checking for existing groups** when new production jobs are created.

Use the same branch of the API created in VIS-3672: DATABAR production job auto-grouping - DATABAR grouping change READY TO DEPLOY for this work.

Write the following new integration tests in the **CreateProductionJobTests** class.

1) CreateDatabarJobAddsToExistingUndeliveredProductionJob

This will be an integration test to do the following.

- **Initial setup**
- Create one production job, sending in a collection of DATABAR components

    - There should be four components in total
        - 2 x **DATABAR Busway Length ST1**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
        - 2 x **DATABAR Busway Length ST2**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
    - The components should all have the same works order, as per the tests created in VIS-4055.

- All four components should be sent into the **CreateProductionJob** method together.

 2 production jobs should be created at this point.

- one for the DATABAR Busway Length ST1 components
- one for the DATABAR Busway Length ST2 components

 Core Test

 Now attempt to create another production job, sending in a collection of DATABAR components

- There should be three components in total
    - 1 x **DATABAR Busway Length ST1**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
    - 1 x **DATABAR Busway Length ST2**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
    - 1 x **DATABAR Busway Length ST3**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
- The components should all have the same works order as the four components created in the **Initial setup** phase, this can be achieved as per the instructions in VIS-4055.

    - All three components should be sent into the **CreateProductionJob** method together.
- The test should succeed if 1 new production job is created at this stage…
    - one for the DATABAR Busway Length ST3 component
- …and if the following components have been added to the existing production jobs created in the **Initial setup** phase.
    - the DATABAR Busway Length ST1 component should be added to the existing production job for that component type
    - the DATABAR Busway Length ST2 component should be added to the existing production job for that component type

Make sure to add any generated Production Job Ids to the **ProductionJobIds** collection, so these will be torn down in the base class tear down method

2) CreateDatabarJobAddsToExistingUndeliveredProductionJobForMultipleWorksOrdersAndM0s

This will be an integration test similar to the previous one, but includes multiple works order and M0s.

- **Initial setup**
- Create one production job, sending in a collection of DATABAR components
    - There should be eight components in total, across two works orders, and with multiple M0s
    - **Works Order 1**
        - 1 x **DATABAR Busway Length ST1**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
        - 1 x **DATABAR Busway Length ST1**; M0: **D02**; Market Sector: **DATABAR**; Rating: **800**
        - 1 x **DATABAR Busway Length ST2**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
        - 1 x **DATABAR Busway Length ST2**; M0: **D02**; Market Sector: **DATABAR**; Rating: **800**
    - **Works Order 2**
        - 1 x **DATABAR Busway Length ST1**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
        - 1 x **DATABAR Busway Length ST1**; M0: **D02**; Market Sector: **DATABAR**; Rating: **800**
        - 1 x **DATABAR Busway Length ST2**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
        - 1 x **DATABAR Busway Length ST2**; M0: **D02**; Market Sector: **DATABAR**; Rating: **800**
    - The components should all have the same works order, as per the tests created in VIS-4055.
    - All eight components should be sent into the **CreateProductionJob** method together.
- Eight production jobs should be created at this point.
    - one for each unique combination of **Works Order** / **M0** / **component type**
- **Core Test**
- Now attempt to create another production job, sending in a collection of DATABAR components
    - There should be three components in total
    - **Works Order 2**
        - 1 x **DATABAR Busway Length ST1**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
        - 1 x **DATABAR Busway Length ST2**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
        - 1 x **DATABAR Busway Length ST3**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**

    - All three components should be sent into the **CreateProductionJob** method together.
- The test should succeed if 1 new production job is created at this stage…
    - one for the DATABAR Busway Length ST3 component
- …and if the following components have been added to the existing production jobs created in the **Initial setup** phase.
    - the DATABAR Busway Length ST1 component should be added to the existing production job for that works order, M0 and component type (in this case, **Works Order 2**, **D01** and **DATABAR Busway Length ST1**)
    - the DATABAR Busway Length ST2 component should be added to the existing production job for that works order, M0 and component type (in this case, **Works Order 2**, **D01** and **DATABAR Busway Length ST2**)

Make sure to add any generated Production Job Ids to the **ProductionJobIds** collection, so these will be torn down in the base class tear down method

3) CreateDatabarJobDoesNotAddToExistingDeliveredProductionJob

This will be an integration test similar to the previous one, but involves a production job that is fully delivered, and so we will not expect the code to add any other components to.

- **Initial setup**
- Create one production job, sending in one DATABAR component
    - There should be one components in total
        - 1 x **DATABAR Busway Length ST1**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
- One production job should be created at this point.
- The created **ProductionJob** object should have an associated **ProductionJobSummary** record, with the following property values set
    - NumberOfItemsDeliveryApplicable: **1**
    - NumberOfItemsDeliveryComplete: **0**
- Modify the **ProductionJobSummary** object to set the following property, then use NHibernate to save it.
    - NumberOfItemsDeliveryComplete: **1**
- **Core Test**
- Now attempt to create another production job, sending in one DATABAR component
    - There should be one components in total
        - 1 x **DATABAR Busway Length ST1**; M0: **D01**; Market Sector: **DATABAR**; Rating: **800**
    - The component should have the same works order as the previous component created in the **Initial setup** phase.
- The test should succeed if a new production job is created at this stage. This is because, as the previous production job created has a summary object indicating it has now been fully delivered, the code should not add these components to that production job, and should instead create a new one.

Make sure to add any generated Production Job Ids to the **ProductionJobIds** collection, so these will be torn down in the base class tear down method

4) CreateIbarJobDoesNotAddToExistingUndeliveredProductionJob

This will be an integration test similar to the previous one, but involves an IBAR component instead of a DATABAR one.

- **Initial setup**
- Create one production job, sending in one IBAR component
    - There should be one components in total
        - 1 x **Straight Feeder**; M0: **B01**; Market Sector: **IBAR**; Rating: **2500**
- One production job should be created at this point.
- **Core Test**
- Now attempt to create another production job, sending in one IBAR component
    - There should be one components in total
        - 1 x **Straight Feeder**; M0: **B01**; Market Sector: **IBAR**; Rating: **2500**
    - The component should have the same works order as the previous component created in the **Initial setup** phase.
- The test should succeed if a new production job is created at this stage. This is because the code to append components to existing production jobs should only apply to DATABAR jobs, not IBAR.

Make sure to add any generated Production Job Ids to the **ProductionJobIds** collection, so these will be torn down in the base class tear down method

5) CreateResinbarJobDoesNotAddToExistingUndeliveredProductionJob

This will be an integration test similar to the previous one, but involves a RESINBAR component instead of an IBAR one.

- **Initial setup**
- Create one production job, sending in one RESINBAR component
    - There should be one components in total
        - 1 x **Straight Feeder**; M0: **B01**; Market Sector: **RESINBAR**; Rating: **2500**
- One production job should be created at this point.
- **Core Test**
- Now attempt to create another production job, sending in one RESINBAR component
    - There should be one components in total
        - 1 x **Straight Feeder**; M0: **B01**; Market Sector: **RESINBAR**; Rating: **2500**
    - The component should have the same works order as the previous component created in the **Initial setup** phase.
- The test should succeed if a new production job is created at this stage. This is because the code to append components to existing production jobs should only apply to DATABAR jobs, not RESINBAR.

Make sure to add any generated Production Job Ids to the **ProductionJobIds** collection, so these will be torn down in the base class tear down method