Preparing your Generic Integration Module integration for multi-dimensional product sizing

Preparation, Other Notes

- May backup existing GIM database just in case, but structure wise, only 1 new table is added
- Make a backup copy of the existing GIM files, particularly the config file appconfig.json within the GIM files, folder should be located in C:\RepSparkDaemon\
- take note of the steps in the next bullet point, but do be aware those steps do not need to be taken until UAT testing is complete, and you are ready to use the updated version on Production
- At some point you will want to stop the existing GIM that is running. You will need to disable the task that launches the GIM when the computer starts from the Windows Task Scheduler.
  - this can be done in 2 different ways, one is using the bat script, but would recommend just doing the following:
    - go to Windows Task Scheduler, find RepSpark Daemon task, then stop the task
  - Then after this is done end the process RepSpark.Integration.Program.exe in the Windows Task Manager to stop the GIM from running

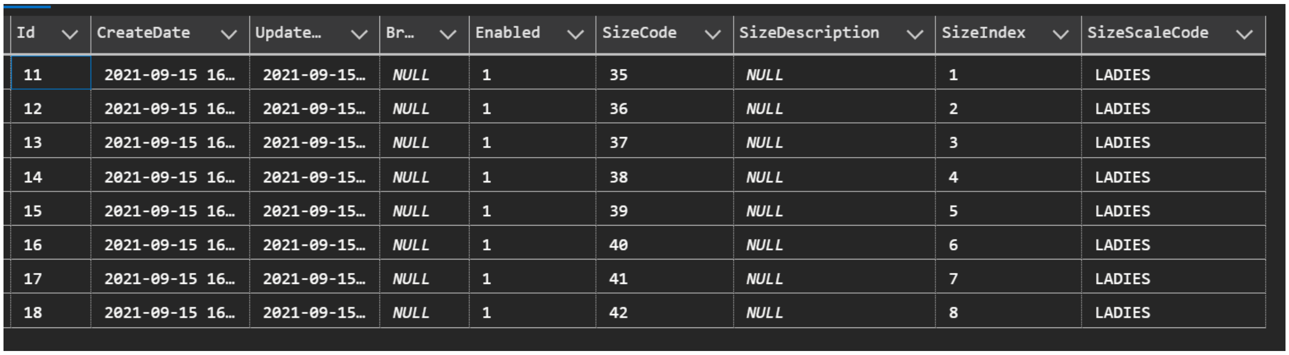
Preparing the Database

- create database table to be used by the GIM (GIM forms the data after pulling from the tables, and sends it to RSK API)
  - create the SizeTypes table in the proper GIM database in MS SQL Server using the following script command

  -
    ```
    CREATE TABLE [SizeTypes] (
        [Id] int NOT NULL IDENTITY,
        [SizeTypeCode] nvarchar(15) NULL,
        [SizeTypeValue] nvarchar(120) NULL,
        [SizingId] int NULL,
        CONSTRAINT [PK_SizeTypes] PRIMARY KEY ([Id]),
        CONSTRAINT [FK_SizeTypes_Sizings_SizingId] FOREIGN KEY ([SizingId])
    REFERENCES [Sizings] ([Id]) ON DELETE CASCADE
    );
    ```

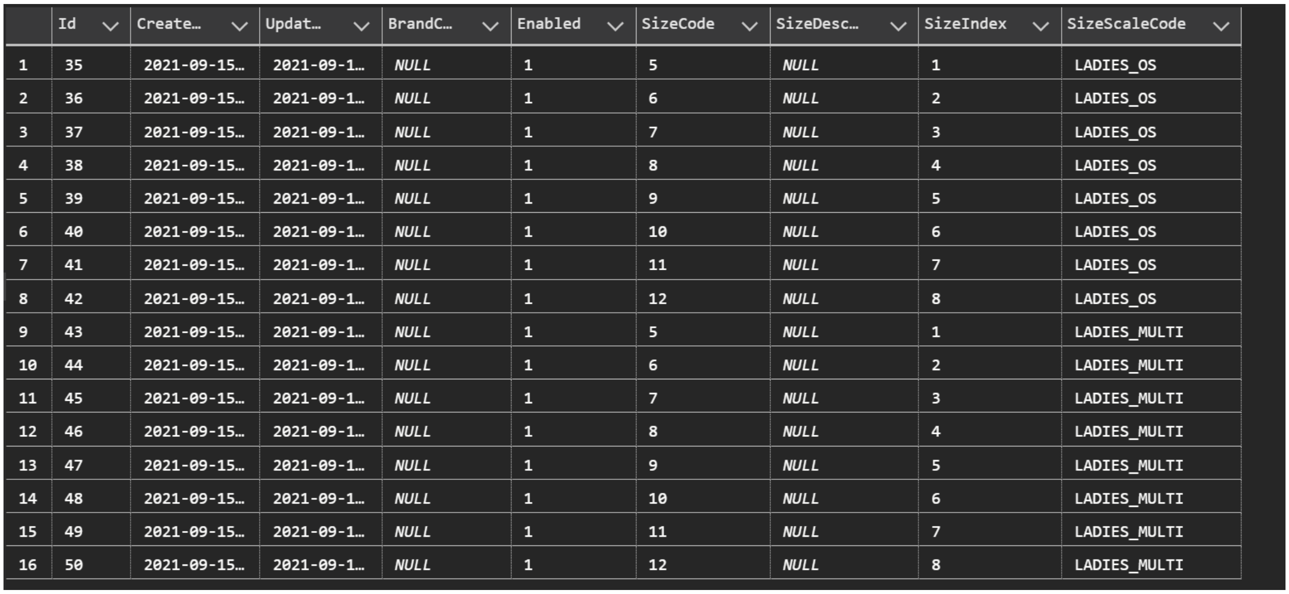
Populating the data

- Currently, your sizing data in the existing Sizings table probably looks something like this for reference



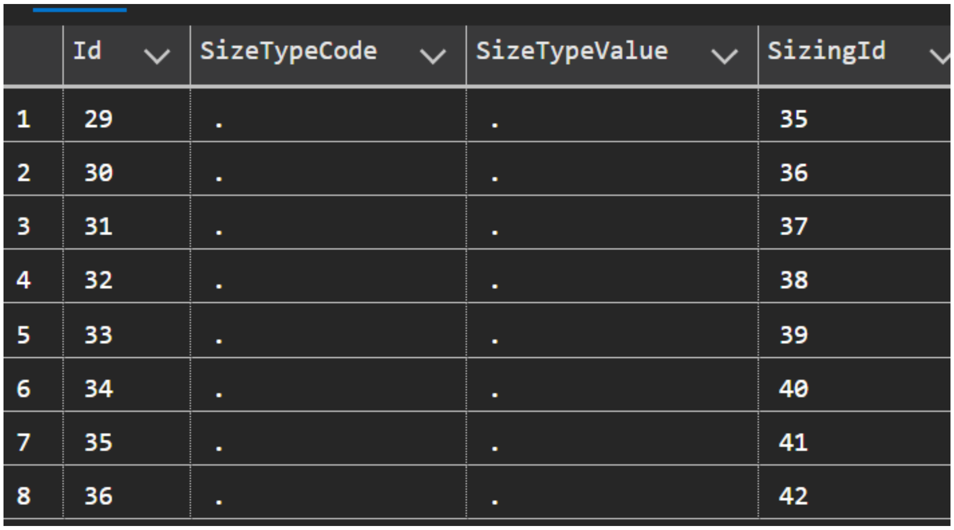
Id	CreateDate	Update…	Br…	Enabled	SizeCode	SizeDescription	SizeIndex	SizeScaleCode
11	2021-09-15 16…	2021-09-15…	NULL	1	35	NULL	1	LADIES
12	2021-09-15 16…	2021-09-15…	NULL	1	36	NULL	2	LADIES
13	2021-09-15 16…	2021-09-15…	NULL	1	37	NULL	3	LADIES
14	2021-09-15 16…	2021-09-15…	NULL	1	38	NULL	4	LADIES
15	2021-09-15 16…	2021-09-15…	NULL	1	39	NULL	5	LADIES
16	2021-09-15 16…	2021-09-15…	NULL	1	40	NULL	6	LADIES
17	2021-09-15 16…	2021-09-15…	NULL	1	41	NULL	7	LADIES
18	2021-09-15 16…	2021-09-15…	NULL	1	42	NULL	8	LADIES

- if you want to use multiple dimensions for a Size Scales (e.g. shoe widths), it is best to create a separate SizeScale/Code like so, while keeping in mind SizeScaleCode is limited to 15 characters:



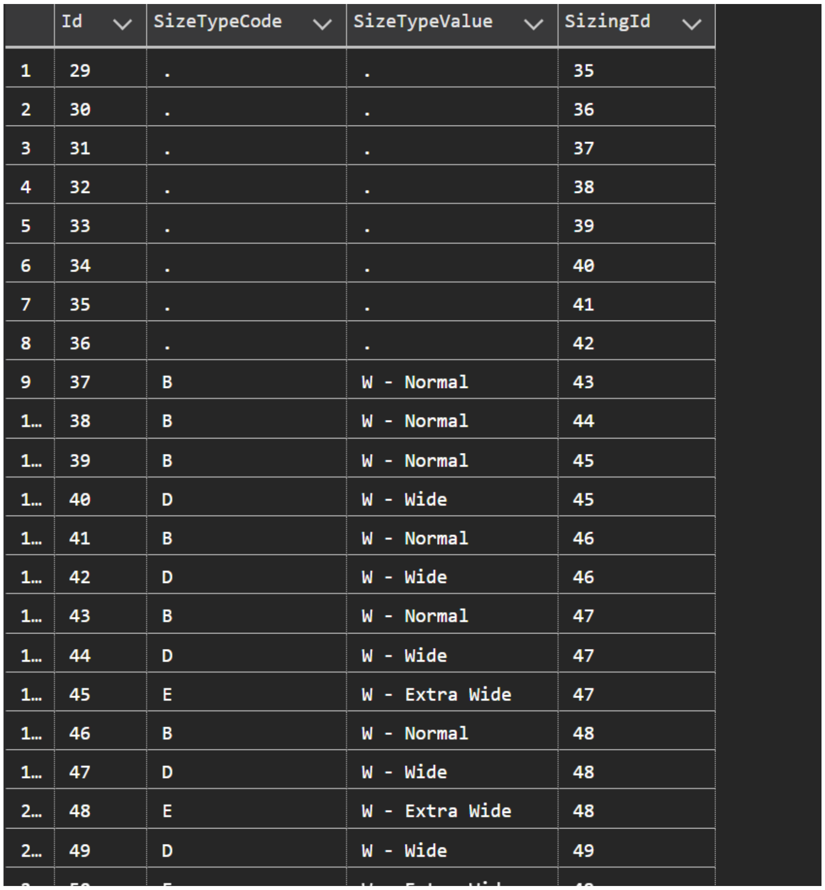
	Id	Create…	Updat…	BrandC…	Enabled	SizeCode	SizeDesc…	SizeIndex	SizeScaleCode
1	35	2021-09-15…	2021-09-1…	NULL	1	5	NULL	1	LADIES_OS
2	36	2021-09-15…	2021-09-1…	NULL	1	6	NULL	2	LADIES_OS
3	37	2021-09-15…	2021-09-1…	NULL	1	7	NULL	3	LADIES_OS
4	38	2021-09-15…	2021-09-1…	NULL	1	8	NULL	4	LADIES_OS
5	39	2021-09-15…	2021-09-1…	NULL	1	9	NULL	5	LADIES_OS
6	40	2021-09-15…	2021-09-1…	NULL	1	10	NULL	6	LADIES_OS
7	41	2021-09-15…	2021-09-1…	NULL	1	11	NULL	7	LADIES_OS
8	42	2021-09-15…	2021-09-1…	NULL	1	12	NULL	8	LADIES_OS
9	43	2021-09-15…	2021-09-1…	NULL	1	5	NULL	1	LADIES_MULTI
10	44	2021-09-15…	2021-09-1…	NULL	1	6	NULL	2	LADIES_MULTI
11	45	2021-09-15…	2021-09-1…	NULL	1	7	NULL	3	LADIES_MULTI
12	46	2021-09-15…	2021-09-1…	NULL	1	8	NULL	4	LADIES_MULTI
13	47	2021-09-15…	2021-09-1…	NULL	1	9	NULL	5	LADIES_MULTI
14	48	2021-09-15…	2021-09-1…	NULL	1	10	NULL	6	LADIES_MULTI
15	49	2021-09-15…	2021-09-1…	NULL	1	11	NULL	7	LADIES_MULTI
16	50	2021-09-15…	2021-09-1…	NULL	1	12	NULL	8	LADIES_MULTI

- once you decide to use multiple dimensions, the new SizeTypes table needs to be updated to support each size row in Sizing
- there will be a new SizeType row per row in Sizing
  - SizingId corresponds to the Id of the parent row in the Sizing table
  - for classic single style dimension sizes, please add a placeholder row with period where values "." in SizeTypeCode and SizeTypeValue like the screenshot below



	Id	SizeTypeCode	SizeTypeValue	SizingId
1	29	.	.	35
2	30	.	.	36
3	31	.	.	37
4	32	.	.	38
5	33	.	.	39
6	34	.	.	40
7	35	.	.	41
8	36	.	.	42

- for new multi-dimension sizes, there will need to be a new SizeType row representing each secondary dimension (e.g. the shoe widths, pants inseams) per main dimension (standard shoe size, pants lengths). SizeTypeCode is **limited to 15 characters**. SizeTypeCode will be a component of the eventual SizeCode that is used, while SizeTypeValue is a description field and is the label that shows up on the RepSpark products page, but SizeTypeCode and SizeTypeValue can also simply be set to the same value for simplicity.
- thus SizeTypes might look like the following if there were 3 possible sizes for women’s shoe width as the secondary dimension (B for normal, D for wide, and 2E for extra wide)



	Id	SizeTypeCode	SizeTypeValue	SizingId
1	29	.	.	35
2	30	.	.	36
3	31	.	.	37
4	32	.	.	38
5	33	.	.	39
6	34	.	.	40
7	35	.	.	41
8	36	.	.	42
9	37	B	W - Normal	43
1…	38	B	W - Normal	44
1…	39	B	W - Normal	45
1…	40	D	W - Wide	45
1…	41	B	W - Normal	46
1…	42	D	W - Wide	46
1…	43	B	W - Normal	47
1…	44	D	W - Wide	47
1…	45	E	W - Extra Wide	47
1…	46	B	W - Normal	48
1…	47	D	W - Wide	48
2…	48	E	W - Extra Wide	48
2…	49	D	W - Wide	49

  - next, products should be updated to use the newly created SizeScaleCodes, and inventory entries should be adjusted to match these new size codes

Changes to inventory

- the Size Code format which was previous just *dim1SizeCode* needs to be converted to support multi dimension's format:
  - *dim1SizeCode_dim2SizeCode*
- so single dimension products should use a sizecode like "12_.", "L_." instead of "12" and "L"
- to specify the sizecodes for multi dimension products in inventory rows, use the SizeTypeCode for the second dimension, so shoe size 11 might have the 3 codes 11_B, 11_D, and 11_E

To test in UAT

- please have credentials for UAT ready
- Copy the new, unzipped files to C: drive, maybe naming the folder to *C:\RepSparkDaemonUAT\* and the .exe within to *C:\RepSparkDaemonUAT\RepSpark.Integration.Program2.exe* (so it can be ran while the existing process named *RepSpark.Integration.Program.exe* is running)

- change the appsettings.json within the UAT folder's directory
- modify the clientkey, devtoken, and ApiUrl to the appropriate UAT values
- potentially db credentials if a different clone of the GIM database isused (optional)
- change the Schedules in appsettings.json so that the Sizing sync can run when the program is started

  -
    ```
    "Schedules": [
     {
      "StartTime": "4:30",
      "RepeatEvery": "1",
      "Interval": "Now",
      "SyncMode": "Full",
      "Entity": "Sizing"
     }
    ],
    ```
- after all the necessary database changes are made following instructions in previous sections
- run C:\RepSparkDaemonUAT\RepSpark.Integration.Program2.exe which should immediately began populating the data
- to run the other syncs
  - stop the currently running process RepSpark.Integration.Program2.exe if it is running using Task Manager
  - change the Entity in the Schedules in appsettings.json to Product then run the exe, wait a bit, then stop the currently active process RepSpark.Integration.Program2.exe again using Task Manager
  - change Entity again to Inventory, then run the exe
- Verify the data is correct on UAT, then stop the running process

AFTER UAT testing is complete and everything looks fine, Install for Production

- have stopped the UAT process
- stop the scheduled task using Windows Task Scheduler based on instructions in first section of document
- stop the currently running process RepSpark.Integration.Program.exe
- rename the existing GIM directory C:\RepSparkDaemon\ to maybe C:\RepSparkDaemonBackup\. This will be the backup files in case
- copy the new version of the GIM files to C: and just rename the publish folder to RepSparkDaemon, so new files should look like C:\RepSparkDaemon\
- as everything should be pointing to PRODUCTION now that testing is complete on UAT, use the **existing appsettings.json** config with existing schedule in the **old** folder (now named C:\RepSparkDaemonBackup\), copy this config to the new C:\RepSparkDaemon\ folder, overwriting the placeholder config

- double click run C:\RepSparkDaemon\RepSpark.Integration.Program.exe to start the process
- re-enable the scheduled task in Task Scheduler