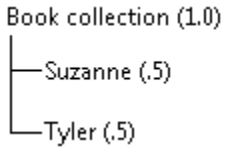
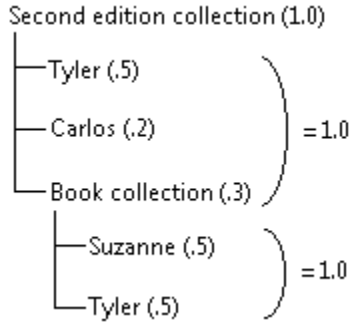
In MDS, you can set weight values on members in a collection, so that certain members count more than others when you do analysis in a subscribing system.

For example, in this sample collection that is used to determine royalties for our MDS book, Tyler (my co-author) and I are equals. The collection contains both of us, and we both get half of the massive royalties we're expecting.



Imagine there is a second edition of the book, and we decide that Carlos, our technical reviewer, will take over my part of the book. I'll still get some royalties, but not the full amount.



You can use this collection to accurately calculate what's owed to Carlos, Tyler, and me. In general, for each member in the collection, you specify an integer. And you want every level in your collection to equal 1.00.

Setting Weight Values

To set weight values in MDS, you open the collection (CN) table for the entity and update the weight value for each member in the collection. You can do this manually or by using a stored procedure. When you create a subscription view with the Collections format, a Weight column includes the values you set. This view is used by subscribing systems to consume the data.

Security prerequisites

To manually set weight values, you must have access to the MDS database. At a minimum, you should be able to read these tables:

- mdm.tblModel
- mdm.tblEntity
- A collection table (ending in CN) for the entity that's enabled for collections

And you must be able to update the collection table that contains the weight values (a table that ends in CM).

If you're going to use the stored procedure, you must also be able to read mdm.tblUser.

How to set the weight values manually

To find the collection table where you will update the weight value, complete the following steps:

1. Open mdm.tblModel. Note the model's ID in the ID column.
2. Open mdm.tblEntity. Find the entity that's enabled for (use the model ID in the Model_ID column for assistance). The names of the collection tables are displayed in this row.
   The model and entity IDs are combined to determine the name of the collection tables. For example, if the model ID is 4 and the entity ID is 12, the collection tables are tbl_4_12_CN and tbl_4_12_CM.

	ID	MUID	Model_ID	Name	EntityTable	SecurityTable	HierarchyTable	HierarchyParentTable	CollectionTable	CollectionMemberTable
11	12	D6C5E1DE-7A26-4B17-8...	4	Product	tbl_4_12_EN	tbl_4_12_MS	tbl_4_12_HR	tbl_4_12_HP	tbl_4_12_CN	tbl_4_12_CM

3. Open the CN table, which lists all collections for the entity. Based on the value in the Name column, note the ID in the ID column.

	Version_ID	ID	VersionMember_ID	Status_ID	ValidationStatus_ID	Name	Code	Description	Owner_ID	EnterDTM	EnterUserID	EnterVersionID
1	4	1	1	1	3	Collection 1	Col1	NULL	1	2011-02-09 22:39:13.503	1	4
2	4	2	2	1	3	Collection 2	Col2	NULL	1	2011-02-09 22:39:26.850	1	4

4. Open the CM table, which lists all members in the collection. In this table, in the Parent_CN_ID column, the ID of the collection (from step 3) is listed. In this example, we can see that there are only two members in the collection with an ID of 1 in the Parent_CN_ID column.

	Version_ID	ID	Status_ID	Parent_CN_ID	ChildType_ID	Child_EN_ID	Child_HP_ID	Child_CN_ID	SortOrder	Weight	EnterDTM
1	4	1	1	1	1	1	NULL	NULL	1	1.000	2011-02-09 22:40:46.893
2	4	2	1	1	1	2	NULL	NULL	2	1.000	2011-02-09 22:40:52.313

5. Update the value in the Weight column for each member. Remember that each level in your collection should equal 1.

How to set the weight values by using a stored procedure

Step 1: Create stored procedure called udpCollectionMemberAddWithWeight

Paste the following text into a new SQL query window. Execute to create the stored procedure.

```
USE [DATABASENAME]
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [mdm].[udpCollectionMemberAddWithWeight]
@User_ID INT, @Version_ID INT, @Entity_ID INT, @Collection_ID INT, @Child_ID
INT, @ChildType_ID TINYINT, @WEIGHT Decimal(10,3), @Remove TINYINT=0
WITH EXECUTE AS N'mds_schema_user'
AS
BEGIN
      SET NOCOUNT ON;
      DECLARE @TranCounter INT,
            @ErrorMessage NVARCHAR(4000);
      SET @TranCounter = @@TRANCOUNT;
      IF @TranCounter > 0 SAVE TRANSACTION TX;
    ELSE BEGIN TRANSACTION;
      BEGIN TRY
            DECLARE @SQL                              AS NVARCHAR(MAX),
                  @CollectionMemberTableName    AS sysname,
                  @CollectionTableName          AS sysname,
                  @Member_ID                    AS INT,
                  @RecordExists                       AS BIT;

      --Get Collection Member table name
      SELECT @CollectionMemberTableName = mdm.udfTableNameGetByID(@Entity_ID,
5);
            IF @Remove = 1 BEGIN--Delete the existing record
                        SET @SQL = N'
                        DELETE FROM mdm.' +
quotename(@CollectionMemberTableName) + N'
                        WHERE Version_ID = ' + CONVERT(NVARCHAR(30),
@Version_ID) + N'
                              AND ' + CASE @ChildType_ID WHEN 1 THEN
N'Child_EN_ID' WHEN 2 THEN N'Child_HP_ID'
                                    WHEN 3 THEN N'Child_CN_ID' END + N' = ' +
                              CONVERT(NVARCHAR(30), @Child_ID) + N'
                              AND ChildType_ID = ' + CONVERT(NVARCHAR(30),
@ChildType_ID) + N'
                              AND Parent_CN_ID = ' + CONVERT(NVARCHAR(30),
@Collection_ID) + N'
                              AND Status_ID = 1;';
                  EXEC sp_executesql @SQL;

      END ELSE  BEGIN --Create the record
                  --Validate @Collection_ID
                  IF @Collection_ID IS NULL
                  BEGIN
                        SELECT  @ErrorMessage =
mdm.udfDBErrorsGetMessageByIDLanguage(100010, @@LANGID, OBJECT_NAME(@@PROCID) +
'%Collection_ID');
                        RAISERROR  100010 @ErrorMessage;
                        RETURN(100010);
                  END;--if

                  --Get Collection table name
                  SELECT @CollectionTableName =
mdm.udfTableNameGetByID(@Entity_ID, 3);
                  --Check to see if a record with the Collection_ID exists
                  SET @SQL = N'
                        SET @RecordExists = 0;
```

```
                        IF EXISTS (
                              SELECT 1 FROM mdm.' +
quotename(@CollectionTableName) + N'
                              WHERE ID = ' + CONVERT(NVARCHAR(30),
@Collection_ID) + N'
                              AND Version_ID = ' + CONVERT(NVARCHAR(30),
@Version_ID) + N'
                        ) SET @RecordExists = 1;';

                  EXEC sp_executesql @SQL, N'@RecordExists BIT OUTPUT',
@RecordExists OUTPUT;

                  IF @RecordExists = 0
                  BEGIN
                        SELECT  @ErrorMessage =
mdm.udfDBErrorsGetMessageByIDLanguage(100010, @@LANGID, OBJECT_NAME(@@PROCID) +
'%Collection_ID');
                        RAISERROR  100010 @ErrorMessage;
                        RETURN(100010);
                  END; --if

                  --Validate @ChildType_ID

                  IF (@ChildType_ID IS NULL) OR (@ChildType_ID < 1 OR
@ChildType_ID > 3)
                  BEGIN
                        SELECT  @ErrorMessage =
mdm.udfDBErrorsGetMessageByIDLanguage(100010, @@LANGID, OBJECT_NAME(@@PROCID) +
'%ChildType_ID');
                        RAISERROR  100010 @ErrorMessage;
                        RETURN(100010);
                  END; --if

                  --Validate @Child_ID
                  --Check to see if a record with the Child_ID exists
                  SET @SQL = N'
                        SET @RecordExists = 0;
                        IF EXISTS (
                              SELECT 1 FROM mdm.' +
quotename(mdm.udfTableNameGetByID(@Entity_ID, @ChildType_ID)) + N'
                              WHERE ID = ' + CONVERT(NVARCHAR(30), @Child_ID) +
N'
                              AND Version_ID = ' + CONVERT(NVARCHAR(30),
@Version_ID) + N'
                        ) SET @RecordExists = 1;';

                  EXEC sp_executesql @SQL, N'@RecordExists BIT OUTPUT',
@RecordExists OUTPUT;

                  IF @RecordExists = 0
                  BEGIN
                        SELECT  @ErrorMessage =
mdm.udfDBErrorsGetMessageByIDLanguage(100010, @@LANGID, OBJECT_NAME(@@PROCID) +
'%Child_ID');
                        RAISERROR  100010 @ErrorMessage;
                        RETURN(100010);
                  END; --if
```

```
                  --Insert into the Correct Collection Member Table
                  SET @SQL = N'
                        INSERT INTO mdm.' +
quotename(@CollectionMemberTableName) + N'
                        (
                              Version_ID,
                              Status_ID,
                              Parent_CN_ID,
                              ChildType_ID,
                              Child_EN_ID,
                              Child_HP_ID,
                              Child_CN_ID,
                              SortOrder,
                              Weight,
                              EnterDTM,
                              EnterUserID,
                              EnterVersionID,
                              LastChgDTM,
                              LastChgUserID,
                              LastChgVersionID
                        ) SELECT
                              ' + CONVERT(NVARCHAR(30), @Version_ID) + N',
                              1, --Status
                              ' + ISNULL(CONVERT(NVARCHAR(30),
NULLIF(@Collection_ID, 0)), N'NULL') + N', --Parent_CN_ID
                              ' + CONVERT(NVARCHAR(30),@ChildType_ID) + N', --
ChildType_ID
                              ' + CASE @ChildType_ID WHEN 1 THEN
CONVERT(NVARCHAR(30), @Child_ID) ELSE N'NULL' END + N', --Child_EN_ID
                              ' + CASE @ChildType_ID WHEN 2 THEN
CONVERT(NVARCHAR(30), @Child_ID) ELSE N'NULL' END + N', --Child_HP_ID
                              ' + CASE @ChildType_ID WHEN 3 THEN
CONVERT(NVARCHAR(30), @Child_ID) ELSE N'NULL' END + N', --Child_CN_ID
                              ISNULL(MAX(ID), 0) + 1,
                              @WEIGHT,
                              GETUTCDATE(),
                              ' + CONVERT(NVARCHAR(30),@User_ID) + N',
                              ' + CONVERT(NVARCHAR(30),@Version_ID) + N',
                              GETUTCDATE(),
                              ' + CONVERT(NVARCHAR(30),@User_ID) + N',
                              ' + CONVERT(NVARCHAR(30),@Version_ID) + N'

                        FROM
                              mdm.' + quotename(@CollectionMemberTableName) +
N';
                        SET @Member_ID = SCOPE_IDENTITY();';

                  EXEC sp_executesql @SQL, N'@Member_ID INT OUTPUT', @Member_ID
OUTPUT;

            END; --if

            --Commit only if we are not nested
            IF @TranCounter = 0 COMMIT TRANSACTION;
            RETURN(0);
```

```
      END TRY
      --Compensate as necessary
      BEGIN CATCH
            DECLARE @ErrorSeverity INT,
                        @ErrorState INT,
                        @Error INT;
            SELECT @ErrorMessage = ERROR_MESSAGE(), @ErrorSeverity =
ERROR_SEVERITY(), @ErrorState = ERROR_STATE(), @Error = @@ERROR;

            IF @TranCounter = 0 ROLLBACK TRANSACTION;
          ELSE IF XACT_STATE() <> -1 ROLLBACK TRANSACTION TX;
            RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
            --On error, return NULL results
            --SELECT @Return_ID = NULL;
            RETURN(1);
      END CATCH;
      SET NOCOUNT OFF;
END;
GO
```

Step 2: Run the stored procedure

The variables for the stored procedure are:

@User_ID – The ID column from mdm.tblUser.
@Version_ID – The Version_ID column from the CN table.
@Entity_ID – The ID column from mdm.tblEntity.
@Collection_ID – The ID column from the CN table.
@Child_ID - The Child_EN_ID column from the CM table.
@ChildType_ID – The ChildType_ID from the CM table.
@WEIGHT – An integer of your choice.

Example

```
EXEC [mdm].[udpCollectionMemberAddWithWeight]
@User_ID=1, @Version_ID=4, @Entity_ID=12, @Collection_ID=2,
@Child_ID=1,@ChildType_ID=1,@WEIGHT=.5
```

When you create a subscription view with the Collections format, a Weight column includes the values you set.