MDX Help Sheet

Everything You Ever Wanted to Know about MDX... *but were afraid to ask.*

MDX...a powerful and flexible way to query and manipulate Essbase data

Member – [Brackets go around a Member]
Tuple – Collection of Members indicating an Essbase intersection (Parenthesis go around a Tuple)
Set – Collection of Tuples {Curly Brackets go around a Set}

Basic Syntax

```
SELECT
        {([Year])} ON COLUMNS,          ← Column Set
        {([Jan])} ON ROWS               ← Row Set
FROM Sample.Basic                       ← Source Cube
WHERE ([East],[Cola],[Sales])           ← Where Clause / Slicer
```

Levels([Dimension], number) – all members of a dimension at a level number.
Generations([Dimension], number) – all members of a dimension at a generation number.
Members([Dimension]) – all members of a dimension.
Children([member]) – Return a member's children, non-inclusive.
{[member], **Children(**[member]**)}** - Return a member's children, inclusive.
Descendants() - Return a member's descendants (inclusive)

```
Descendants([Product])                          ← All descendants of Product including Product
Descendants([Colas], Levels([Product], 0))      ← All level 0 members under Colas
Descendants([Product],[Product].Level,AFTER)    ← All descendants excluding Product
```

Crossjoin({Set 1}, {Set 2}) ← Join two sets together

Order() – Sort the rows
By data – Order(Children([Product]), [Sales])
By metadata – Order(Children([Product]), [Product].CurrentMember.MEMBER_NAME)
Sort descending – Order(Children([Product]), [Sales], BDESC)

Filter()
By value – Filter(Children([Market]), Profit < 0)
By Boolean – Filter(Children([Market]), IsUDA([Market].CurrentMember, "Major Market"))
Suppress Shared Members –
Filter(Descendants([Products]),Not [Products].CurrentMember.SHARED_FLAG)

Generate() / Looping – For each Tuple in Set1, return Set2. Ex. For each child of [Product], return the member's first child. Note: Add curly brackets on the second argument.

```
Generate({Children([Product])},
        {FirstChild([Product].CurrentMember)})
```

Hierarchize() – Put a set of members in the same order as the outline. Add optional POST argument so children are listed before parents.

```
Hierarchize(Members([Product]), POST)   ← All Products with children listed before parents.
```

DIMENSION PROPERTIES – Add this keyword to return member properties.
List of properties: MEMBER_NAME, MEMBER_ALIAS, LEVEL_NUMBER, GEN_NUMBER, IS_EXPENSE, COMMENTS, MEMBER_UNIQUE_NAME, SHARED_FLAG (undocumented).
Example returns the Product level # on each row:

```
DIMENSION PROPERTIES [Product].LEVEL_NUMBER ON ROWS
```

Property_Expr()
Example returns all Product dimension members, parent and alias:

```
Members([Product])
DIMENSION PROPERTIES
Property_Expr([Product],MEMBER_NAME,Parent(CurrentAxisMember()),"Parent"),
Property_Expr([Product],MEMBER_ALIAS,CurrentAxisMember(),"Alias") ON ROWS
```

MDX ...a powerful and flexible way to query and manipulate Essbase data

Comments
/* Multi Line
Comment */
-- Single Line Comment

Suppressing Missing Values / Blocks
Use NON EMPTY keywords before the Set. Example:
NON EMPTY Descendants([Product]) on Rows ← eliminates any rows with all #MISSING
NONEMPTYBLOCK Descendants([Product]) on Rows ← can improve performance in BSO cubes

WITH MEMBER – Define a calculated member that can be used in the query.
UDA([Member Name], "UDA") – Selects all Essbase members with that UDA.
Sum() - Returns the sum of values of Tuples in a Set.
Example sums all Markets with Major Market UDA:

```
WITH MEMBER
        [Market].[Major Markets] as 'SUM(UDA([Market], "Major Market"))'
SELECT
        Children([Year]) ON COLUMNS,
        {[Major Markets]} ON ROWS
        WHERE ([Product],[Actual],[Sales])
```

WITH SET – Create reusable, custom sets.

```
WITH SET [New England] as '{[Connecticut],[New Hampshire],[Massachusetts]}'
SELECT {[New England]} ON COLUMNS,
        {[Sales]} ON ROWS
```

Shared Member Issues
Ex. Get all the children of a Set of Shared Members

```
Generate(Children([Alt Org Hier]),{Children(StrToMbr([Org].CurrentMember.MEMBER_NAME))})
```

INSERT – Copy data from one intersection to another

```
INSERT
        "([Year].[Jan])" TO "([Year].[Feb])"        ← Copy data from Jan to Feb
INTO [Sample].[Basic]
FROM(Select
                {[Jan]} on Columns,
                {[Connecticut]} on Rows
        FROM [Sample].[Basic]
        Where ([Actual],[Sales],[Cola]))
```

ASO Member Formula examples using **IIF()** and **CASE**
Only ASO member formulas use MDX. They are expressions that return a value.
Example to create a Measure to calculate average sales per month in a quarter:

```
IIF(IsLevel([Time].CurrentMember,0),
        Avg(Siblings([Time].CurrentMember),[Sales]),MISSING)
```

Same example but this works for the Quarter level members too:

```
CASE [Time].CurrentMember.LEVEL_NUMBER
        WHEN 0
                THEN Avg(Siblings([Time].CurrentMember),[Sales])
        WHEN 1
                THEN Avg(Children([Time].CurrentMember),[Sales])
        ELSE MISSING
END
```

Server Export – OVERWRITE and COLUMNDELIMITER arguments are optional.

```
EXPORT INTO FILE "filename" OVERWRITE USING COLUMNDELIMITER "delimiter"
SELECT…
```

Local Export – run the following command in MaxL before MDX statement for delimited export.

```
set column_separator "delimiter";
```