

Building Dimensions Dynamically



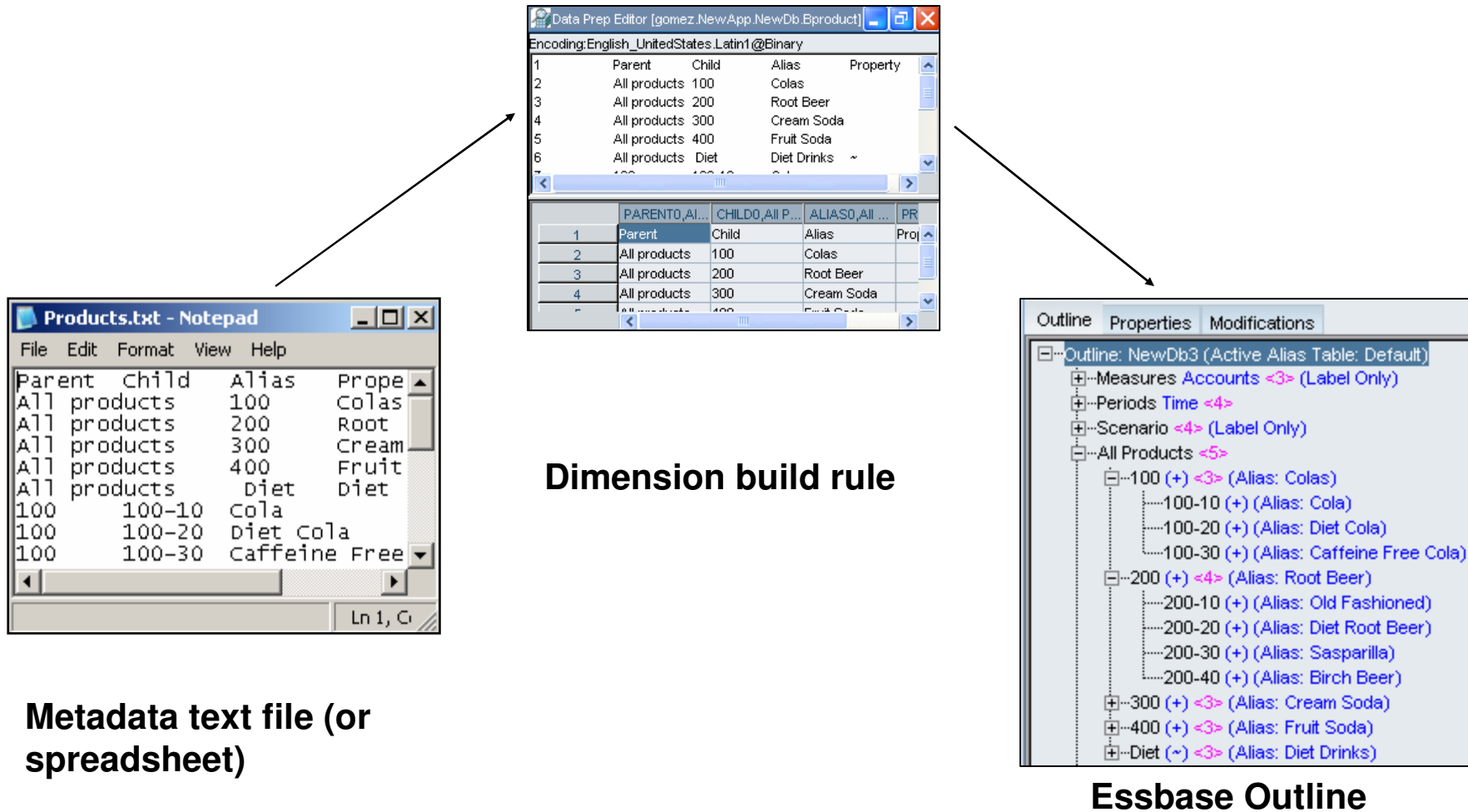
MARKETING TECHNOLOGIES GROUP
New York Information Technology Center:
55 Broad Street, 10th Floor, New York, NY 10004

[Marketing Technologies Group](#)

Dynamic Dimension Building

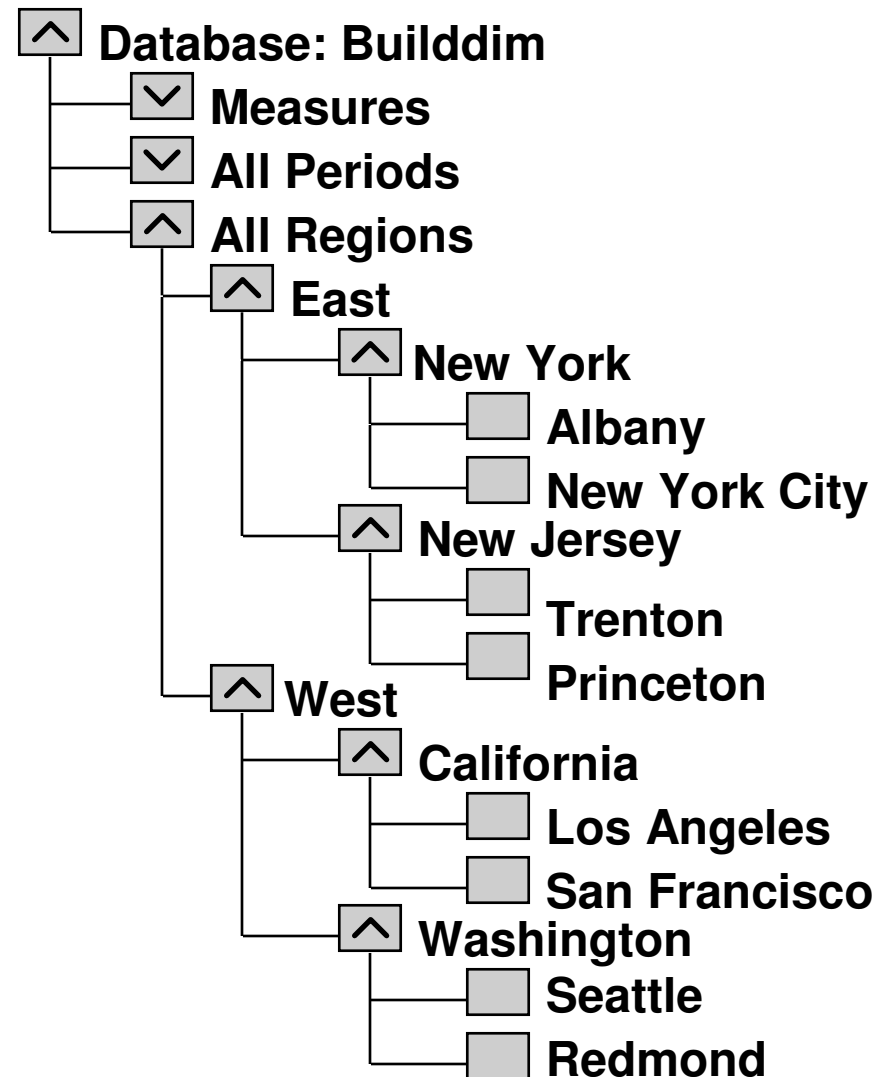
- Automate large builds
- Make model data driven
- Use the Data Prep (rules) Editor
- Generation Build, Level Build, Parent/Child Build
- Build the Rule, then Update Outline
- Source files are not changed

Rules Interpret Metadata Files



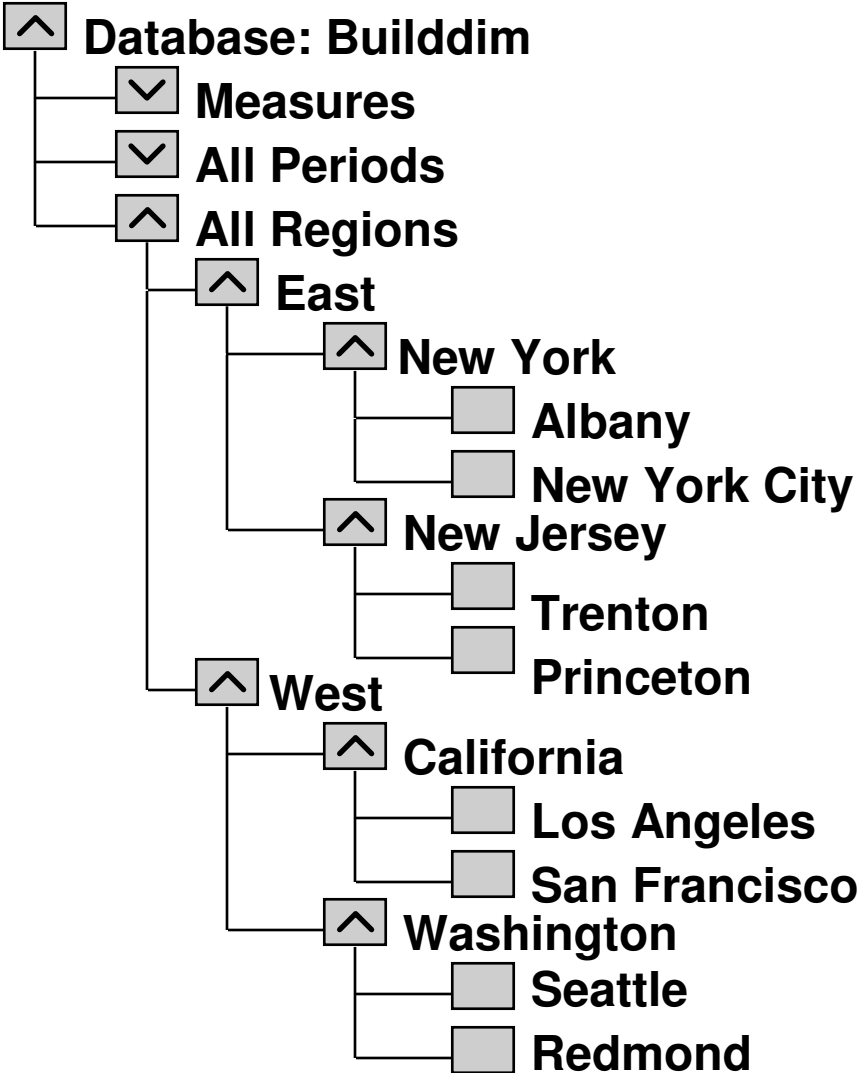
Generation Build

<u>GEN 2</u>	<u>GEN 3</u>	<u>GEN 4</u>
East	New York	Albany
East	New York	New York City
East	New Jersey	Trenton
East	New Jersey	Princeton
West	California	Los Angeles
West	California	San Francisco
West	Washington	Seattle
West	Washington	Redmond



Level Build

<u>LEV 0</u>	<u>LEV 1</u>	<u>LEV 2</u>
Albany New York City	New York New York	East East
Trenton Princeton	New Jersey New Jersey	East East
Los Angeles San Francisco	California California	West West
Seattle Redmond	Washington Washington	West West



Parent/Child Build

PARENT 0

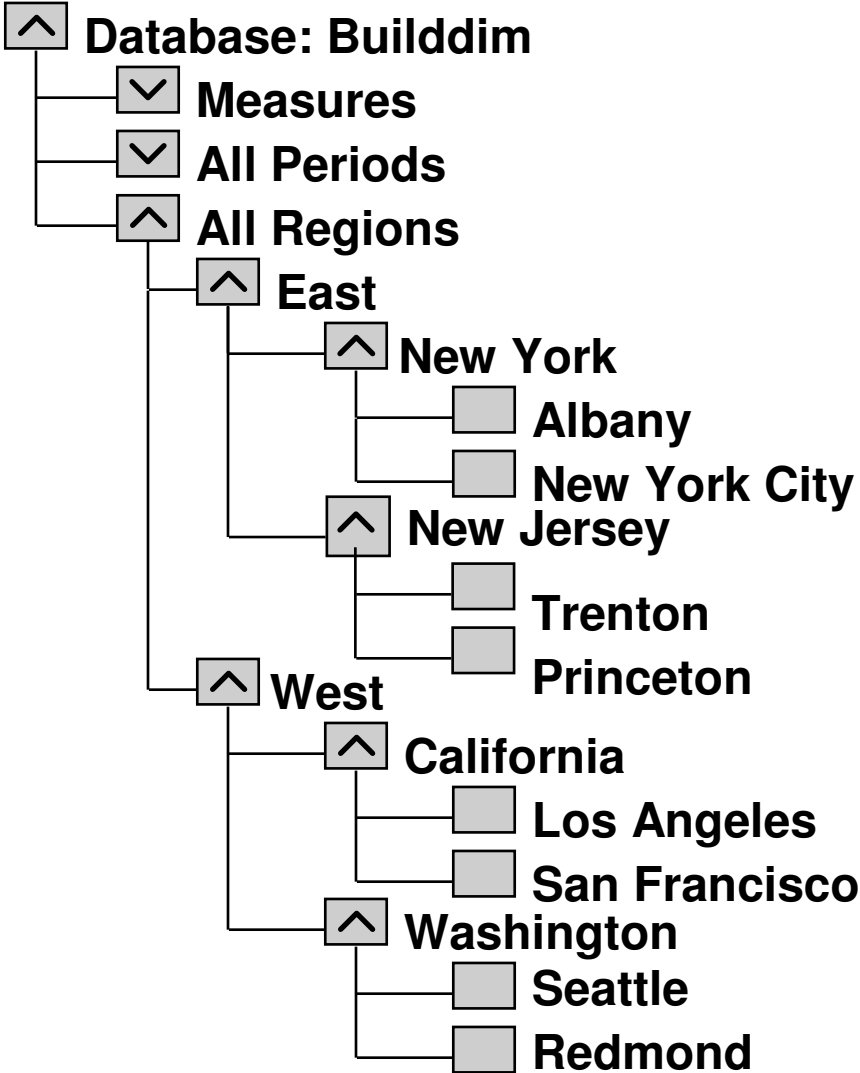
East
New York
New York
East
New Jersey
New Jersev

West
California
California
West
Washington
Washington

CHILD 0

New York
Albany
New York City
New Jersey
Trenton
Princeton

California
Los Angeles
San Francisco
Washington
Seattle
Redmond



Transforming Metadata

- Field properties per column
- Text replacement
- Split/join/move/create columns
- Select/reject rows

Primary and Secondary Fields

- ◆ Primary fields create a new layer
- ◆ Secondary fields apply properties to the preceding primary field
- ◆ Properties may include aliases, consolidation, storage, twopass, formulas, time balance, expense UDA's etc.)

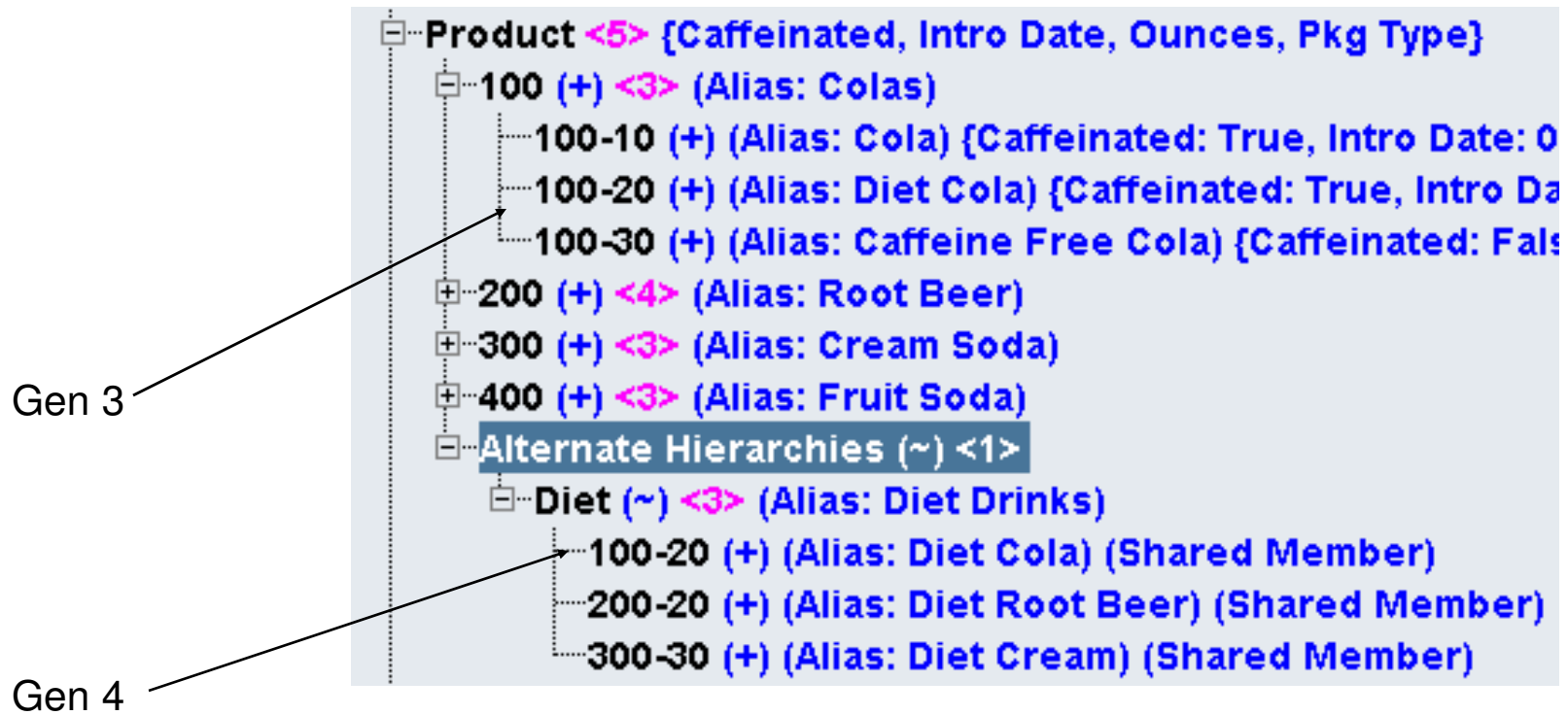
Rules for Field Properties

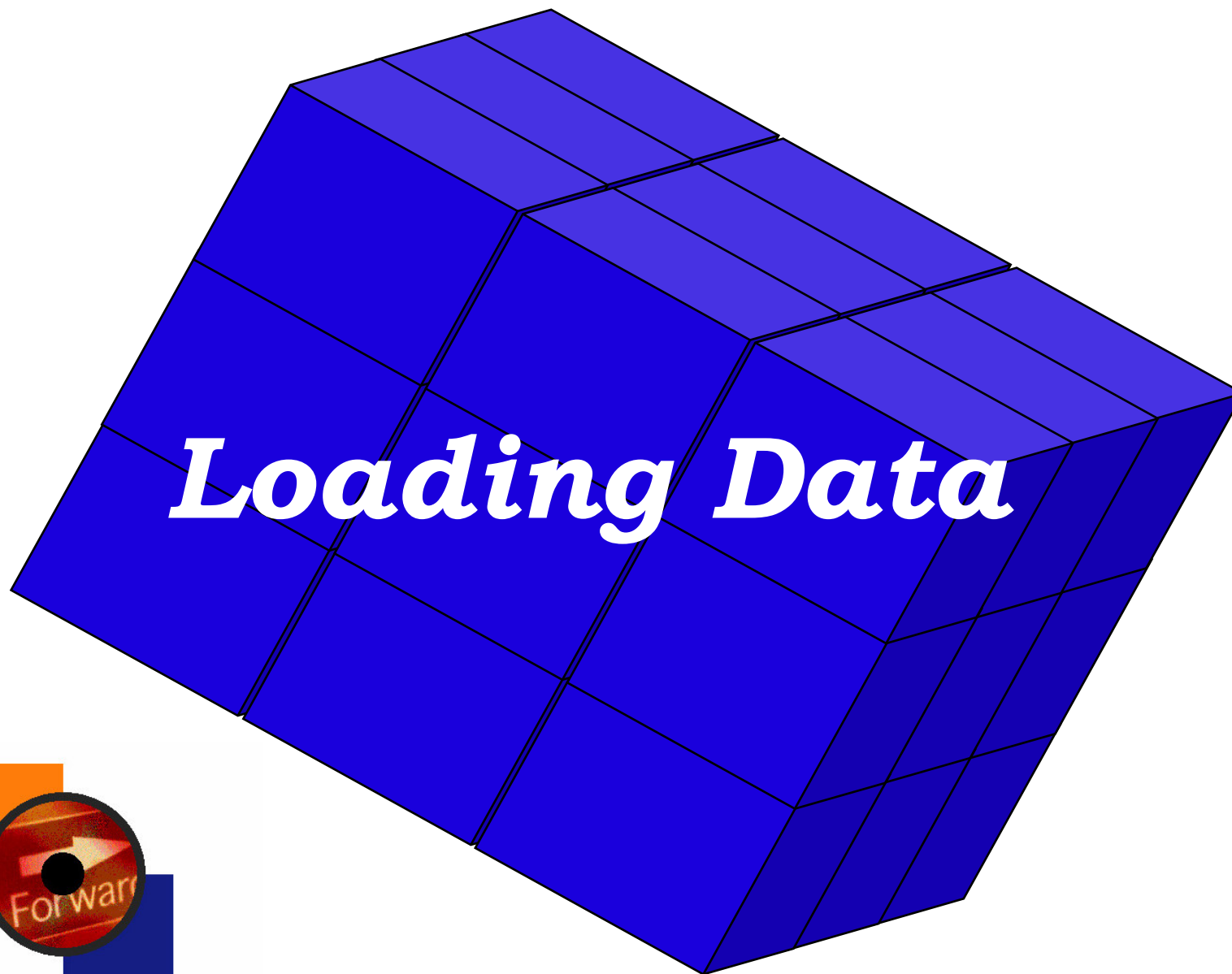
- Gen/Lev fields must be in order
- Gen/Lev numbers must be contiguous
- Secondary fields (member properties) must immediately follow primary field
- Cannot build Generation 1 (dimension name)

Comparing Build Methods

- ◆ To share members generation and level methods require a “dupgen” or “duplevel” column.
- ◆ Parent /Child shares automatically
- ◆ Generation method can't share members at different generations
- ◆ Level method can share members at different generations
- ◆ Parent/Child method is easiest to share and easiest to change granularity

Shares at Different Generations





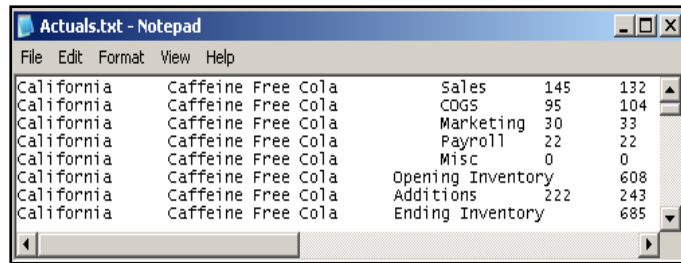
MARKETING TECHNOLOGIES GROUP
New York Information Technology Center:
55 Broad Street, 10th Floor, New York, NY 10004

Marketing Technologies Group

Methods for Loading Data

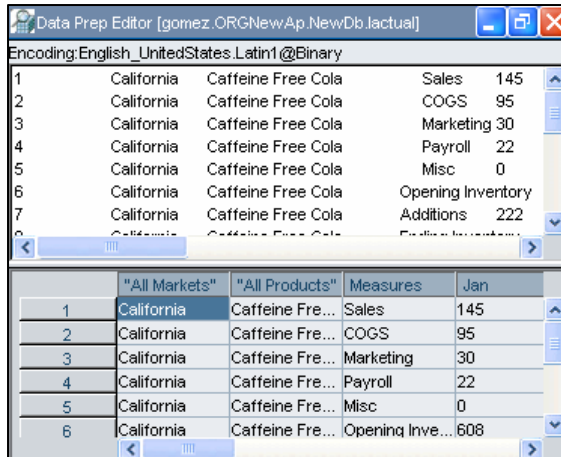
- Data load rule
- Free form
- Spreadsheet lock & send
- Integration Services

Rules Interpret Data Files



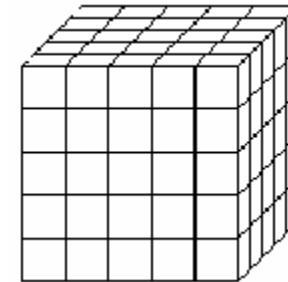
Location	Product	Measure	Value
California	Caffeine Free Cola	Sales	145
California	Caffeine Free Cola	COGS	95
California	Caffeine Free Cola	Marketing	30
California	Caffeine Free Cola	Payroll	22
California	Caffeine Free Cola	Misc	0
California	Caffeine Free Cola	Opening Inventory	608
California	Caffeine Free Cola	Additions	222
California	Caffeine Free Cola	Ending Inventory	685

Data text file (or spreadsheet)



	"All Markets"	"All Products"	Measures	Jan
1	California	Caffeine Fre...	Sales	145
2	California	Caffeine Fre...	COGS	95
3	California	Caffeine Fre...	Marketing	30
4	California	Caffeine Fre...	Payroll	22
5	California	Caffeine Fre...	Misc	0
6	California	Caffeine Fre...	Opening Inve...	608

Data load rule



Essbase database

Data Load Rule Concepts

- ◆ Every cell (intersection) has a name including one member from each stored dimension (coordinates)
- ◆ Loading data require a complete intersection name for each number to be loaded
- ◆ Use “Headers” to supply missing member names
- ◆ You can use aliases
- ◆ You can load data “high” (not best practice)

Field Properties for Data Load

- ◆ If the column contains member names the field property is a dimension name
- ◆ If the column contains numbers the field property is a (level 0) member name

Loading Data Using Free Form

- Only files that Essbase understands “naturally”
- All dimensions identified
- All member co-ordinates before data
- Clear separation of page, row and column fields
- Usually files that Essbase wrote

Optimizing Loading Data

- Sort by sparse dimensions
 - Touch each block only once
- Use “wide” files instead of “long” files (wide on dense dimension!!!)
 - Multiple data columns e.g. 12 months across
 - File is far smaller and requires less interpretation
- Increase index cache
 - More index in RAM requires less disk I/O