



# Model Design I



## Objectives

- Present the DI Model and its structure
- Discuss the Build Process
- Show use of the Visual Builder to create Models
- Discover the fundamentals of the DivePlan relationship to the Model process
- Show use of DiveMaster to create DivePlans





## Scope

- Presented by starting with an example set of input fields
- Finish by discussing DivePlans
- Application to your environment will require you to know your data and what you want to do with it



## Chapter 1: Diver Fundamentals

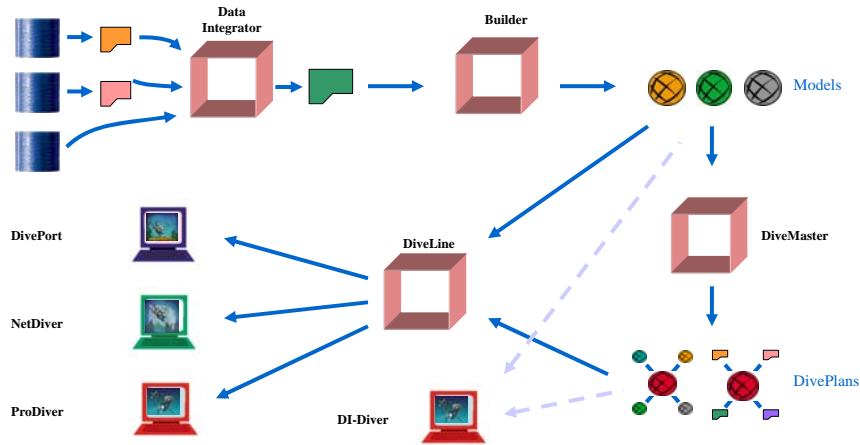
- The Diver Solution™
- Key Terms



# Product Family

## The Diver Solution™

### Data Flow



## The Default Dive Window

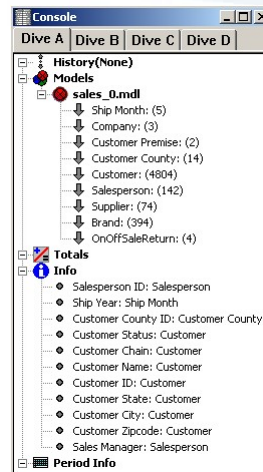
- Tabular
  - Spreadsheet-like for analysis
- Contains:
  - Dimension
    - “Dive-able Field”
  - Summaries
    - Numeric Fields
  - Columns
    - Other Info, Calculations
  - Totals

Payor	Number of Discharges	Total Disch Days	Avg LOS	Total Charges	Total Operating Costs	Total Payments
Totals	29,456	76,891	2.6	78,683,932	27,169,538	27,546,604
BLUE CROS ( 52 )	3,078	8,500	2.8	7,761,138	2,632,703	3,468,794
CAID PEND ( 23 )	58	207	3.6	503,247	188,378	10
CAID/HMO ( 22 )	631	1,145	1.8	1,383,934	489,299	269,474
CARE/HMO ( 15 )	158	790	5.0	813,135	282,321	274,520
CHAMPUS ( 80 )	136	348	2.6	311,101	99,403	108,040
CHARITY ( 70 )	134	519	3.9	872,001	298,761	8,819
COMM/AUTO ( 30 )	903	2,330	2.6	2,540,936	881,274	1,226,402
EMPL/OTH ( 60 )	1,136	2,872	2.5	1,601,316	541,104	64,872
HMO/PPPO ( 50 )	4,189	10,492	2.5	12,058,685	3,973,861	5,960,417
LIABILITY ( 32 )	1	1	1.0	272	138	121
MCAREMORE ( 11 )	11	450	40.9	154,573	58,432	50,719
MEDICAID ( 20 )	1,399	4,615	3.3	4,609,453	1,680,877	1,032,112
MEDICARE ( 10 )	15,345	39,394	2.6	43,964,130	15,359,016	14,556,423



## Console Window

Dive control window which displays the Model's Dimensions and Info Fields.



## DI-Diver as a Developer's Tool

You may wish to install DI-Diver, Dimensional Insight's stand-alone Diver client on the production server to aid in:

- Quick checking of source data
- Exposing Data Integrity problems
- Validating Data
- Developing DivePlans
- Developing Markers and DiveBooks



## Chapter 2: Model Design Basics

### Model Field Classifications

- Dimensions
- Info Fields
- Summaries

### Field Requirements

- Names
  - *29 characters maximum displayed, best to avoid special characters other than space, underscore*
- Length
  - *512 characters maximum displayed*



## Model Field Classifications

- Dimensions
  - *Core - 32 maximum*
  - *Dynamic*
  - *Combine - 200 maximum*
- Info Fields
  - *Relationship to ...*
    - Dimensions
    - Use as Dynamic Dimensions
    - Columns
- Summaries



## The Example Data

Invoice ID	Product Name	Quantity Sold
Invoice Date	Product ID	Unit Price
Invoice Month	Product Line	Unit Cost
Invoice Quarter	Product Class	Quantity in Stock
Invoice Year	Product Size	
Invoice Year-Mo	Salesperson	
Customer Name	Sales Region	
Customer ID	Sales Manager	
Customer Address		
Customer City		
Customer State		
Customer ZIP		



## Dimensions and Info Fields

### Model

#### Dimensions

Invoice ID  
Invoice Date  
Invoice Month  
Invoice Quarter  
Invoice Year  
Invoice Year-Mo

Customer Name  
Customer ID  
Customer City  
Customer State  
Customer ZIP

Product Name  
Product ID  
Product Line  
Product Class  
Product Size

Salesperson  
Sales Region  
Sales Manager

#### Info Fields

Customer Address

#### Summaries

Quantity Sold  
Unit Price  
Unit Cost  
Quantity in Stock



## Dimensions and Info Fields

### Model

#### Dimensions and Info Fields

Invoice Year-Mo  
Invoice Quarter  
Invoice Year

Invoice Date  
Invoice Month

#### **Customer**

Customer Name  
Customer ID  
Customer Address  
Customer City  
Customer State  
Customer ZIP

Invoice ID

#### **Product**

Product Name  
Product ID  
Product Line  
Product Class  
Product Size

Salesperson  
Sales Region  
Sales Manager

#### Summaries

Quantity Sold  
Unit Price  
Unit Cost  
Quantity in Stock



## Summary Fields: "Numbers"

#### Dimensions and Info Fields

Invoice Year-Mo  
Invoice Quarter  
Invoice Year

Invoice Date  
Invoice Month

#### Customer

Customer Name  
Customer ID  
Customer Address  
Customer City  
Customer State  
Customer ZIP

Invoice ID

#### Product

Product Name  
Product ID  
Product Line  
Product Class  
Product Size

Salesperson  
Sales Region  
Sales Manager

#### Summaries

**Quantity Sold**  
**Unit Price**  
**Unit Cost**  
**Quantity in Stock**



## Summary Fields

Summary fields exist at the record level and are summed over all rows of data.

Customer Name	Product Name	Quantity Sold	Unit Price
Dan	Coca-Cola	12	0.75
Dan	Diet Coke	6	1.00
Fred	Dr Pepper	12	0.75
Fred	Coca-Cola	24	0.60
		54	3.10



## Summary Fields

Since Summary fields are summed over all rows of data, this must make sense!

Customer Name	Product Name	Quantity Sold	Unit Price	Quantity * Unit Price
Dan	Coca-Cola	12	0.75	Extended Price 9.00
Dan	Diet Coke	6	1.00	6.00
Fred	Dr Pepper	12	0.75	9.00
Fred	Coca-Cola	24	0.60	14.40
		54	XXXXXX	38.40



## Summary Fields: "Levels"

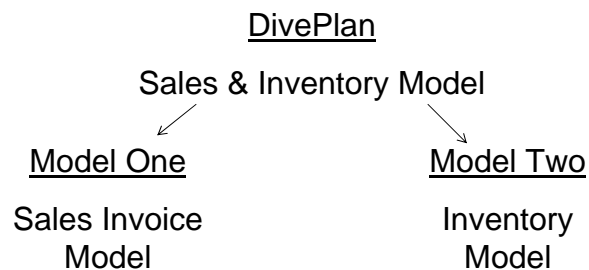
In addition to creating numbers that can be summed, we have to make sure we know the "level" at which they can be summed.

Not every number can be summed at the transaction level!



## Summary Fields: "Levels"

One solution: Build two Models separately and join by a DivePlan as a multi-level Model





## Distribution Data: A Sample Model Design

There is more than one solution. The correct solution is determined by understanding the use of the final Model.

Date of Sale	02/16/2010	Dimension
Account Name	Bill's Place	Part of a new Dimension concatenation with Account
Account	123457	Part of a new Dimension concatenation with Account Name
Address 1	23 Independence Ave	Part of a new Info Field, Address concat with Address 2
Address 2	Suite 200	Part of a new Info Field, Address concat with Address 1
City	Worcester	Part of a new Info Field concat with State
State	MA	Part of a new Info Field concat with City
Zip + 4	01750-2345	Info Field
Telephone	5087621492	Info Field, add format
Total Dollars	345.20	Summary
YTD Dollars	545.80	Not needed – Builder will sum
Product	FG2345A	Dimension



© 2010. Dimensional Insight, Inc



## Health Care Data: A Sample Model Design

Discharge Date	02/16/2010	Dimension, type as a date
Patient Name	Kirkland, R	Part of a new Dimension concatenation with Patient ID
Patient ID	123457	Part of a new Dimension concatenation with Patient Name
Patient ZIP Code	01750	Info Field, type as a string
Hospital	Powers Memorial	Dimension
Patient Type	Inpatient	Info Field
Payor	MEDICARE	Dimension
Total Payments	1345.20	Summary
Days	5	Summary
ALOS	5.0	Diver Calculated field after Build



© 2010. Dimensional Insight, Inc



## Chapter 3: The Builder

- **Builder**
- **Builder Components**
- **Creating the Description File**
- **Building the Model**



## Builder in The Diver Solution™

- **Visual Builder**
  - Windows OS only
- **Production Builder**
  - Command Line
  - Platform-independent Description scripts
  - Special Limited Versions Available
- **Memory Builder**
  - DI-Diver
- **Tunnel**
  - DiveLine client Memory Model
- **Building a detail Model**
  - Command Line



## Builder

Components of Builder

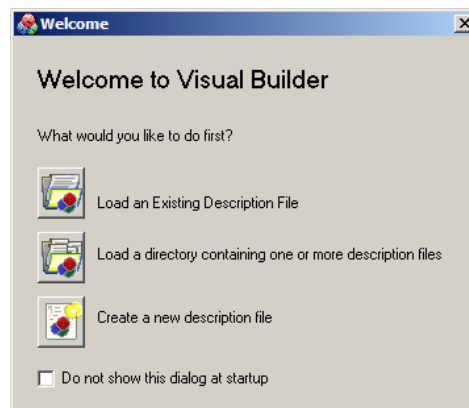
### Description File:

- **Input**
  - File
  - Dictionary File
  - (if no Column Headers or not ODBC)
- **Model Structure**
  - Dimension, Info Fields, and Summaries
- **Output**
  - Build Journal File
  - Model
- **Options**
  - Settings



## Visual Builder

What would you like to do first?







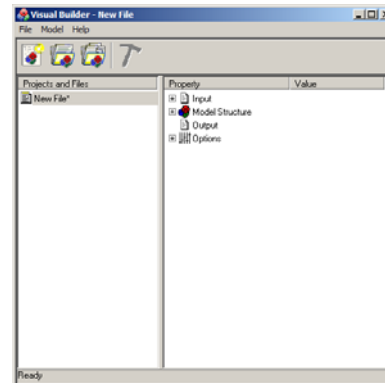


## Create a New Description File

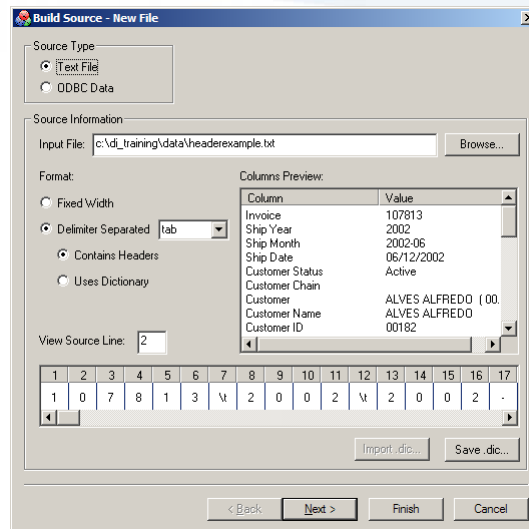


### • Four Step Process

-  *Set the Input File*
-  *Define the Model Structure*
-  *Set the Model Output*
-  *Define Options*



## Set the Input File - Text File





## Set the Input File - ODBC Data



Build Source - New File

Source Type

Text File

ODBC Data

Source Information

Data Sources:

- dBASE Files
- Excel Files
- Invoice Data Warehouse
- MS Access Database

Tables:

- Customer
- Sales\_90day

Columns Preview:

- Column
- ID
- Customer ID
- Customer Name
- City
- State
- Zipcode
- Premise
- ChainID
- Customer Status
- CustNew

Database: C:\DL\_TRAINING\ID

Owner:

Use Custom SQL

select \* from 'C:\DL\_TRAINING\DATA\Sales'.Customer where state = 'NY'

Keep Trailing Space in SQL Output

Import SQL... Save SQL...

< Back Next > Finish Cancel



## Saving the Description File



- The Visual Builder is used to create a Description file (.dsc) that can be used by Builder to create a Model.
- Use File > Save File As... to create and save the Description file.
- You can save the Description file during any stage of the Visual Build Process.



# Model Specification - Dimensions and Summaries



Define Model Structure - New File

Sort Data and Computed Columns into Dimension, Summary, and Info categories. Drag and Drop items from the list to the Tree or use the "D", "S", or "I" keys to set the category type of each item in the lists.

Available Columns (32)

From Source:

- Invoice
- Ship Year
- Ship Month
- Ship Date
- Customer Status
- Customer Chain
- Customer
- Customer Name
- Customer ID
- Customer State

Calculated Fields:

Model Specification

- Dimensions (2), Infos (0)
  - Customer
    - Ship Date
- Summaries (3)
  - Cases
  - Revenue
  - Cost
- Details

New Edit Delete

Properties

< Back Next > Finish Cancel



DATA - INFORMATION - ACTION

© 2010. Dimensional Insight, Inc.



# Model Specification - Info Fields



Define Model Structure - New File

Sort Data and Computed Columns into Dimension, Summary, and Info categories. Drag and Drop items from the list to the Tree or use the "D", "S", or "I" keys to set the category type of each item in the lists.

Available Columns (32)

From Source:

- Customer Chain
- Customer
- Customer Name
- Customer ID
- Customer State
- Customer County ID
- Customer County
- Customer City
- Customer Zipcode
- Customer Premise

Calculated Fields:

Model Specification

- Dimensions (2), Infos (5)
  - Customer
    - Customer Name
    - Customer ID
    - Customer State
    - Customer Zipcode
    - Customer City
  - Ship Date
- Summaries (3)
  - Cases
  - Revenue
  - Cost
- Details

New Edit Delete

Properties

< Back Next > Finish Cancel



DATA - INFORMATION - ACTION

© 2010. Dimensional Insight, Inc.



## Model Specification - Column Properties



**Column Properties**

Name:

Alias:

Sort by:

Column is Always a String

Column Precalculates Dimcounts

Column Contains Dates

Date Formatting

Accept 2-digit years

First Year in Range:



## Model Specifications - Calculated Fields



**Define Calculation**

Calculation Name:

Definition:

```
concat(Customer Name, " ", Customer ID, " ")
```

Columns:

- Invoice
- Customer Status
- Customer Chain
- Customer Name
- Customer ID
- Customer State
- Customer County ID
- Customer County
- Customer City
- Customer Zipcode

Functions:

- abs
- add
- age
- and
- arccos
- arcsin
- arctan
- ascii
- capitalize
- ceil



## Create a New Description File



Build Output - c:\di\_solution\executables\test.dsc

Output Model:  
c:\di\_training\_hc\models\test.mdl Browse...

Build Journal:  
c:\di\_training\_hc\models\test.jou Browse...

Append Build Info to Journal

< Back Next > Finish Cancel



## Building the Model



- Save the Description File
- Start the Model Build process

Building...

Thu Jan 03 10:49:18 2008 Starting phase 4...

Detail information:  
Dtl row size = 36; bucket size = 240871

Thu Jan 03 10:49:18 2008 Starting phase 5...

Thu Jan 03 10:49:19 2008 Starting phase 6... (elapsed time 0:00:01)

Cancel



## Chapter 4: Data Discovery

- Gather Information
- Quickly Assess Data Attributes
- Handle Data Quality & Missing Values
- Identify Dimensions
- Enable Time Series
- Promote Info Fields
- Combine Dimensions
- Focus on Analysis



## Data Discovery Information Gathering

### What will be the Purpose of the Model?

- Analysis or reporting?
- Questions to be answered
- Reports to be replaced
- Key users
- How data will be validated
- Expectations to be met
  - Response Time
  - Ease of Use
  - Ease of Maintenance



## Data Discovery

Information Gathering

- Sources & Data Volumes
- Production Window / Timing of Input
- Access Methods
- Security
- Handling Change (Reclassify)
- DimCounts
- Types of Calculations
- Business Rules and Data Definitions
- Detail vs. Summary Data



## Data Discovery

Quickly Assess Data Attributes

- Where's the data, and what do we have?
- Might Use DI-Diver or Builder
  - Limit rows (-first 5000)
- Validate Dictionary or Column Headers
- Determine Strings vs. Numerics

Customer ID  
Customer ZIP  
Product ID



## Data Discovery

Handling Data Quality & Missing Values

### Options:

- Return to the Source & Correct
- Identify Calcs & Determine Calc Locations
  - source (e.g. SQL)
  - Data Integrator
  - Builder
  - Diver

### Result:

- “Good Clean” Data



## Data Discovery

Handling Date Values

- Generally used for more detailed data
- “Date” is stored in the Model as an integer:
  - 01/01/2010 = 40179      07/15/2004 = 38183
- Builder / Diver handle proper sorting
- Integrator / Builder functions
  - date\_value() - converts “DI date” to integer
  - format\_date() - specify date format & convert to “DI date” (yyyy/mm/dd)
- Diver
  - open a Dive Window with the Date Dimension
  - double-click on its heading to edit the Date display Format
  - save in DivePlan



## Data Discovery

Enabling Time Series - Period Value

- Typical way to define time periods
  - Some form of date roll-up :
    - *Week Month Quarter Half Year*
  - String values in Model : (e.g. 2010-01)
- Integrator / Builder functions
  - substr() - segments a date field
  - concat() - re-assembles date field segments
- Diver
  - open a Dive Window with the Period Dimension
  - double-click on column heading to classify as a Period type
  - specify "type" and "format" of the "period" value
  - save in DivePlan



## Data Discovery

Determine Summary Types Required

- Count
  - Always created
    - *includes Count, % Count*
- Misc. Options: Sum Types
  - Sum (default)
    - *includes Total, Average, % Total*
  - Min
  - Max
  - Std Dev



## Data Discovery

Identify Dimensions

- Focus on Non-Summary Fields
- Base on Data Relationships and Importance
- Strive to Minimize # of Core Dimensions

Customer	Product
Customer Name	Product Name
Customer ID	Product ID
Customer Address	Product Line
Customer City	Product Class
Customer State	Product Size
Customer ZIP	



## Data Discovery

Candidates for Core Dimensions

- Included in first 3-4 dives
  - Year-Month
  - Company
- Value at transaction time is important
  - Inventory
  - Current Salesperson
- Values can not be re-classified
- To associate a Lookup
- Security – based
  - Sales Manager
  - Salesperson
- Dim:Info ratio exceeds 20:1
  - Invoice Date
  - Invoice Month
  - Invoice Year



## Data Discovery

Reduce the Number of Core Dimensions

- **Combine Dimensions**
  - *Concatenate two or more fields*
- **Classify the field as an Info Field**
  - *Identify a 1:1 Relationship with another Dimension field*



## Data Discovery

Considering Performance

- **Summaries**
  - Count: Always there
  - Sum Types: These are not free!
  - DimCounts in Model Build: Fast, Limited
- **Dimensions**
  - Fewer is better
  - Flatter is better
  - Values: Not too few, not too many
- **Info Fields**
  - One to one relationship to Core Dimension
  - Stored in internal table (or *external* lookup)
  - Practically free!



## Chapter 5: DI Model Structures

- The Model Structure
- The Time Dimension Across Models



## DI Model Structures

- **General analysis (“Ad Hoc”)**
  - Most flexibility
  - Eases data validation
  - Fewer Models: long and thin
- **Specialized reporting and analysis**
  - Many business reports
  - More Models: short and fat
- **Report Palette / DivePort**
  - Speed is the driving factor !!!
  - Many Models: short & fat; long & thin



## The Time Dimension Across Models

- Only the current data which is changeable needs to be refreshed in your Models.
- History can be combined with current data through the use of a Multi-Model DivePlan containing Consolidated Models.
- The History Models are only rebuilt if necessary.



## Options for Large Data Sets

- Core Model with a minimum of Dimensions
- Detail Model
- Console Jump
- Tunnel



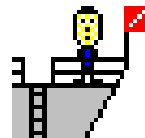
## Model Building Best Practices

- Keep Core Dimensions to a minimum  
—*Rule of thumb: 12 to 18*
- Concatenate Unique ID's and Descriptions for Core Dimensions
- Include Liberal Use of Info Fields
- Set Date Properties in the Description File
- Plan to create a DivePlan for Your Models (covered next)



## Chapter 6: DiveMaster

- About DiveMaster
- Multi-Modeling
- Using DiveMaster
  - Set Model
  - Set Lookups
  - Set Dynamic Dimensions
  - Set Categories





## About DiveMaster

A User Interface to create a standard “business view” of the data in a DI-Model, organized into a DivePlan for use by Diver

- Combining Physical Models
- Aliasing
- Lookups
- Dynamic Dimensions
- Categories



## DiveMaster Multi-Modeling





## Consolidated DivePlans

- Most common type, used when Model structures and transaction level are similar
- Creates a new Dimension, “Model”
- Sums all Summaries
- Consolidates Dimension values from all Models where the Dimension Name is the same



## Multi-Level DivePlans

- Can be used when the transaction level is different
- There are shared Dimensions, but no shared Summaries
- “Model” Dimension is not available



## Multi-Level Diveplans

### Model One

#### Dimensions and Infos

Invoice Date  
Invoice Month  
Invoice Quarter  
Invoice Year  
Invoice Year-Mo

Invoice ID

Customer  
Customer Name  
Customer ID  
Customer Address  
Customer City  
Customer State  
Customer ZIP

Product  
Product Name  
Product ID  
Product Line  
Product Class  
Product Size

Salesperson  
Sales Region  
Sales Manager

#### Summaries

Quantity Sold  
Extended Price  
Extended Cost

### Model Two

#### Dimensions and Infos

Product  
Product Name  
Product ID  
Product Line  
Product Class  
Product Size

#### Summaries

Quantity in Stock



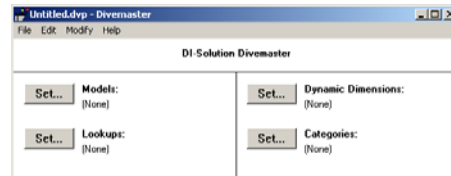
## Complex Multi-Model DivePlans

- A Complex DivePlan can be created that is composed of other DivePlans
- Care must be taken as a Complex DivePlan may not contain the path to the member DivePlans
- Mixed Consolidated and Multi-Level DivePlans can be created



## Steps to creating a DivePlan in DiveMaster

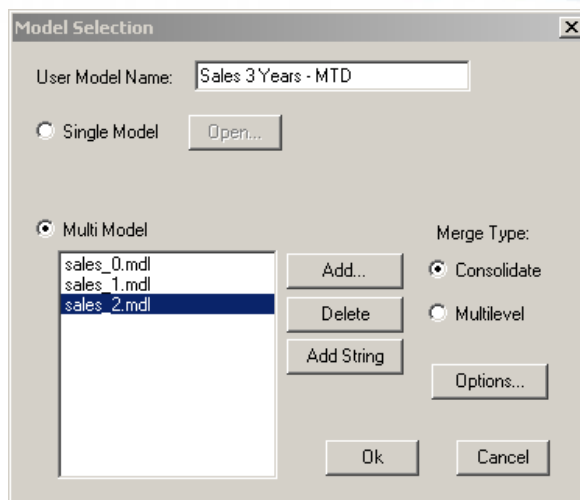
- Set Models
- Set Lookups
- Set Dynamic Dimensions
- Set Categories



Use Diver for additional customization



## Steps to a DivePlan - Set Models





## Steps to a DivePlan - Set Lookups

- “External” Info Fields
- Associated with a Core Dimension
- Loaded into memory when DivePlan opened
- Multiple associations per Dimension
- Multiple Info Fields per Lookup
- Tab-delimited w/ column headers
- ASCII file w/ appropriate Dictionary



## Steps to a DivePlan - Set Dynamic Dimensions

### Dynamic Dimensions

- Info Fields made available as Dimensions
- Over 32 Core Dimension Limit
- Total Mix: 200 dive-able Dimensions

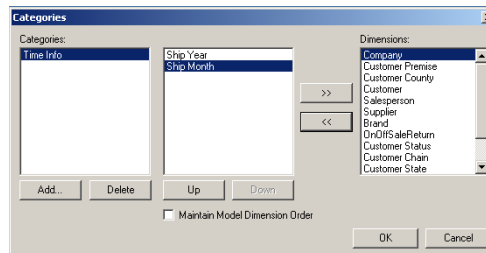
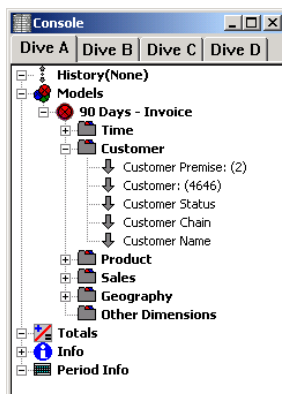


## Steps to a DivePlan - Set Categories

- Set Models
- Set Lookups
- Set Dynamic Dimensions
- **Set Categories**



## Steps to a DivePlan - Set Categories





## DivePlan Best Practices

### Two types:

#### Ad Hoc

- Select Columns: 6 or less Summaries
- Available columns for common Time Periods
- Create multiple DivePlans for logical Model groupings
  - *Multiple years and current period*
  - *Consolidated divisions and individual divisions*

#### Report-Specific

- Restrict data to that needed in Report
  - *Selected Columns*
  - *Models to be included*



## Chapter 7: Production Considerations

- DI Standard Directory Structure
- Building the Model on a Server
- Moving Files (Models, Dictionaries, Builder Descriptions, DivePlans)
- Using Parameters
- Security Considerations
- Setting the Cache File
- Complex Hierarchical DivePlans



## Project Structure Sample

```
\DI_< project >
  \Data                production data downloads
    \subject 1         history data subject 1
    \subject 2         history data subject 2
  \Programs
    \subject 1         program subject 1
    \subject 2         program subject 1
  \Temp                Integrator output
  \Models              Models, DivePlans
  \Logs                error logging
```



## DI Standard Project Directory

Drive:

```
\DI_<Project name>
  \data                Files brought to the Diver
                       server for processing
  \logs                Redirect all logs here
  \models              Includes everything user sees
  \programs            Integrator and Builder Scripts
  \temp                Store Integrator output
```



## Developing Effective Implementations

### Effective implementation is a top-down methodology

- Discuss expected deliverable
  - Project areas covered
  - Creation of DivePort storyboard
  - Creation of phased plan
  - Future goals and roll-out
- Perform needs assessment
  - Number and type of users
  - Performance requirements
  - Validation and verification plan
  - Goals for information delivery
- Identify data sources
  - Integrate disparate systems
  - Build optimized data Models



## Project Development

A “project” is ...

- A set of Models and DivePlans
- Includes an Ad Hoc DivePlan



## Project Challenges

- Staying within scope
  - *Proof of Concept vs. Pilot*
- When is an iterative, evolutionary process “Complete”?
- Data validation as a spectator sport. You built it, why isn't the data right? Discovering the business rules and getting the right resources is not easy.
- Having too much data, or “data never dies”



## Information Gathering

### What's the data?

- Questions to be answered
- Reports to be:
  - Required
  - Desired
  - Replaced
- Key users
- Expectations to be met
  - Response Time
  - Ease of Use
  - Ease of Maintenance



## Information Gathering

### Where's the Data?

- Data Volumes
- Production Window / Timing of Input
- Access Methods
- Security
- Handling Change
- Types of Calculations



## Production Considerations

- Building the Model on a Server
- Moving Files (Models, Dictionaries, Builder Descriptions, DivePlans)
- Using Parameters
- Security Considerations
- Cache
- Complex Hierarchical DivePlans



## Production Considerations

- Building the Model on a Server
- Moving Files (Models, Dictionaries, Builder Descriptions, DivePlans)
- **Using Parameters**
- Security Considerations
- Cache
- Complex Hierarchical DivePlans



## Production Considerations

- Building the Model on a Server
- Moving Files (Models, Dictionaries, Builder Descriptions, DivePlans)
- **Security Considerations**
- **Cache**
- Complex Hierarchical DivePlans



## Production Considerations

- Building the Model on a Server
- Moving Files (Models, Dictionaries, Builder Descriptions, DivePlans)
- Security Considerations
- Cache
- **Complex Hierarchical DivePlans**