ASUG SAP BusinessObjects
USER CONFERENCE

September 10-13, 2012
Orlando, Florida

Business Intelligence
for a PASSIONATE
COMMUNITY

Session 0611 - Preparing for Life on Planet UNX

Alan Mayer – Solid Ground Technologies

Learning Points

- Learn the pros and cons of the new semantic layer
- Discover the steps required for conversion
- Understand the limitations of this latest version

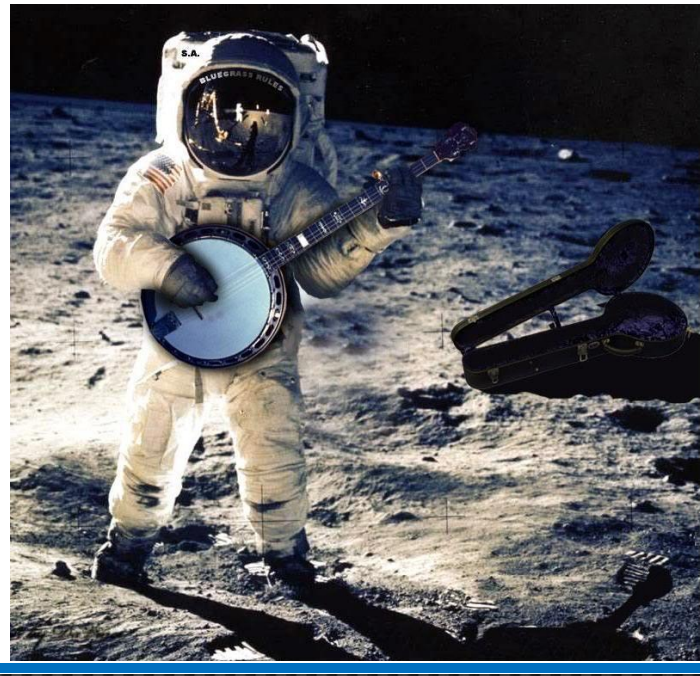
Agenda

- **Introduction**
- Architecture
- Migration
- Features
- Publishing
- Other Considerations
- Wrapping Up

Introduction

Setting Expectations …

- This session will not show you how to build a universe from scratch
  - Could do that in another presentation
  - Not a planet-building discussion
- We will show you how to live on Planet .UNX
  - How to shift from your existing .UNV world
  - What to expect
  - Real life experience spices up the conversation



Agenda

- Introduction
- **Architecture**
- Migration
- Features
- Publishing
- Other Considerations
- Wrapping Up

Gazing at the Planets

Planet **UNV**

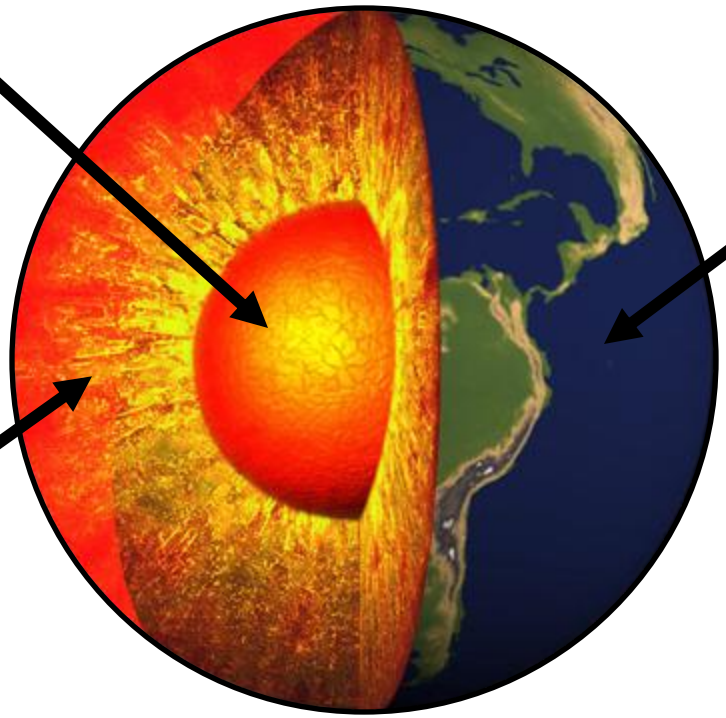
Planet **UNX**

Planet .UNV

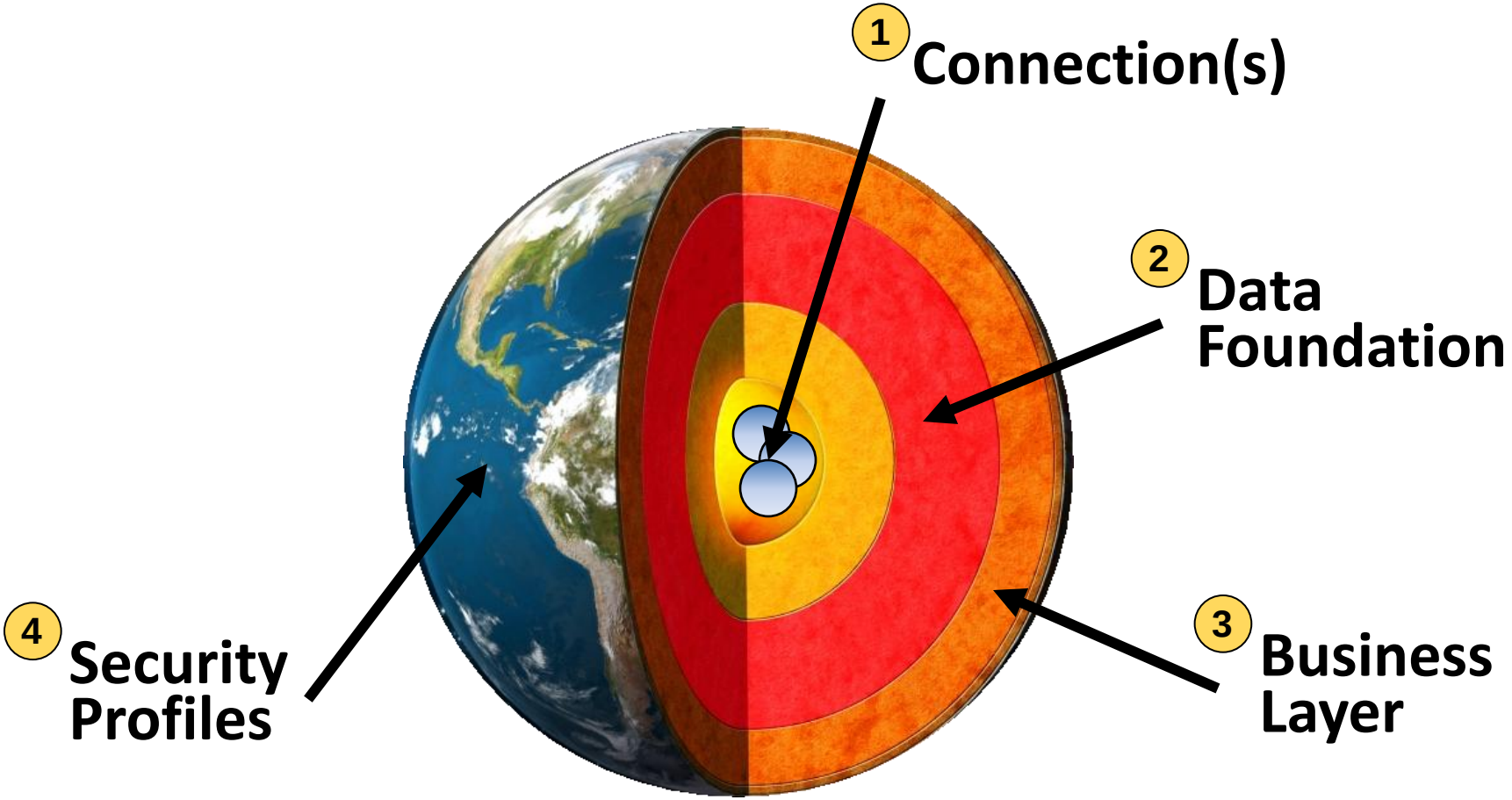
1 Connection

2 Structure and Objects

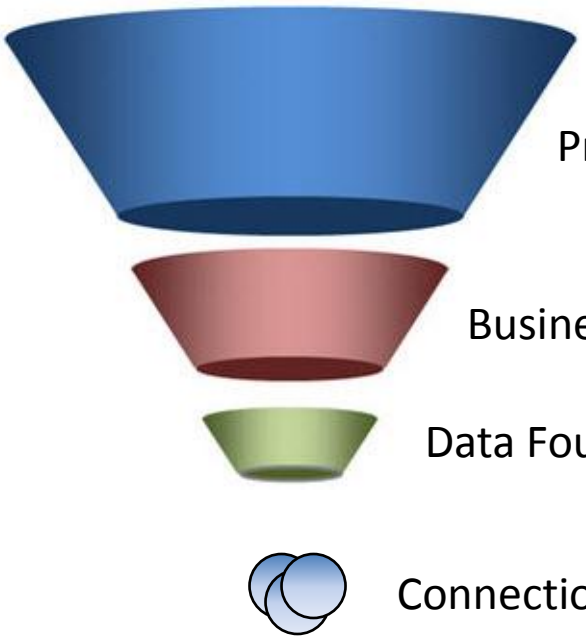
3 Restrictions



Planet .UNX


1
Connection(s)
2
Data
Foundation
3
Business
Layer
4
Security
Profiles

Another Look at Planet .UNX



Many views can be based on a single Business Layer

.blx

Business Layer built from one foundation

.dfx

Data Foundation constructed from one or more connections

.cnx, .cns

Connections created from many data sources

Project

Do the Planetary Math

.cnx or .cns

+ .dfx

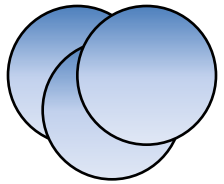
+ .blx

= .unx

Question:
Do layers come with a cost?

Answer:
No.
The .unx file representing the final universe is compiled from all previous layers

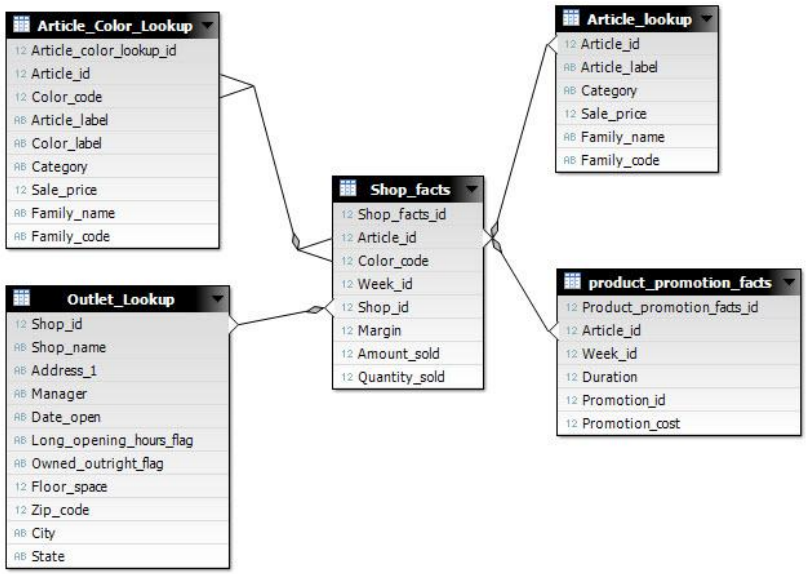
Connections



- Universe connections play the same role as in XI 3.1
- More data sources available since 4.0 was released
  - SAP ERP (Feature Pack 3)
  - Essbase OLAP (Feature Pack 3)
  - CSV files (Feature Pack 3)
  - Stored Procedures (Feature Pack 3)
- New Features
  - Connection folders
  - May not seem like much, until you manage hundreds of connections

Data Foundation

- Similar to the Structure Panel in XI 3.1
  - Tables, derived tables, views, joins
- Many new features have been added
  - The most exciting will be covered later in this presentation
- Why separate data structure from objects?
  - Separation = Greater control, more flexibility
  - Remember connection separation in XIR2?
  - Many universe developers could benefit from the same common data layer



Business Layer

- Similar to the Classes and Objects Panel in XI 3.1
- Enhanced to support OLAP-based data sources
  - Analytical dimensions
  - Hierarchies (**not** v3.1 hierarchies)
  - Named sets
  - Calculated members

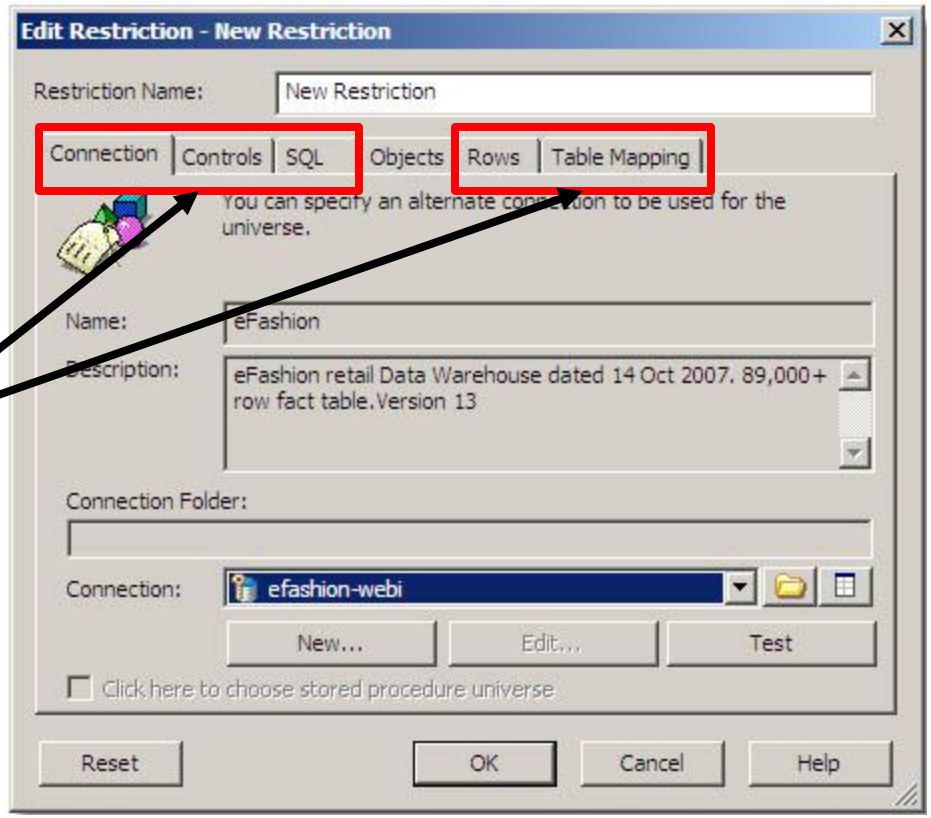
Profiles

- XI v3.1 Universe Restrictions have been split into **PROFILES**

Data Security Profiles control the Data Foundation layer

- Replacement connections
- Query timeouts and rows
- Query options
- WHERE clause restrictions
- Replacement tables / views

v3.1 Universe Restriction Window



Profiles, cont'd

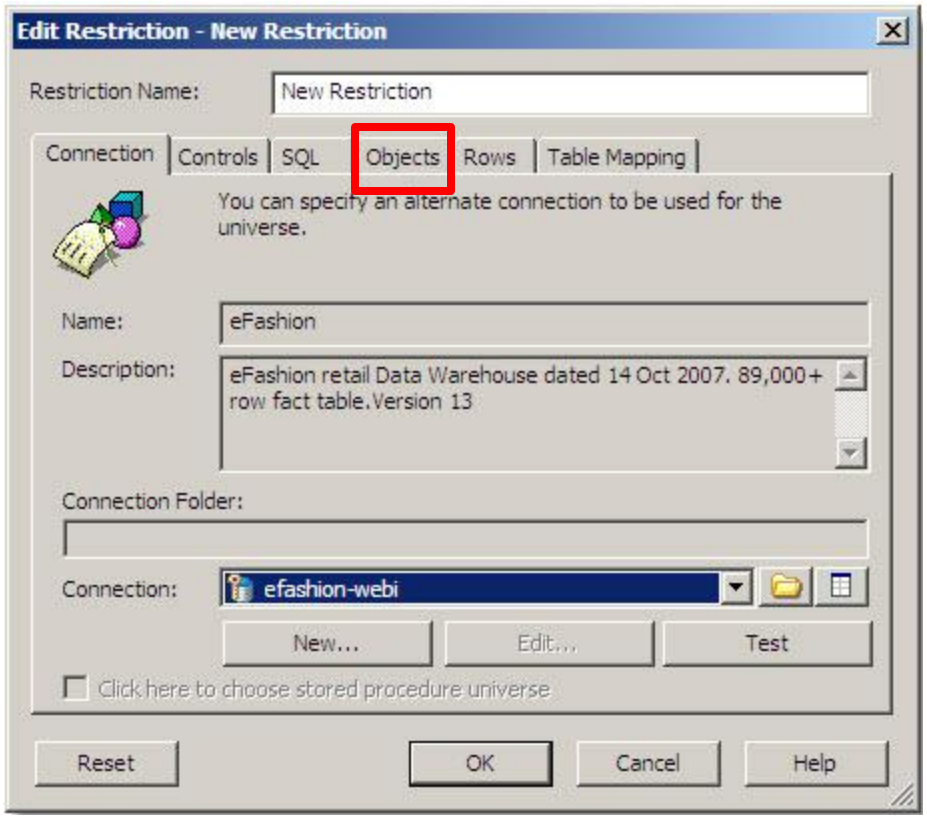
- Business Layer Profile controls which objects are hidden

Former Hidden Objects pains are resolved!

Users with hidden objects can now refresh queries

Controlled by AUTO_UPDATE_QUERY parameter

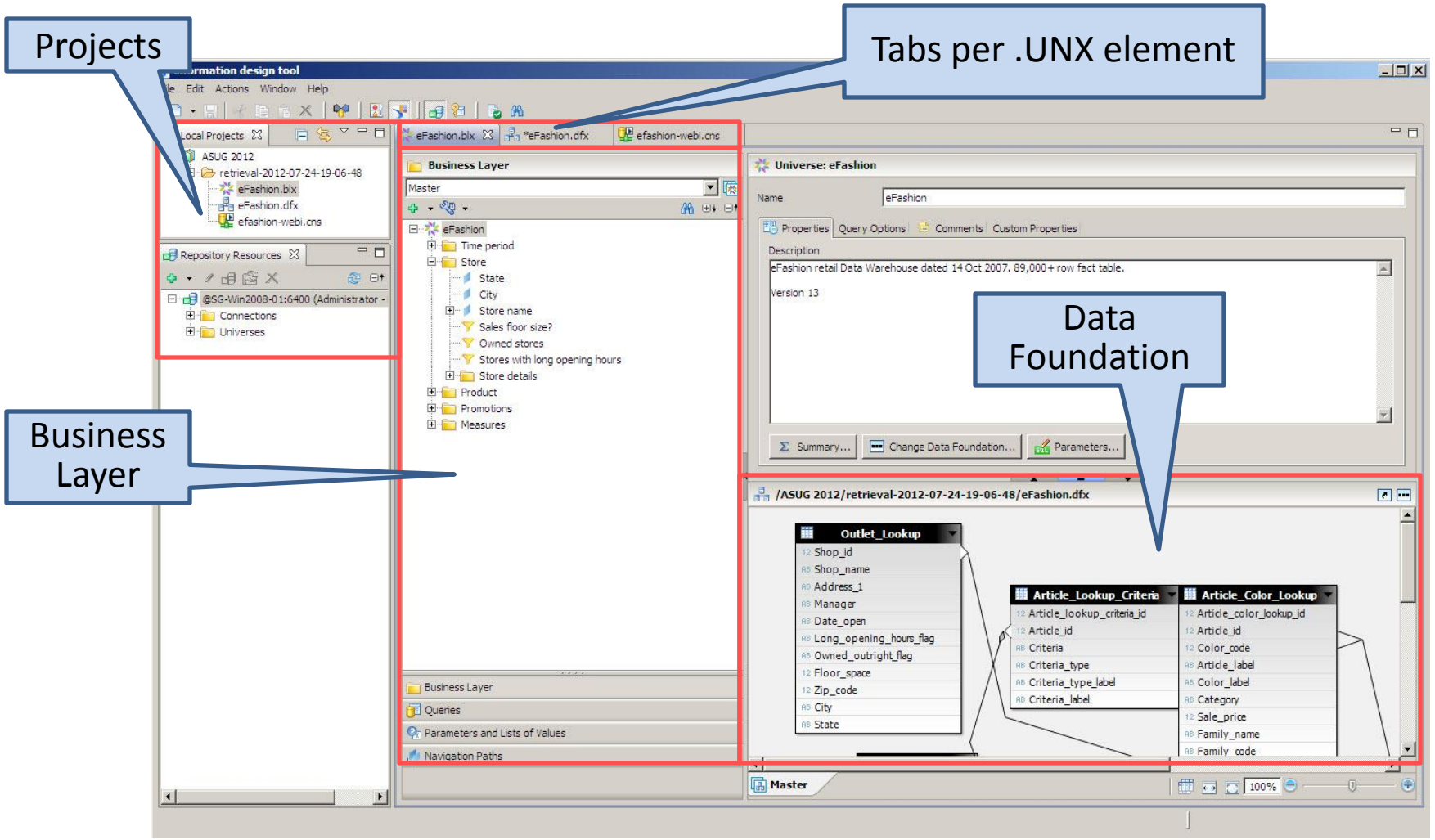
v3.1 Universe Restriction Window



Name Changes

- Changes made since XI v3.1
  - Classes → Folders
  - Detail objects → Attributes
  - Hierarchies → Navgation paths
  - Prompts → Parameters
  - Self joins → Column filters
  - Universe Designer → Universe Design Tool (.unv)
- Why changes now?
  - To be even more difficult to learn (new tool, new terms …)
  - To better align with existing SAP definitions and tools
    - Hierarchies mean something totally different in SAP BW
  - Some names came from lesser known products
    - Business view terminology, maybe?

Information Design Tool (IDT)



Security Changes (CMC)

Needed to create profiles

Used for federated (multi-connection) universes

Sharing / synching layers after publishing project

Applications

Specific Rights for Information Design Tool	Implicit Value	
Administer security profiles	Not Specified	
Compute statistics	Not Specified	
Create, modify, or delete connections	Not Specified	
Publish universes	Not Specified	
Retrieve universes	Not Specified	
Save for all users	Not Specified	
Share projects	Not Specified	

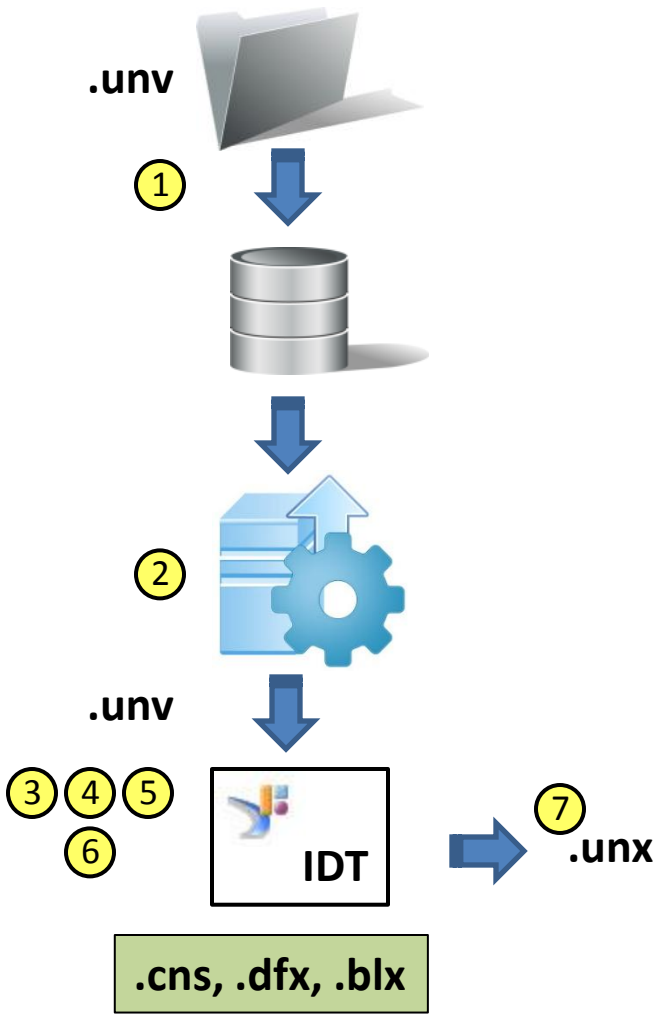
Agenda

- Introduction
- Architecture
- **Migration**
- Features
- Publishing
- Other Considerations
- Wrapping Up

Converting .UNV to .UNX

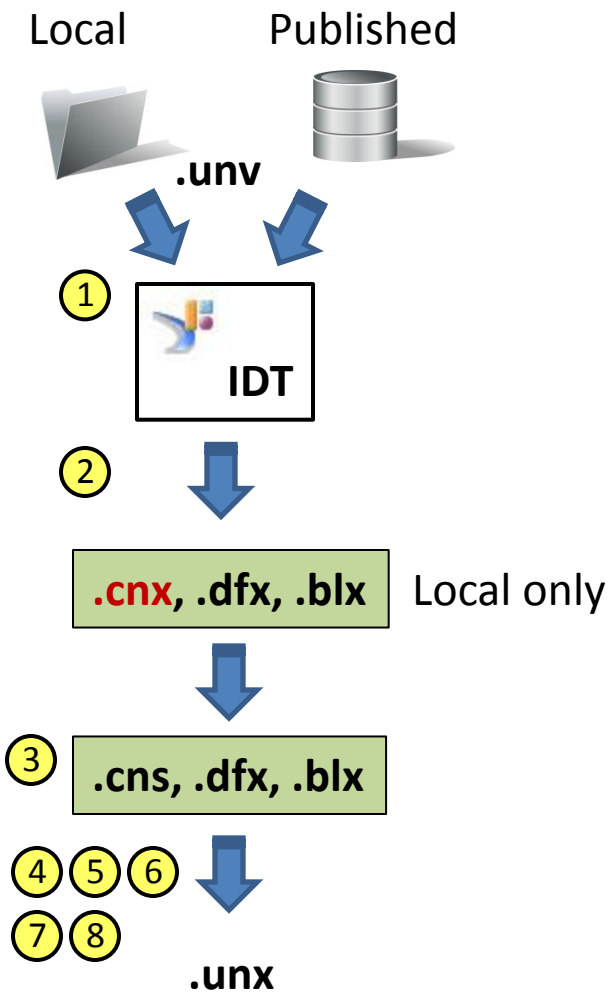
- Two different approaches to converting legacy universes
- Select the migration strategy based on where the universe is located
  - Approach #1: Universe exported to XI v3.1 Repository
  - Approach #2: Universe created in BI 4.0

#1: Legacy Universe in Previous Versions



1. Export any locally stored universes to the legacy repository (system database)
2. Use the Upgrade Management Tool (UMT) to migrate to BI 4.0
3. Convert .UNV to .UNX using the Information Design Tool (IDT)
4. Refresh structure of the Data Foundation layer (.dfx)
5. Run an Integrity Check without publishing
6. Fix any discrepancies (see next section)
7. Publish business layer

#2: Universe Created in BI 4.0



1. Select the legacy universe (published or local) using Information Design Tool (IDT)
2. Convert that universe using Information Design Tool (IDT)
3. Create secured connection (.cns) – local universe only
4. Adjust Data Foundation layer with secured connection – local only
5. Refresh structure of the Data Foundation layer (.dfx)
6. Run an Integrity Check without publishing
7. Fix any discrepancies (see next section)
8. Publish business layer

What's Converted?

- Schema
  - Tables
  - Aliases
  - Derived tables (including nested)
  - Joins (self joins → column filters)
  - Connections (local → .cns, secured → .cnx)

- Classes and Objects
  - Classes → folders
  - Dimensions
  - Measures
  - Details → Attibutes
  - Conditions → Filters
  - @ Functions (@Select, @Where, @Variable, @Derived_Table, @Aggregate_Aware)

What's Converted – cont'd?

- Universe Restrictions
  - Objects → Business security profile
  - All others → Data security profile
  - Derived tables (including nested)
  - Joins (self joins → column filters)

- Others
  - @Prompt → Optionally converted to named parameters (Business Layer only)
  - List of Values
  - Universe controls → same default restrictions in Business Layer (BL)

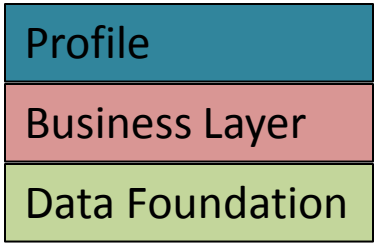
What's Not Converted ?

- Linked universes
  - More on this later in the presentation
- Custom Strategies
- Customized .PRM file values
- @Prompt in Joins
  - Self-joins only that refer to an object using a List of Values

Agenda

- Introduction
- Architecture
- Migration
- **Features**
- Publishing
- Other Considerations
- Wrapping Up

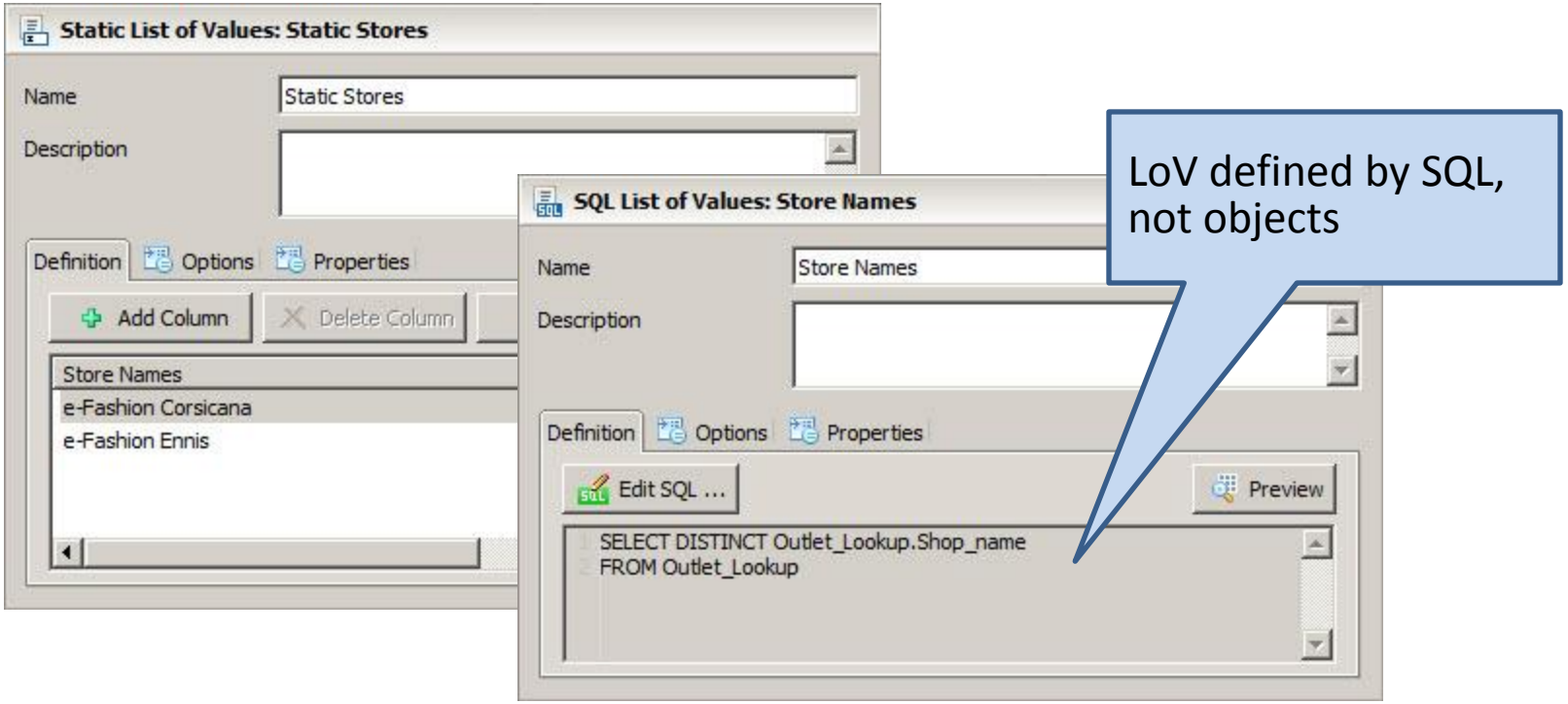
Features Overview



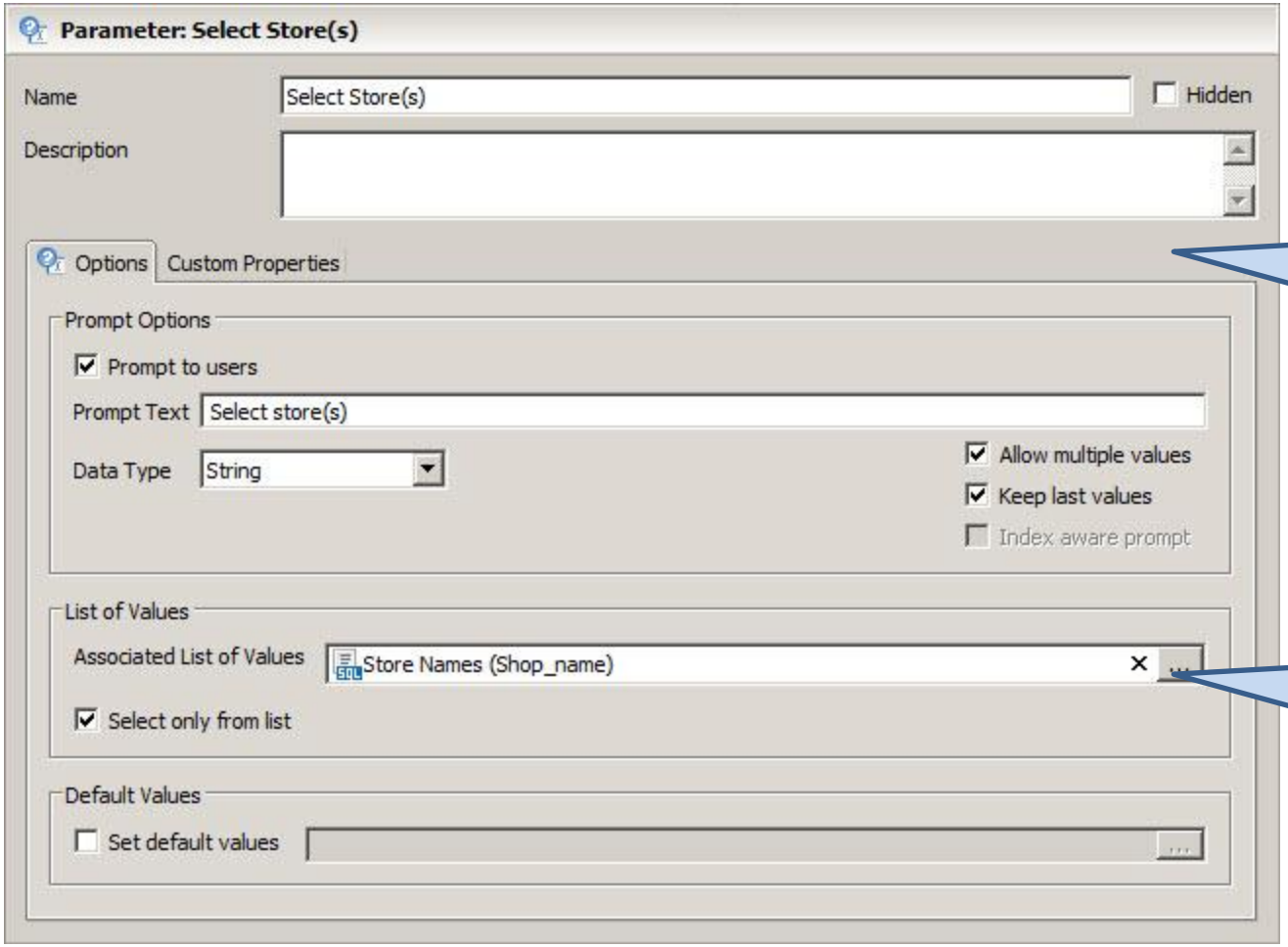
- Features will be discussed by layer
  - Data Foundation
  - Business Layer
  - Profiles
- Advice is given for each feature
  - When to use
  - Whether to use
- All recommendations are based on my experience
  - Your mileage may vary …

List of Values (LoV)

- Can be created in the Data Foundation
  - Inherited by all Business Layers based on that foundation
  - Defined by custom SQL or static values



Parameters

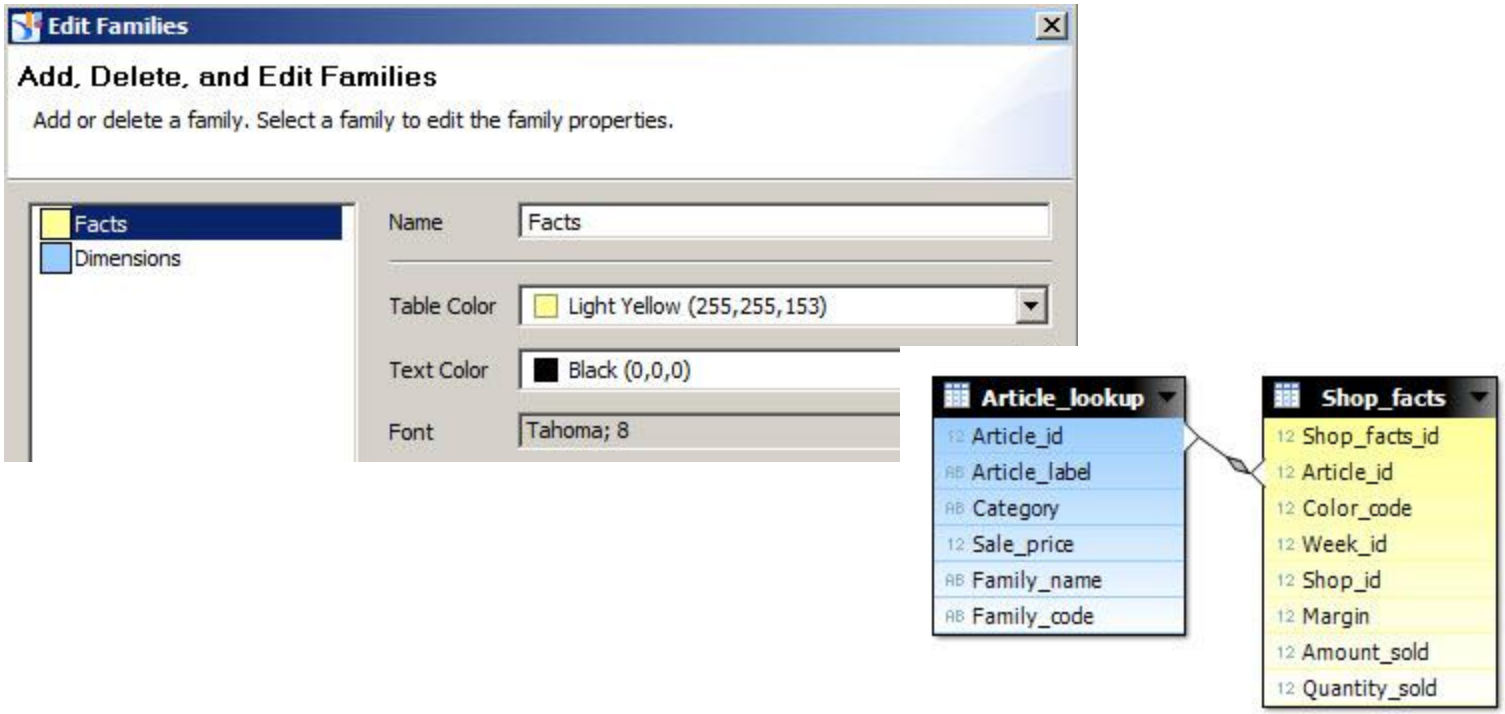


Parameters can be inherited by all Business Layers

Can use LoV defined in the foundation

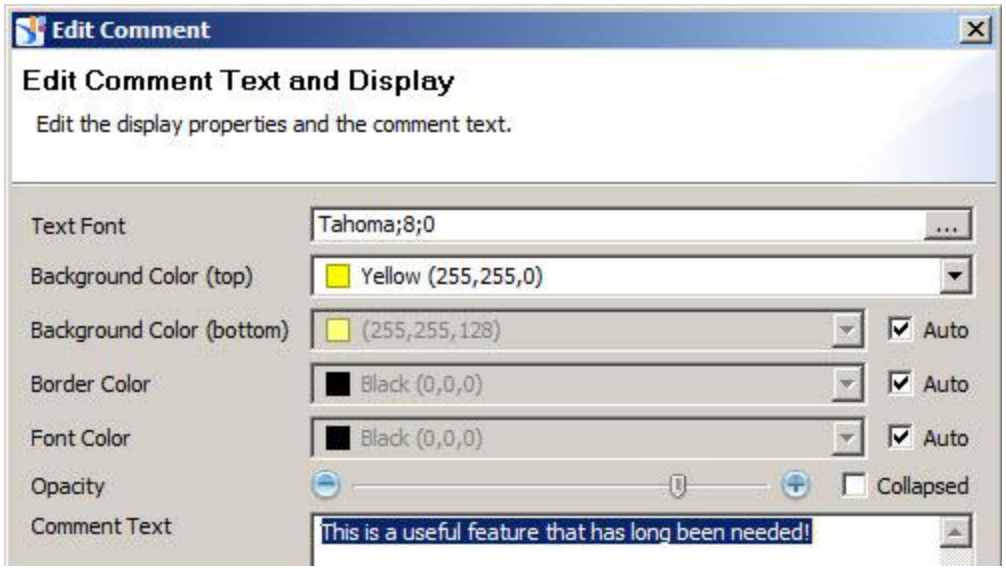
Families

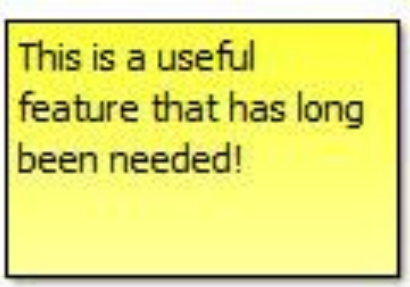
- Tables can be grouped by type
  - Select a color for every family
  - Defined by custom SQL or static values



Comments

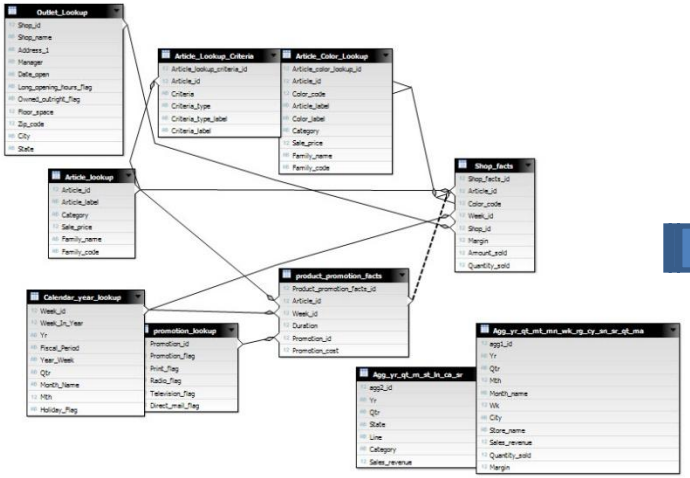
- Now added like tables or joins
  - Can be created, formatted, and MOVED
  - Great for annotating the foundation



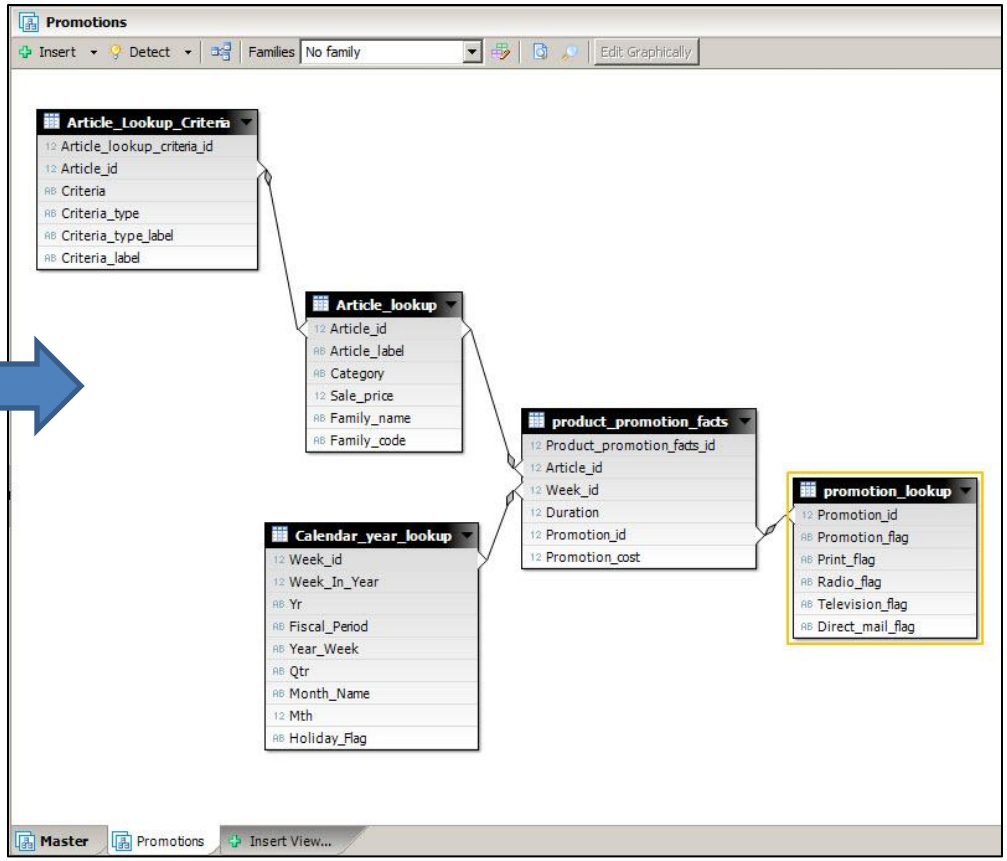


Custom Views

- Easily separate sections of a complex schema
  - Does not affect SQL that is generated
  - Alternate view

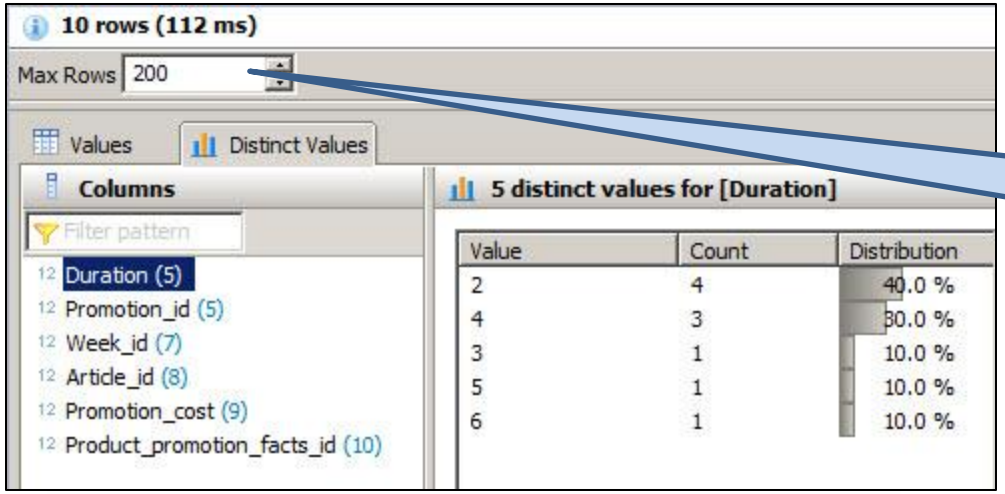


Master View

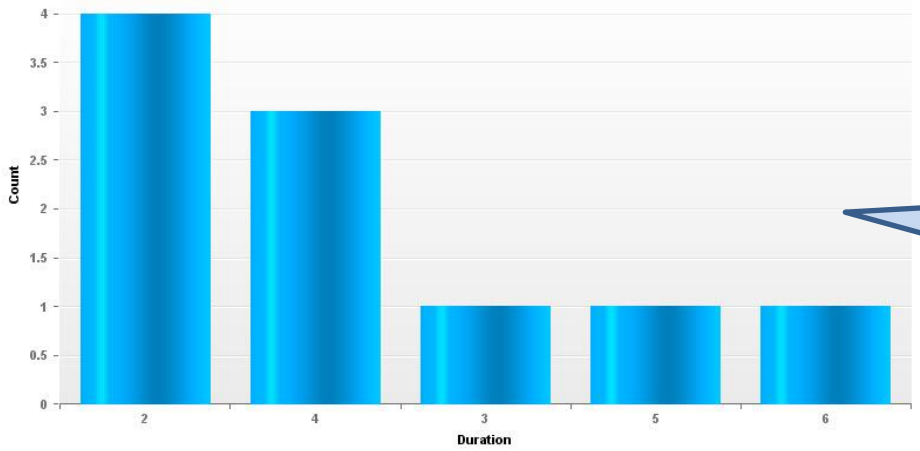


Data Profiling

- Data can be profiled per column



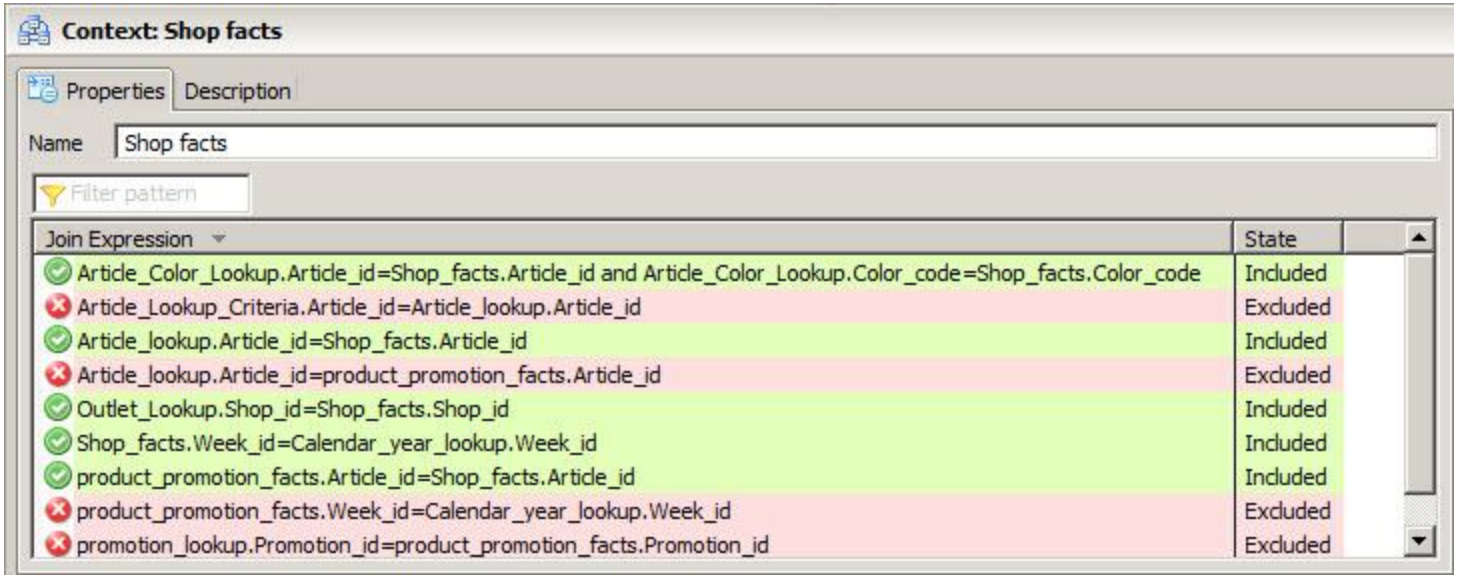
Adjust this value higher for more meaningful distributions



Value distribution can be shown as a Bar or Pie chart

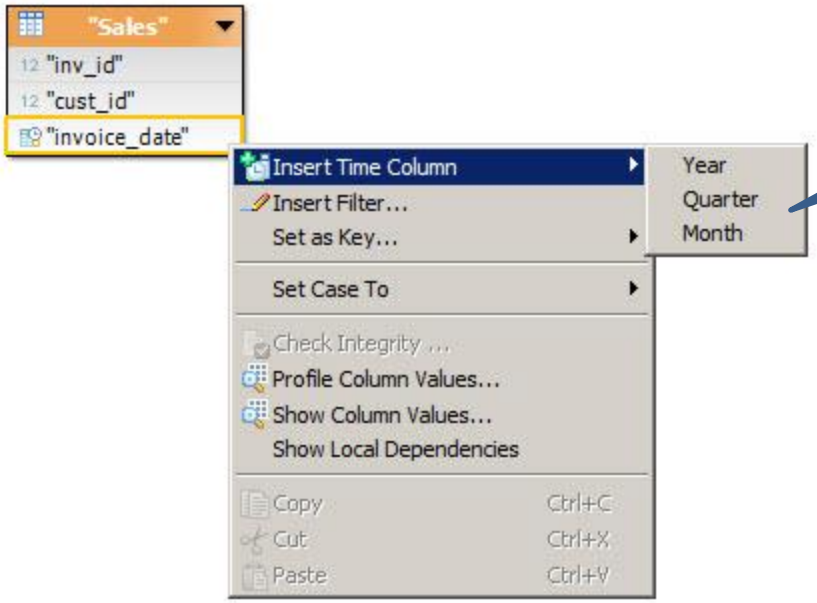
Contexts

- Joins in a context can be:
  - Included (part of the context)
  - Excluded (never part of the context)
  - Neutral (implicitly included)
- New joins that are added after contexts are defined are neutral
  - Always included in all contexts
  - Makes join and context maintenance a bit easier



Calculated Columns

- New columns can be added to existing tables
  - Calculated = consisting of columns from the same table
- Special example of this are time-part columns
  - IDT automatically creates the column for you
  - Only works for DateTime columns

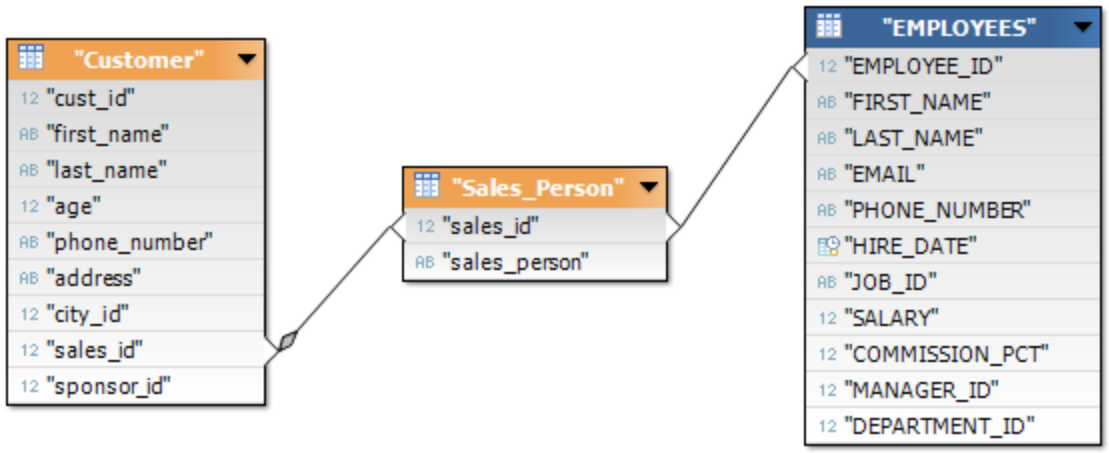


Different from XI 3.1?

Absolutely! Created in the **DATA** schema, not as objects

Federation

- Foundations can be based on multiple connections
  - Must decide BEFORE data foundation is created
- Special rules in effect for the resulting foundation (and business layer)
  - Joins between tables of two or more sources must be ANSI-92
  - Same goes for calculated columns, derived tables
  - Universes against SAP BW, SAS require multisource foundations



Federation, cont'd

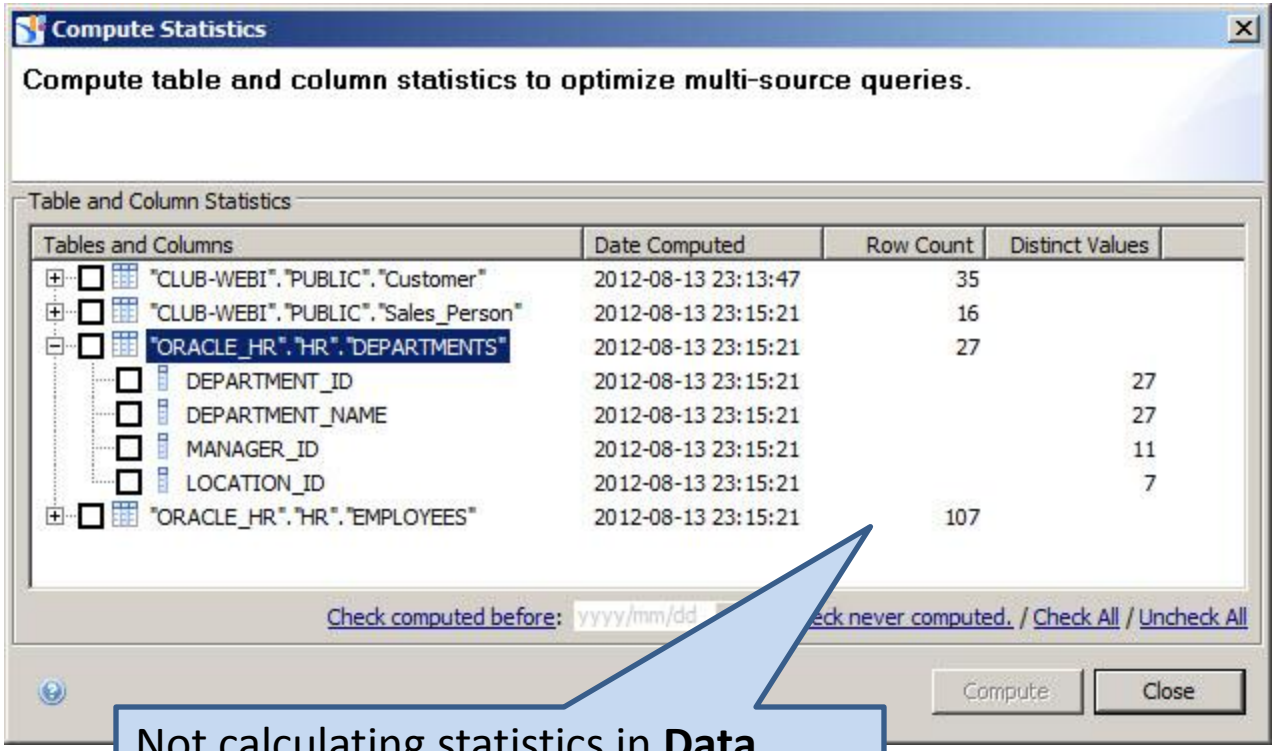
- But what about database-specific functions
  - Analytical functions
  - DECODE
- Create DB-specific derived tables, calculated columns

Should I Federate?

- This is cool!
- Any downside?
- A few things to consider …
  - Based on Data Federator technology
  - It must logically combine data from all sources
  - Tries to push down logic to each data source
  - Not good for large data sets per connection
  - Could bottleneck
- Recent IDT features help
  - Compute Statistics on resulting universe

Compute Statistics

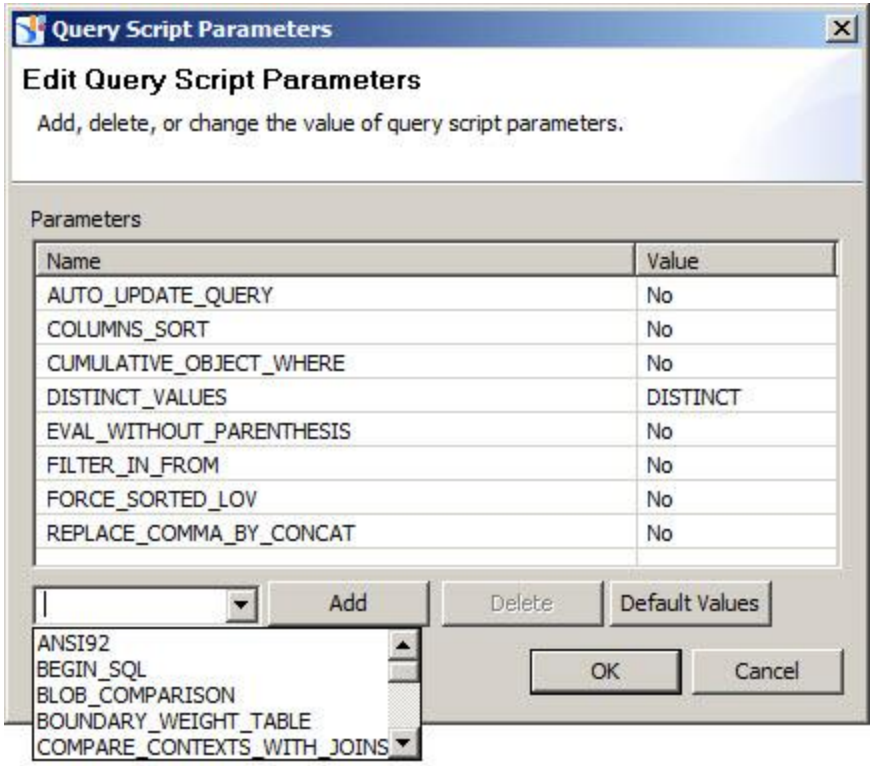
Business Layer



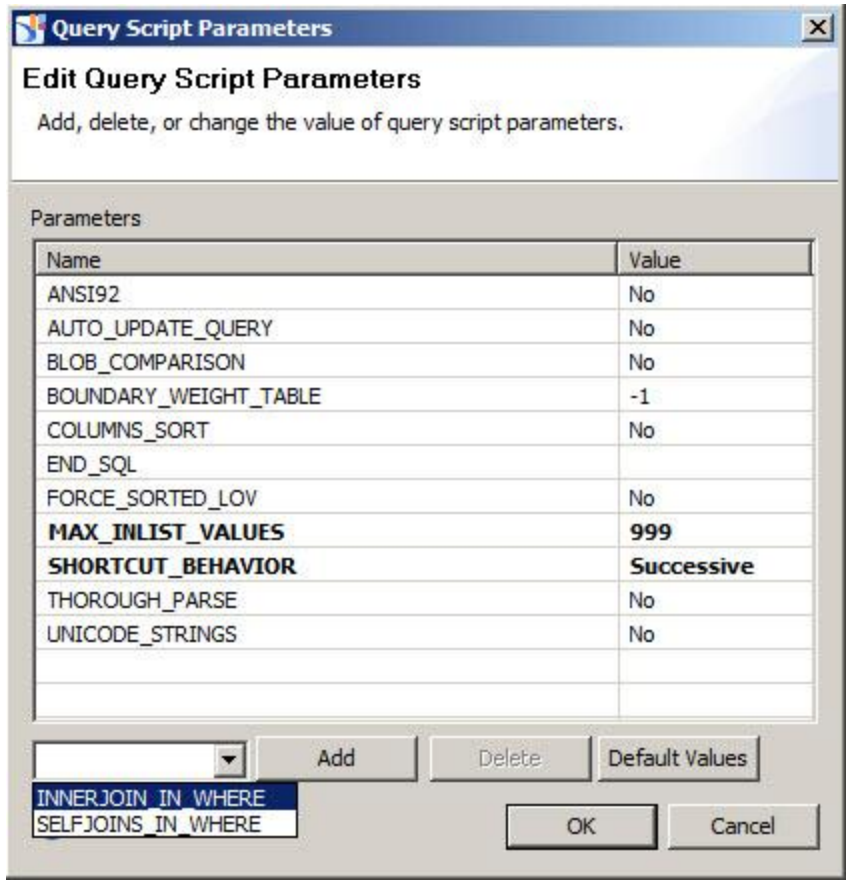
Not calculating statistics in **Data Federator** robbed many projects of maximum performance

SQL Parameters

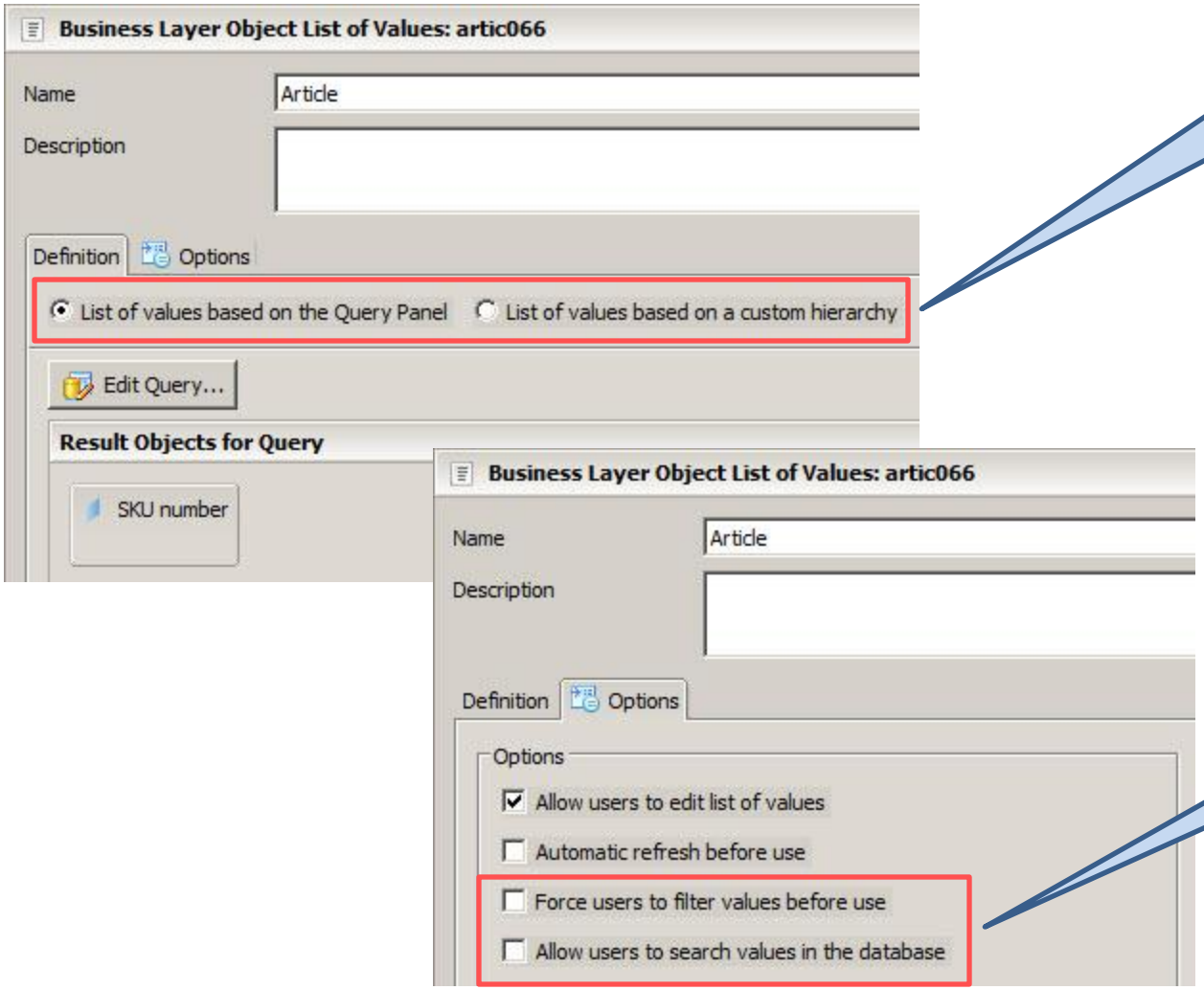
Business Layer



Data Foundation



List of Values



Traditional ways to define the list

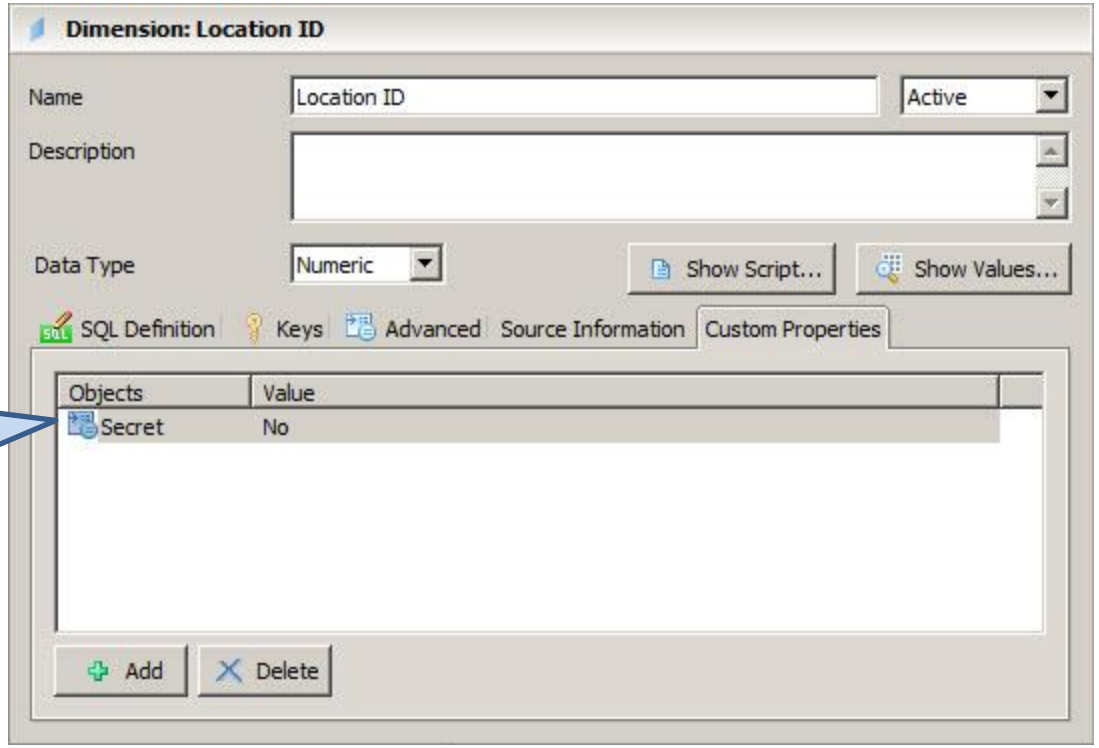
New options …

Custom Properties

- Additional attributes can be recorded per object
  - Attribute and value added
  - Attributes for universes, folders, objects

Can't easily retrieve this information through reporting apps **<yet>.**

SDK-based retrieval a possibilty



@Functions

- @Execute
  - Added in BI 4.0 SP04
  - Runs a List of Values (LOV) query
  - Used to add LOV results as a filter or WHERE clause
  - List of Values defined in Data Foundation or Business Layer

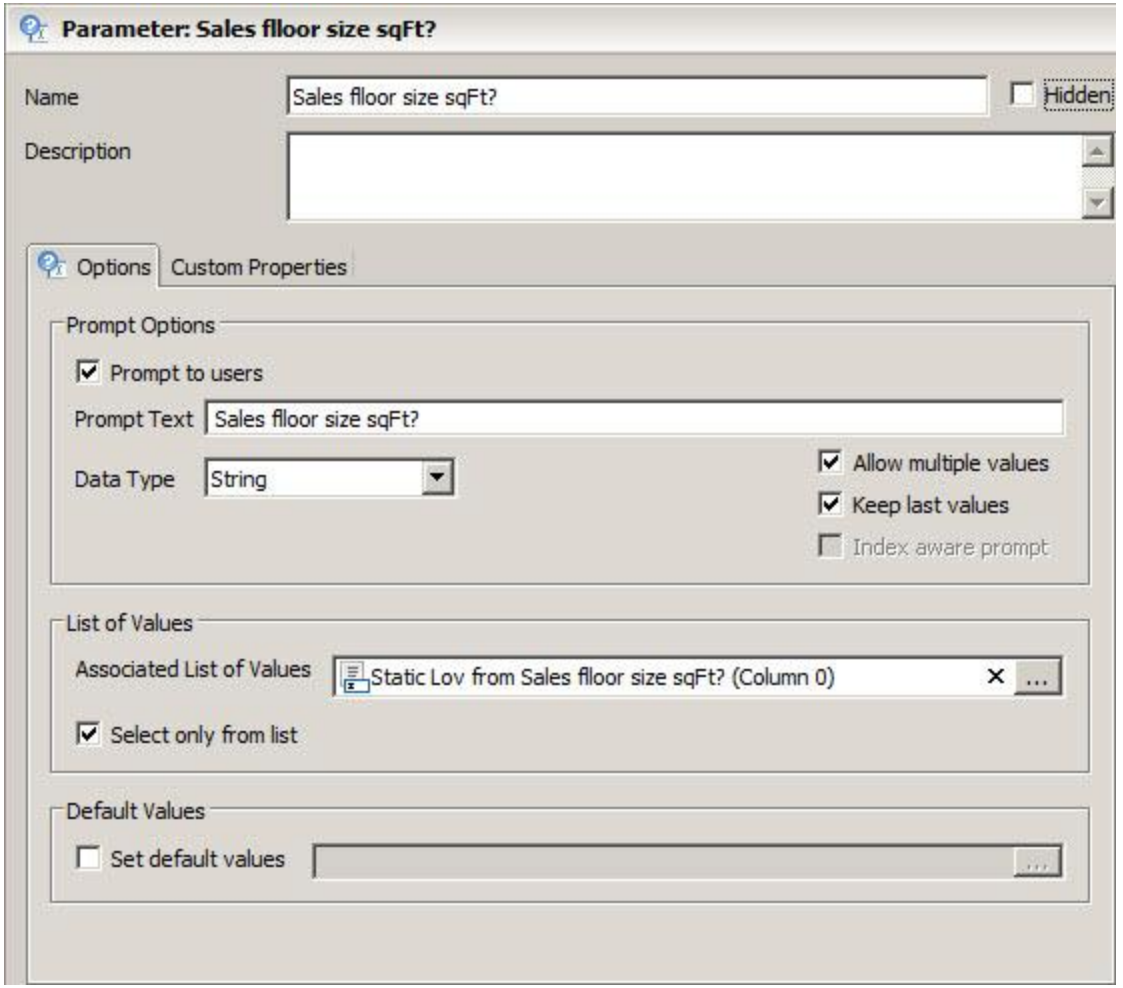
WHERE Customers.last_name IN **@Execute(Big_Spenders)**

Big_Spenders:

```
SELECT
  c.last_name
FROM
  Customer c, Invoice_Line il, Service sv, Sales s
WHERE
  c.cust_id = s.cust_id and s.inv_id = il.inv_id and il.service_id=sv.service_id
GROUP BY c.last_name
HAVING sum( il.days * sv.price)  >  50000
```

Parameters

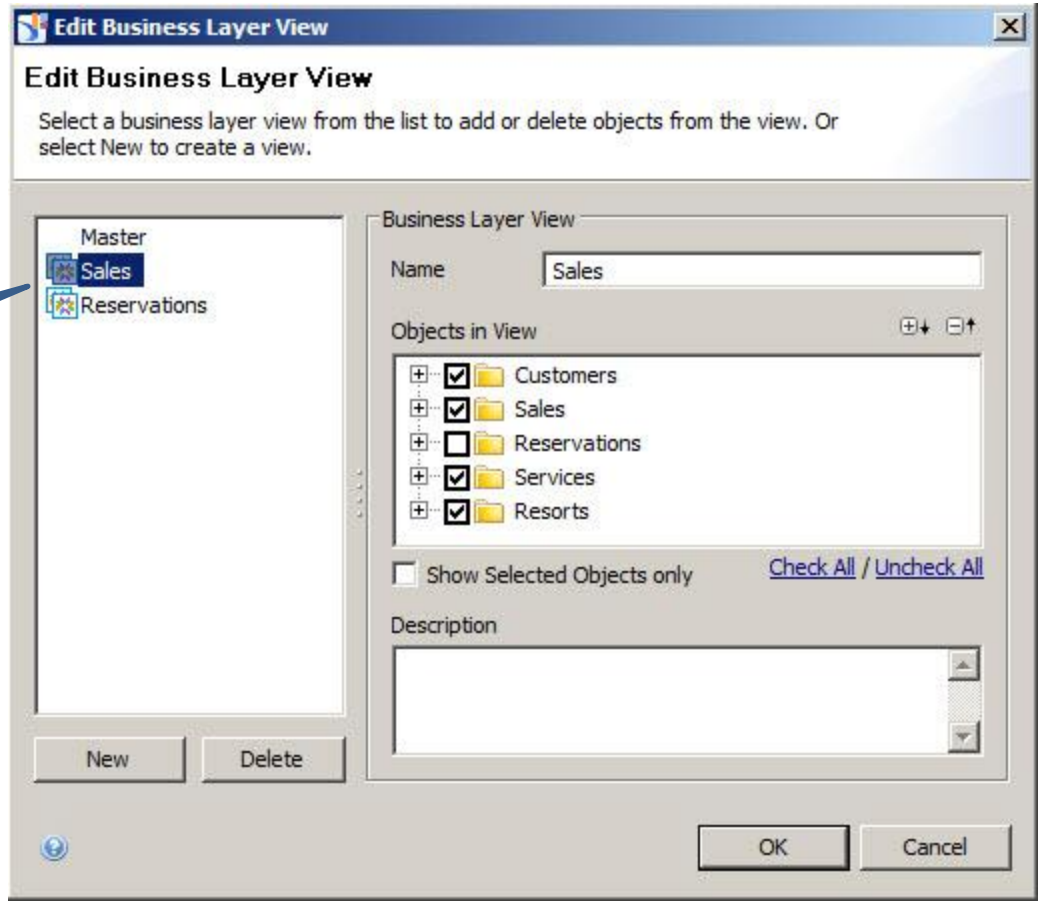
- Prompts can be named, saved, and inherited
  - Prompts were embedded in objects and joins in XI 3.1
  - Parameters are inherited from the Data Foundation layer



Views

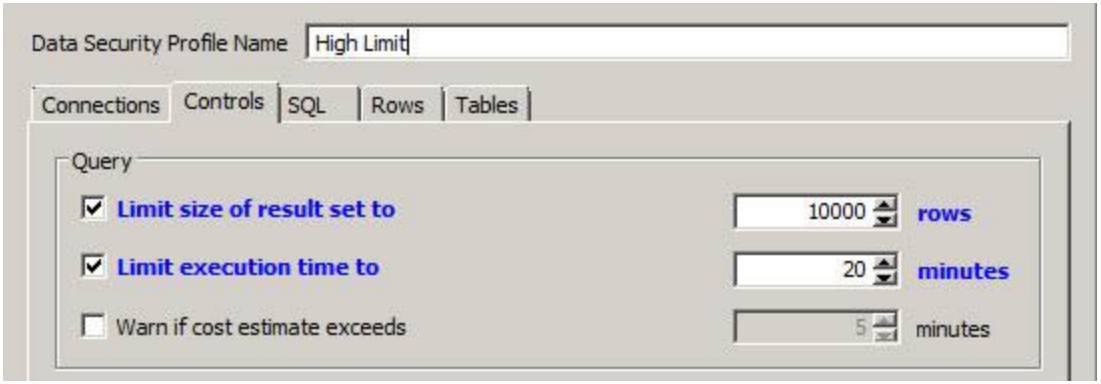
- Views represent selected classes and objects
  - Similar to Object universe restrictions in XI 3.1
  - Those restrictions **HID** objects
  - 4.0 Views **SHOW** selected objects

These views can be referenced by Business Security Profiles



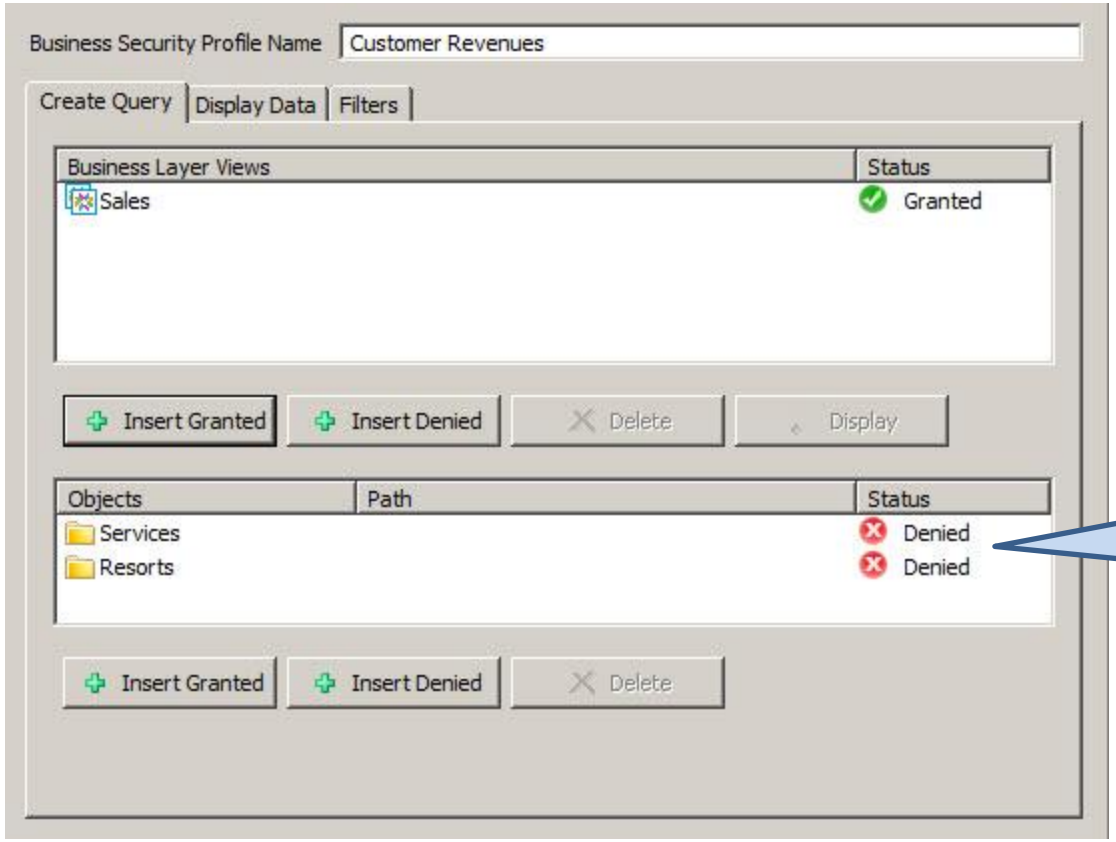
Data Security Profile

- Adjust various universe aspects by user or group
  - <Almost> the same as XI 3.1 universe restirctions
  - **NOTE**: Object restriction is missing!!



Business Security Profile

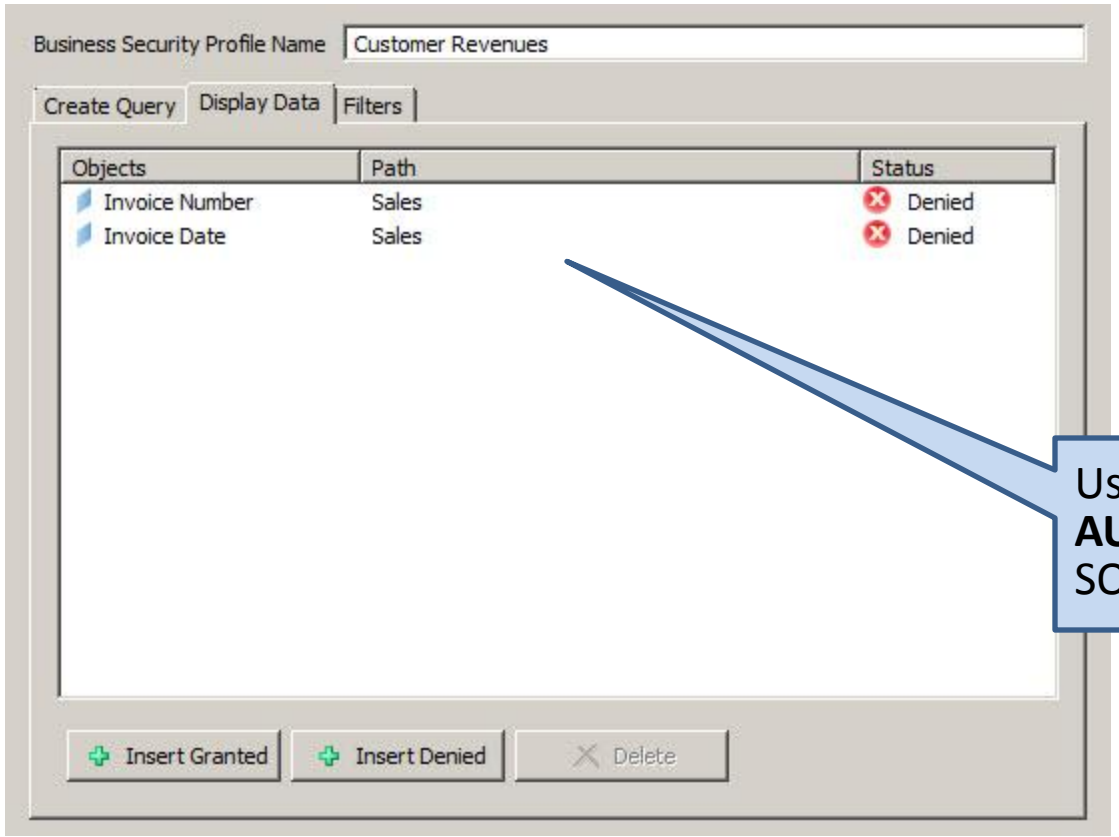
- Much more powerful
  - Allows classes / objects to be used OR restricted



Universe views can be fine-tuned here

Business Security Profile, cont'd

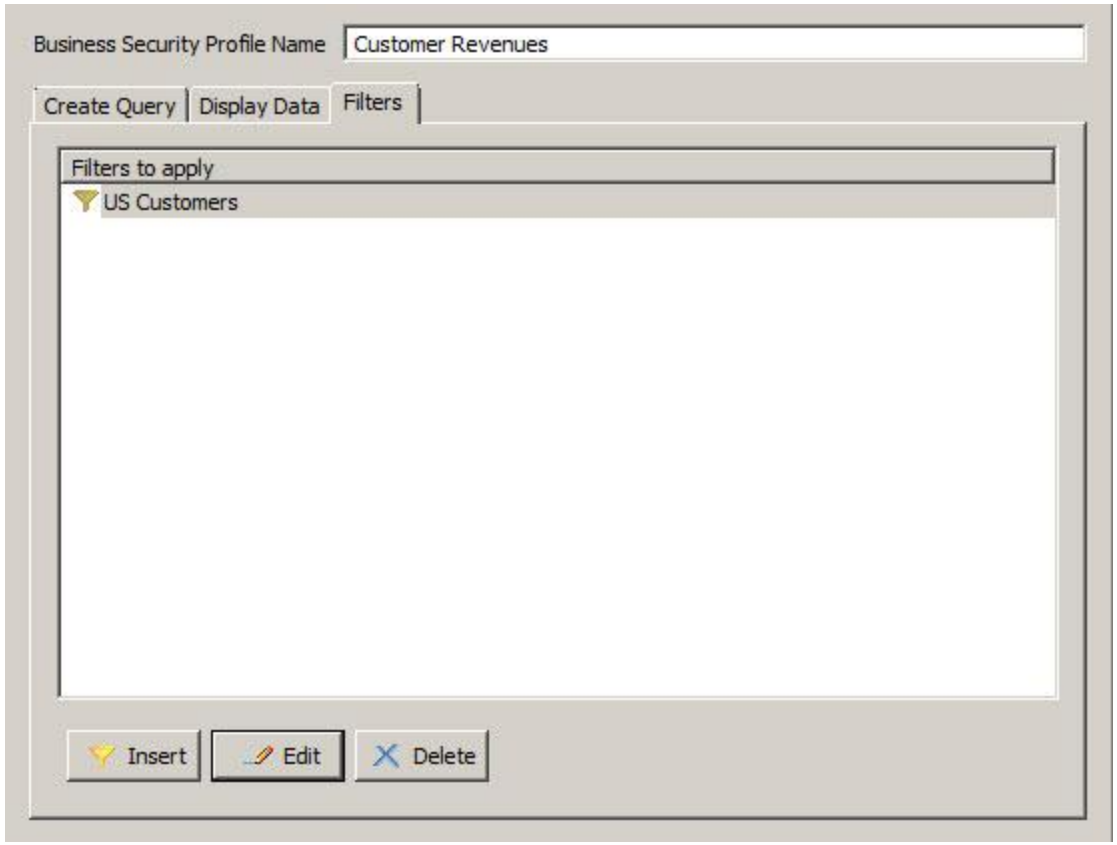
- Data can be displayed … or not
  - Very useful for hiding confidential / secret information



Use with **AUTO_UPDATE_QUERY** SQL parameter

Business Security Profile, cont'd

- Data can be filtered as well
  - Filters added to the resulting SQL Query

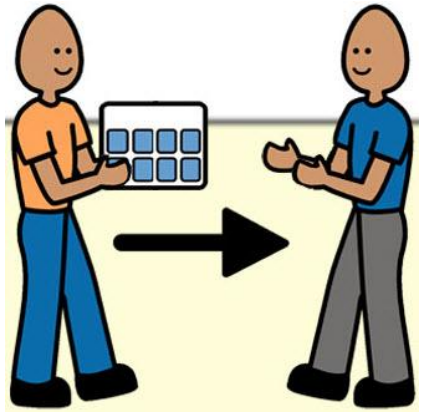


Agenda

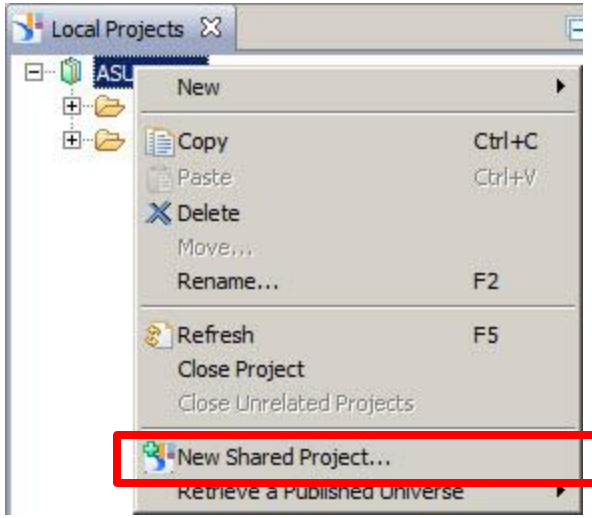
- Introduction
- Architecture
- Migration
- Features
- **Publishing**
- Other Considerations
- Wrapping Up

Share your Solution … In Pieces!

- Developers can share projects
  - Layers can be uploaded to repository (synchronized)
    - Connections
    - Data Foundations
    - Business Layers
  - Many developers can contribute to final project
  - Methods available to resolve discrepancies
  - In XI 3.1, sharing done on a connection and universe basis



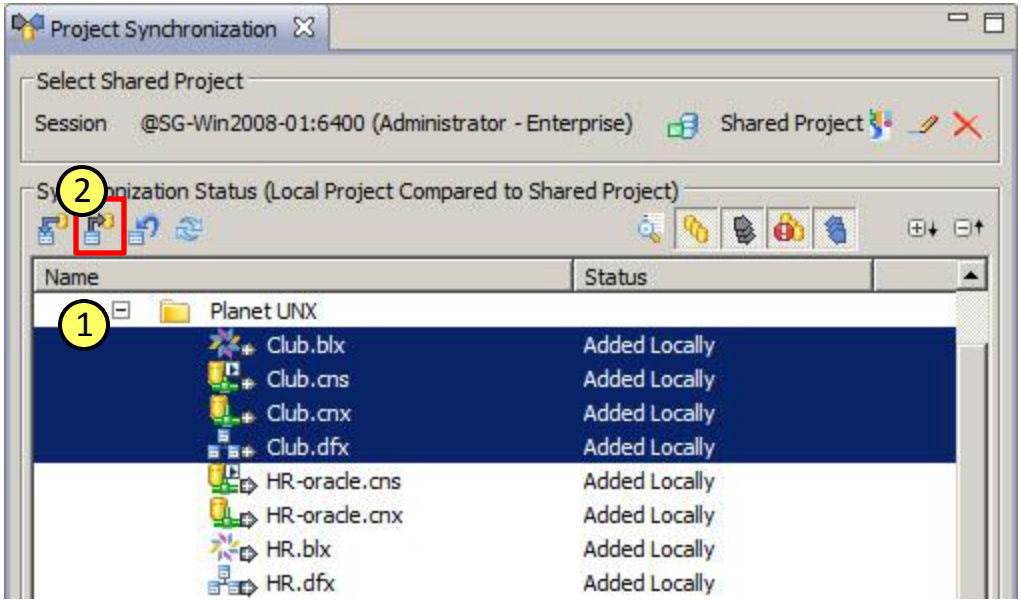
Sharing a Project



Right-click on any project within the Local Projects pane to share

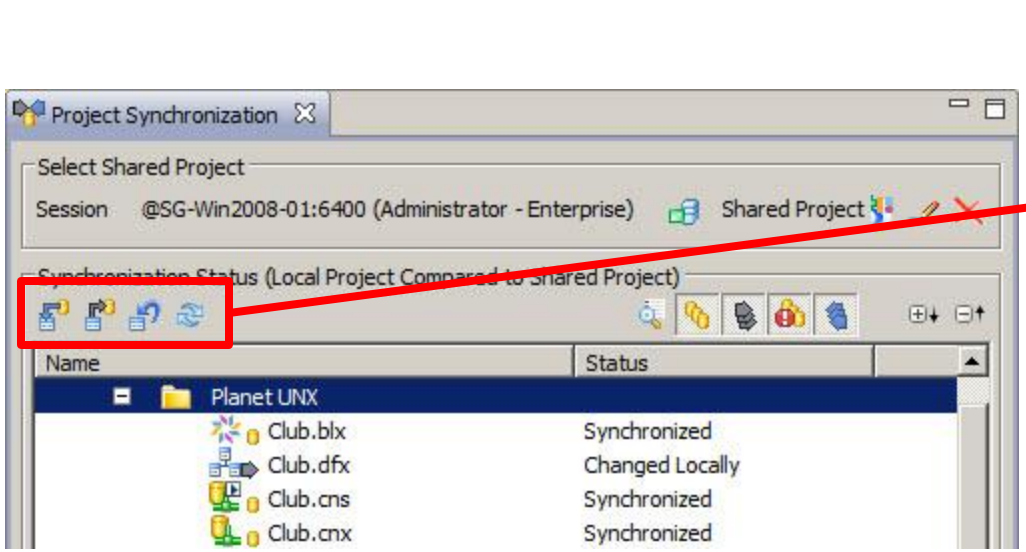
Synchronizing a Project

- Layers within the shared project not shared
  - Highlight the layer to be shared
  - Use the **Save Changes on Server** button to synchronize



Resolving Discrepancies

- Projects can easily get out of synch
  - Local definitions will change as work progresses
  - Other developers may post modified layer
- Use Project View controls to adjust the published project



Get changes from server

Save changes on server

Revert changes

Refresh

Agenda

- Introduction
- Architecture
- Migration
- Features
- Publishing
- **Other Considerations**
- Wrapping Up

Other Considerations

Agenda

- Introduction
- Architecture
- Migration
- Features
- Publishing
- Other Considerations
- **Wrapping Up**

Wrapping Up

Thank you for participating.

Please provide feedback on this session by completing a short survey via the event mobile application.

SESSION CODE: 0611

Learn more year-round at www.asug.com