Welcome to the Lists of values and Parameters lesson for Information design Tool in
SAP BusinessObjects 4.0.

The aim of this lesson is to introduce you in the new design (or authoring) of Lists of
values and Parameters (also named Prompts in the Reporting and Analysis tools) from
the Information Design Tool 4.0

Indeed, in Business Intelligence 4.0, Lists of values and Parameters have fully been
revisited in order to promote their sharing (as metadata entity) and usability.
List of values and Parameter objects: Lesson Objectives

After completing this lesson, you will know:

- The fundamental concepts of List of values and Parameter objects
- The main workflows to define List of values and Parameter objects
- How to use and share List of values and Parameter objects
1. Introducing the Lists of values
2. Defining a static list of values
3. Defining a custom list from BO objects using the Query Panel
4. Defining a hierarchical list from a custom hierarchy
5. Defining a dynamic list from a custom SQL
6. Using and Sharing a list of values across many objects
7. Introducing the Parameter objects
8. Defining a parameter object
9. Defining dependent parameter objects
10. Using and sharing parameter objects inside universes and across queries
By definition, a List of values (LOV) is an entity containing one or more values that can be specified for a parameter (or prompt) or an object (used in filters).

In Business Intelligence 4.0, the list of values is one of the features that have fully been revisited to enrich it but also to make it shareable in multiple places: the final goal is to make them available outside of its universe in future releases.

So, Lists of values are now standalone metadata, but lists of values defined in previous universes version are supported and migrated as it.

With Lists of values managed as metadata, you are able to:
- Duplicate LOV
- Remove LOV
- Check LOV integrity
- Show LOV impact analysis
- Show/Hide LOV.

Users can create multiple lists of values in the Business Layer and also in the Data Foundation.

In Business Intelligence 4.0, we support several types of list of values:
- Static Lists of values
- Dynamic Lists based on queries on top of business objects
- Dynamic hierarchical Lists based on custom hierarchies
- Dynamic Lists based on free hand SQL (only available for RDBMS data sources)
- Dynamic Lists based on stored procedures (only available for RDBMS data sources)
Introducing the List of values (2)

Common options for all LOV Types. Some options are not available according to the LOV types.
1. Introducing the List of values

2. **Defining a static list of values**

3. Defining a custom list from BO objects using the Query Panel

4. Defining a hierarchical list from a custom hierarchy

5. Defining a dynamic list from a custom SQL

6. Using and Sharing a list of values across many objects

7. Introducing the Parameter objects

8. Defining a parameter object

9. Defining dependent parameter objects

10. Using and sharing parameter objects inside universes and across queries
Introducing the static list of values

The static list of values allows users to specify quickly and easily value set stored in the Universes or Data Foundations.

Contrary to the dynamic lists of values, the values of static LOV are stored as grid tables in the parent container such as the Universe or the Data Foundation.

The static LOV can be associated to objects or parameters; and a it has the same properties as any other list of values.

In Business Intelligence 4.0, users are able to create a static list with multiple columns.

There are two ways to create a static list of values:
- Manually create the columns of the static list of values and then manually enter the values
- Import the values from an external file
From the Import File Wizard, users can import text files with formats such as TXT, CSV, ASC and PRM. You must first specify file path and file data settings (such as the data separator, the text delimiter and the data formats). Click on OK to import file data into the static list according to you import settings.

From the Import File Wizard, users can import text files with formats such as TXT, CSV, ASC and PRM.

The Import File Wizard allows users to specify:
- the file connection i.e. the file to import
- the file data settings i.e. set of settings specifying how data have been write in the file
- import settings i.e. set of settings specifying how file data will be imported in the list
Users can access to the static list structure in order to:

- Change the column name and data type
- Hide a column for end-users
- Set a column as the key value provider of another one
How to create a static list

To create a static list, go to the **List of values** pane; click on the insert list button and select **Static List of values**

Users can create static lists of values in the business layer (for Relational and OLAP data sources) and also in the data foundation.
1. Introducing the List of values
2. Defining a static list of values
3. **Defining a custom list from BO objects using the Query Panel**
4. Defining a hierarchical list from a custom hierarchy
5. Defining a dynamic list from a custom SQL
6. Using and Sharing a list of values across many objects
7. Introducing the Parameter objects
8. Defining a parameter object
9. Defining dependent parameter objects
10. Using and sharing parameter objects inside universes and across queries
Users can use the Query Panel tool to custom rich lists of values on the top of business objects in universes. With the Query Panel tool, users can:

- include additional result objects in order to get a multi-column list
- apply data filters using custom or shared filters and using custom or shared prompts
- apply query sorts in order to order values

Like in XI 3.0, users can use the Query Panel to enrich lists of values on the top of business objects in universes.
With the Query Panel for the list definition, users are able to:

- include additional result objects in order to get a multi-column list
- apply data filters using custom or shared filters and using custom or shared prompts
- apply query sorts in order to order values
- combine queries with UNION/INTERSECT/MINUS operators
Users can only create custom lists of values on the top of business objects in the business layer (for Relational and OLAP data sources). The authoring of this LOV type is not supported in the data foundation container.
1. Introducing the List of values
2. Defining a static list of values
3. Defining a custom list from BO objects using the Query Panel
4. **Defining a hierarchical list from a custom hierarchy**
5. Defining a dynamic list from a custom SQL
6. Using and Sharing a list of values across many objects
7. Introducing the Parameter objects
8. Defining a parameter object
9. Defining dependent parameter objects
10. Using and sharing parameter objects inside universes and across queries
Introducing the hierarchical list from a custom hierarchy

Users can build hierarchical list (i.e. with a tree display) from custom level-based hierarchies. The custom hierarchies are created from dimensions used as hierarchy levels.

This is a new type of list of values that allow creating a hierarchical list of values.

To create such list of values, you just have to select which business objects will be part of this list of values and reorder the levels of the hierarchy if needed.

As for other list of values you can preview its content.
For Relational Data Sources, the consumption of this List type in query filters or query filter prompts carry out two new features (not available on OLAP data sources):

- Users are able to select values at any hierarchy level
- In order to preserve the selection uniqueness, SL generates SQL including ancestor levels of selected nodes in the WHERE clause combined with sub queries

Note that these features are not available for OLAP data sources.
Users can only create hierarchical lists of values based on custom hierarchies in the business layer (for Relational and OLAP data sources). The authoring of this LOV type is not supported in the data foundation container.
1. Introducing the List of values
2. Defining a static list of values
3. Defining a custom list from BO objects using the Query Panel
4. Defining a hierarchical list from a custom hierarchy
5. **Defining a dynamic list from a custom SQL**
6. Using and Sharing a list of values across many objects
7. Introducing the Parameter objects
8. Defining a parameter object
9. Defining dependent parameter objects
10. Using and sharing parameter objects inside universes and across queries
Introducing the dynamic list from a custom SQL

Advanced users can create dynamic lists of values in writing custom SQL. This LOV type allows them to benefit from the SQL vendor features and benefits for the LOV.

As in BOE XI 3.x, it is possible to write a list of values using SQL expressions that return one or more columns.

Once the SQL written, the user needs to refresh the list of values structure in order to display the available columns.
The SQL Editor button takes you to the SQL editor. One can drag and drop items from the panes below the expression. Click on Validate to check your custom expression. Click on to build quickly the SQL.

Users can use the SQL assistant to edit its SQL expression
Users can create SQL lists of values in the business layer (only for Relational) and also in the data foundation.

You cannot create this LOV type from the OLAP data sources.
1. Introducing the List of values
2. Defining a static list of values
3. Defining a custom list from BO objects using the Query Panel
4. Defining a hierarchical list from a custom hierarchy
5. Defining a dynamic list from a custom SQL
6. Using and Sharing a list of values across many objects
7. Introducing the Parameter objects
8. Defining a parameter object
9. Defining dependent parameter objects
10. Using and sharing parameter objects inside universes and across queries
By default, the Business objects are associated with their default LOV.

Users can associate them LOV metadata

When the list of values is associated to an object or a prompt it is possible to select which column values will be sent to the database.
Sharing a List across many Business Objects and Parameters

A same list of values can be share across many objects and parameters with different mapping columns and runtime behaviors

- Hierarchical LOV

- Multi-column LOV

As metadata, a same list of values can be share across many objects and parameters with different mapping columns and runtime behaviors
1. Introducing the List of values
2. Defining a static list of values
3. Defining a custom list from BO objects using the Query Panel
4. Defining a hierarchical list from a custom hierarchy
5. Defining a dynamic list from a custom SQL
6. Using and Sharing a list of values across many objects

7. **Introducing the Parameter objects**
8. Defining a parameter object
9. Defining dependent parameter objects
10. Using and sharing parameter objects inside universes and across queries
A Parameter object is metadata that requires answers before the system continue action. The parameter answers can be provided by interactions with end-users; or programmatically.

By definition, a Parameter object is metadata that requires answers before the system continue action. The parameter answers can be provided by interactions with end-users (this case corresponds to the prompts); or programmatically.

As Lists of values, in Business Intelligence 4.0 Parameter object (named prompts in Reporting tools) is one of the features that have fully been revisited to enrich it but also to make it shareable in multiple places: the final goal is to make them available outside of its universe in future releases.

So prompts are now standalone metadata, but prompts defined in previous universes version are supported and migrated as it.
1. Introducing the List of values
2. Defining a static list of values
3. Defining a custom list from BO objects using the Query Panel
4. Defining a hierarchical list from a custom hierarchy
5. Defining a dynamic list from a custom SQL
6. Using and Sharing a list of values across many objects
7. Introducing the Parameter objects
8. Defining a parameter object
9. Defining dependent parameter objects
10. Using and sharing parameter objects inside universes and across queries
It is now possible to create prompts without writing any code respecting, in some cases, a advanced syntax.

Now an editor provides all the existing prompt capabilities with the ability now to choose any of the available lists of values.
As new feature, the parameter designer can also provide the parameter answers for the runtime in unselecting the ‘Prompt to users’ checkbox option. In this case, he must provide the runtime answer values that will be automatically used by the system at the runtime without prompting. The runtime answer values can be:

- **Constants**
  - You must provide runtime values
  - You can use BO System variables as runtime values

- **Variables**
  - Set runtime value(s): Variable('BOUSER')

As new feature, the parameter designer can also provide the parameter answers for the runtime in unselecting the ‘Prompt to users’ checkbox option. In this case, he must provide the runtime answer values that will be automatically used by the system at the runtime without prompting. The runtime answer values can be:

- Constants
- Variables
Agenda

1. Introducing the List of values
2. Defining a static list of values
3. Defining a custom list from BO objects using the Query Panel
4. Defining a hierarchical list from a custom hierarchy
5. Defining a dynamic list from a custom SQL
6. Using and Sharing a list of values across many objects
7. Introducing the Parameter objects
8. Defining a parameter object
9. **Defining dependent parameter objects**
10. Using and sharing parameter objects inside universes and across queries
The Universe Parameter $P_A$ depends on the Universe Parameter $P_B$ if the List of values $L_A$ of $P_A$ depends on the $P_B$ answers.

Design time:
- Region Parameter
  - Associated list of values
- Country Parameter
  - Associated list of values

Runtime:
- Prompt summary
  - Enter Region: Not answered
  - Enter Country: Not answered
- Country Parameter
  - Select Country Parameter

The LOV associates to Country Parameter has a filter using the Region Parameter.

At the runtime, users must enter answer for Region Parameter before providing answering the Country Parameter answers from its LOV.

The Universe Parameter $P_A$ depends on the Universe Parameter $P_B$ if the List of values $L_A$ of $P_A$ depends on the $P_B$ answers.
Agenda

1. Introducing the List of values
2. Defining a static list of values
3. Defining a custom list from BO objects using the Query Panel
4. Defining a hierarchical list from a custom hierarchy
5. Defining a dynamic list from a custom SQL
6. Using and Sharing a list of values across many objects
7. Introducing the Parameter objects
8. Defining a parameter object
9. Defining dependent parameter objects
10. Using and sharing parameter objects inside universes and across queries
Users can use Parameter objects:

- In the binding expressions of universes (Select and Where object parts) or Data Foundations (Joins, etc.)

  You can use an universe parameter as prompt in binding expressions for RDBMS and OLAP universes

- In the new Business Filter Types (see Universe sessions for more details about the Business Filters)

  You can type or drag and drop a parameter object into binding expressions

Users can use Parameter objects:

- In the binding expressions of universes (Select and Where object parts) or Data Foundations (Joins, etc.)
- In the new Business Filter Types (see Universe sessions for more details about the Business Filters)
Users can use the universe parameter objects from ad-hoc queries using the Query Panel.
List of values and Parameter objects: Lesson Summary

After completing this lesson, you are now able to:
- The fundamental concepts of List of values and Parameter objects
- The main workflows to define List of values and Parameter objects
- How to use and share List of values and Parameter objects
Thank you for attending.