

Microsoft Dynamics® AX

An introduction to surrogate key replacement in Microsoft Dynamics AX 2012

White Paper

This paper defines surrogate keys in the context of Microsoft Dynamics AX and describes how they can be displayed to the user in the client.

October 2012

www.microsoft.com/dynamics/ax

Andy Stach, Senior Development Engineer, Microsoft Dynamics AX Client

Send suggestions and comments about this document to adocs@microsoft.com. Please include the title with your feedback.



Table of Contents

| | |
|--------------------------------------------------------------------------------------------------------------|-----------|
| Introduction..... | 3 |
| Terminology used in this paper..... | 3 |
| How are surrogate foreign keys modeled in Microsoft Dynamics AX 2012? | 4 |
| What problems do surrogate keys present to the UI? | 5 |
| Displaying surrogate keys..... | 5 |
| Editing surrogate foreign keys | 5 |
| MorphX objects and X++ classes | 6 |
| Tables | 6 |
| AutoIdentification field group | 6 |
| Forms..... | 7 |
| Reference Control | 7 |
| Reference Group control | 8 |
| Segmented Entry control | 8 |
| Reference Data Source | 8 |
| Reference Fields | 9 |
| Classes..... | 10 |
| SysReferenceTableLookup..... | 10 |
| FormRun | 10 |
| Examples | 10 |
| Example 1: Basic Reference Group end-to-end scenario..... | 10 |
| Key points | 14 |
| Example 2: Ambiguous data entry | 15 |
| Key points | 16 |
| Example 3: Overriding the default reference resolving behavior..... | 16 |
| Key points | 18 |
| Example 4: Adding a custom lookup..... | 18 |
| Key points | 19 |
| Example 5: Using a Reference Data Source to display additional read-only data from a related record | 20 |
| Key points | 20 |
| Appendix A: SharedProject_ReferenceGroupSample_1..... | 21 |
| Appendix B: SharedProject_ReferenceGroupSample_2..... | 30 |
| Appendix C: SharedProject_ReferenceGroupSample_3..... | 39 |
| Appendix D: SharedProject_ReferenceGroupSample_4..... | 49 |
| Appendix E: SharedProject_ReferenceGroupSample_5..... | 59 |

Introduction

A *surrogate key* is a system-generated value that acts as the primary key of a table. A surrogate key is typically used in place of a set of fields, called the *natural key*, which would normally act as the primary key of a table.

Surrogate keys have the following properties:

- The value is unique within an entity type, including its inheritance hierarchy.
- The value cannot be affected by the user.
- *The value is not visible to the user.*
- The value is system generated.

The **RecId** data type (64-bit integer) exhibits these properties and therefore is the surrogate key for Microsoft Dynamics AX. In this paper, surrogate keys exist on a *primary key table*.

A *surrogate foreign key* is a foreign key that relates one table to another by using a surrogate key. In simple terms, a surrogate foreign key can be thought of as a richer version of the **RefRecId** extended data type. In this paper, surrogate foreign keys exist on a *foreign key table*.

Terminology used in this paper

| Term | Definition |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Surrogate key | A system-generated value that acts as the primary key of a table. This value is typically used in place of a set of fields, called the natural key, which would normally act as the primary key of a table. |
| Natural key | A set of fields that uniquely identify a record and would have formed the primary key of the table, if not for the existence of a surrogate key. |
| Surrogate foreign key | A foreign key to a surrogate key. |
| Surrogate key replacement fields | A set of human-readable fields that are displayed in place of the surrogate foreign key. Rich client forms will use joined data sources to retrieve these fields at query time. The set of fields to use will be defined by a property on the new Reference Controls. This property will let the developer choose a field group, with AutoIdentification being the default choice. |
| Resolving (a user's entered value) | The act of determining which record the user's entered values resolve to. For example, the FirstName and LastName fields form the natural key of the Person table, and will act as the surrogate key's replacement on foreign key references. Furthermore, a record with values (RecId, FirstName, LastName) = (1, John, Smith) exists. If the user types John and Smith into a control that is replacing the surrogate key with human-readable values, the system will determine that they belong to the record identified by RecId = 1 . This act is considered the <i>resolution</i> of the entered values. |
| Surrogate foreign key replacement | The act of replacing a surrogate foreign key with the appropriate set of surrogate key replacement fields. |
| Explicit surrogate foreign key replacement | Surrogate foreign key replacement during Form design, when the developer purposely models a surrogate foreign key to be |

| Term | Definition |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | replaced with a set of fields. This can be done through Microsoft MorphX, X++ code, and so on. |
| Implicit surrogate foreign key replacement | Surrogate foreign key replacement during, for example, kernel generation of an AutoLookup or an AutoReport, or when the TitleField1 and TitleField2 properties are consumed. No developer action is required. |
| Foreign key table | The table that contains the definition of the surrogate foreign key relation linking two tables together. |
| Primary key table | The table that contains the surrogate key that a surrogate foreign key points to. |

How are surrogate foreign keys modeled in Microsoft Dynamics AX 2012?

Surrogate foreign keys are modeled in the Application Object Tree (AOT), under **Data Dictionary > Tables**.

The first thing to note when you attempt to create a surrogate foreign key is that the related primary key table must be using a surrogate key, as shown in Figure 1.

| | |
|--------------|--------------|
| PrimaryIndex | SurrogateKey |
| ClusterIndex | SurrogateKey |

Figure 1—Sample property sheet of a table using a surrogate foreign key

To create a surrogate foreign key, identify a target foreign key table—for example, `_ForeignKeyTable`—and browse to its **Relations** node. Right-click the **Relations** node, and then click **New Relation**. Open the property sheet, select a name for the Relation, and then set the **Table** property to the appropriate primary key table, as shown in Figure 2.

| Name | PrimaryKeyTable |
|-------------------------|-------------------------------|
| Table | <code>_PrimaryKeyTable</code> |
| Validate | Yes |
| EntityRelationshipRole | |
| RelatedTableCardinality | NotSpecified |
| Cardinality | NotSpecified |
| RelationshipType | NotSpecified |
| RelatedTableRole | |
| Role | |

Figure 2—Sample surrogate foreign key Relation property sheet

Right-click the newly created and configured **Relation**, and then click **New > New Foreign Key**. You should see something similar to Figure 3.

```

PrimaryKeyTable(sys)
_ForeignKeyTable_PrimaryKeyTable == _PrimaryKeyTable.ReclId

```

Figure 3 Sample surrogate foreign key Relation

Note that the creation of a surrogate foreign key relationship automatically adds a field to the foreign key table for you.

What problems do surrogate keys present to the UI?

Displaying surrogate keys

Unsuitability of displaying surrogate keys in the UI

Surrogate keys—and, in turn, surrogate foreign keys—are represented by RecIds (64-bit integers) in Microsoft Dynamics AX 2012. In standard application scenarios, RecIds should not be displayed to the user, because a RecId by itself does not provide useful or easily understandable information. An alternative set of fields must be displayed in place of the surrogate foreign key.

Explicit and implicit surrogate key UI replacement

The general surrogate foreign replacement problem can be broken into two more specific scenarios:

- Explicit replacement (by the developer at design time)
- Implicit replacement (by the kernel at run time)

Explicit replacement occurs during Form design, when the developer explicitly models a surrogate key—or surrogate foreign key—to be replaced with a specified set of fields in the UI. Note that this doesn't preclude the kernel from assisting with the design-time experience. For example, dragging a surrogate foreign key field onto a Form design will result in the appropriate default control being used.

Implicit replacement occurs when the kernel is internally generating a non-modeled form (such as a lookup), or when it is rendering pieces of the UI that are modeled by the developer to use surrogate keys, but expose no metadata for picking replacement metadata (such as a Form caption or Field Groups). In these cases, the kernel considers the context of the UI element and automatically replaces it with the "best" set of replacement fields.

Editing surrogate foreign keys

Resolving a user-entered value into a related record

The user will not only see the surrogate foreign key replaced by a set of replacement fields, but will also enter the surrogate key through the replacement fields. From the user's perspective, he or she is just entering values as though they exist on the foreign key table—however, what's actually happening is that the user is indirectly entering a surrogate foreign key by selecting a record from the related table.

The kernel handles *resolution* of the user's entered values into the appropriate related record and then, in turn, sets the surrogate foreign key that the user is editing.

The uniqueness problem

Most of the time, a surrogate key will be replaced by the primary key table's natural key (which must be unique; see the definition in terms table). However, there are currently areas of the application where a natural key that could be used to replace the surrogate foreign key either does not exist, or would be inappropriate to display to the user (for example, too many fields). In this scenario, if the user enters values that are not unique in the context of the replacement fields, it will not be possible to automatically determine which record the user was referring to.

This issue is addressed by providing both the X++ developer and the end user a means of disambiguating the entered data. In the case of the X++ developer, there is an X++ override (more about this later) that lets the developer consider Form and control state to determine which record the user was referring to. If the X++ developer doesn't intervene, *and* if the user's entered

value is ambiguous, the user is presented with a lookup that will let that user choose which record he or she was referring to.

MorphX objects and X++ classes

This section lists the MorphX objects and X++ classes that you can use to work with surrogate keys.

Tables

AutoIdentification field group

X++ name: Not applicable

Overview: **AutoIdentification** is a system-generated field group that provides the default set of surrogate foreign key replacement fields.

The field group has two modes of population:

- The first—and default—mode automatically populates the field group from the **ReplacementKey** property defined on the containing table's base property sheet.
- The second mode, which should be used only in advanced scenarios, turns off automatic population and lets the developer manually specify the fields contained by **AutoIdentification**.

To toggle the population mode, open the property sheet of the appropriate **AutoIdentification** field group from the AOT, and change its **AutoPopulate** property to **Yes** or **No**, as desired.

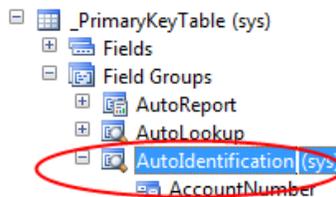


Figure 4—AutoIdentification AOT node

| Name | AutoIdentification |
|--------------|--------------------|
| Label | Hello |
| AutoPopulate | Yes |

Figure 5—Sample AutoIdentification property sheet

Quick facts: AutoIdentification:

- Provides the default set of **human-understandable** fields that are used to replace surrogate foreign keys to the field group's containing table. This field group is used automatically in implicit replacement scenarios.
- Is automatically populated by the containing table's **ReplacementKey** property.
- Can be manually populated in *advanced* scenarios by switching the **AutoPopulate** property to **No**.

Important: Switch **AutoPopulate** to **No** only in advanced scenarios where an appropriate natural key does not exist.

Forms

Reference Control

X++ name: **FormReferenceControl**

Overview: The abstract base class of a new set of controls used to display surrogate foreign keys in the rich client's UI. The Reference Control uses two bindings. The first binding—known as the “physical” binding—declares which surrogate key the control is bound to; and the second binding—known as the “logical” binding—declares which fields from the primary key table will replace the surrogate foreign key in the UI. Figure 6 shows an example property sheet with the new **ReferenceField** and **ReplacementFieldGroup** properties.

| | |
|-----------------------|-----------------------------------------------------------------------------------------------------------|
| DragDrop | None |
| DataSource | _ForeignKeyTable |
| ReferenceField |  _PrimaryKeyTable |
| ReplacementFieldGroup | AutoIdentification |
| Value | 0 |

Figure 6—Example of a Reference Control property sheet. All Reference Controls have the DataSource, ReferenceField, and ReplacementFieldGroup properties.

One key thing to note is that the ReplacementFieldGroup is selected from the *primary* key table, not from the foreign key table that it is physically bound to.

You'll find the **FormReferenceControl** class itself used as a parameter in new X++ methods that are designed specifically for Reference Controls.

Quick facts: Reference Controls:

- Let the developer display human-understandable data in place of surrogate foreign keys in the rich client.
- Require two bindings. The first (“physical”) binding defines the surrogate foreign key field that the control is bound to; and the second (“logical”) binding defines the set of fields that will be displayed instead of the surrogate foreign key.
- Have their physical binding defined through the **ReferenceField** property.
- Have their logical binding defined through the **ReplacementFieldGroup** property.
- Require the **AutoIdentification** field group to be populated before they can be automatically used on a data source field Drag + Drop action or within a field group used on a Form design.

Reference Group control

X++ name: **FormReferenceGroupControl** and **FormBuildReferenceGroupControl** for run-time and design-time instances of the control, respectively

Overview: The Reference Group control is the generic control that can be used to edit any kind of surrogate foreign key. This will be the most frequently used type of Reference Control. It is essentially a specialized version of the group control—hence the name Reference Group.

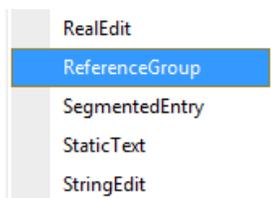


Figure 7—The Reference Group and Segmented Entry controls on the New Control context menu

Quick facts: Reference Group:

- Lets the user enter data either manually by typing, or through lookup forms that display data from the primary key table.
- Presents the user with a lookup when manually entered data is ambiguous.
- Is a hybrid container and edit control.

Segmented Entry control

X++ name: **FormSegmentedEntryControl** and **FormBuildSegmentedEntryControl** for run-time and design-time instances of the control, respectively

Overview: The Segmented Entry control is the specialized control designed specifically for use with account number entry.

Quick fact: Segmented Entry is a specialized control that provides editing of account numbers in the Microsoft Dynamics AX 2012 client.

Reference Data Source

X++ name: **FormDataSource** and **FormBuildDataSource** base classes for run-time and design-time instances, respectively

Overview: A new type of read-only data source used to supply the data for replacing the surrogate foreign key in the UI—the “logical” binding.

The most important thing to remember about Reference Data Sources is that you do *not* have to manually model them in most cases—the kernel will handle that for you automatically.

Quick facts: Reference Data Sources:

- Are read-only, and therefore cannot be inserted or updated.
- Contains no Methods node.
- Are nested within standard data source nodes in the AOT.
- Are forced to be outer-joined to the containing data source node, as illustrated in Figure 8.
- Are bound to a table through the selection of a relationship that is defined on the foreign key table, rather than through selection of the table directly.
- Are automatically added by the kernel if they are not manually modeled.

Important: Reference Data Sources should be manually modeled only in advanced scenarios.

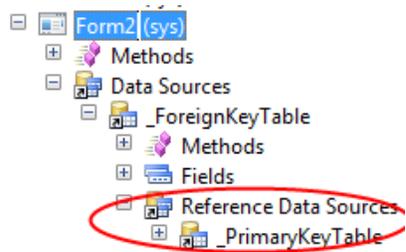


Figure 8—Reference Data Sources AOT node

| Name | PrimaryKeyTable |
|-------------------------|------------------------|
| Table | _PrimaryKeyTable |
| AllowCheck | No |
| AutoSearch | Yes |
| AutoNotify | Yes |
| AutoQuery | Yes |
| CrossCompanyAutoQuery | No |
| OnlyFetchActive | No |
| JoinSource | _ForeignKeyTable |
| JoinRelation | PrimaryKeyTable |
| LinkType | OuterJoin |
| DelayActive | Yes |
| ValidTimeStateAutoQuery | AsOfDate |
| ValidTimeStateUpdate | Revision |

Figure 9—Sample Reference Data Source property sheet

Reference Fields

X++ name: FormReferenceObject

Overview: A new type of data source field representing a surrogate foreign key. Reference Fields supply the “physical” binding of a Reference Control and provide additional X++ methods that can be overridden.

Interesting overrides: The **resolveReference** method provides a means of supplying custom logic for resolving a user’s entered value into a related record. The **lookupReference** method lets a developer provide a custom lookup form for Reference Controls bound to the Reference Field.

Quick facts: Reference Fields:

- Belong to Form data sources.
- Are generated when a surrogate foreign key is represented.
- Provide the physical binding to Reference Controls.
- Provide an additional set of overrides beyond those offered by standard Data Fields.
- Let the X++ developer provide custom user-entered value resolution through overrides of its **resolveReference** method.
- Use the **lookupReference** method (instead of the classical **lookup** method) when presenting a lookup form through Reference Controls.

Classes

SysReferenceTableLookup

X++ name: SysReferenceTableLookup

Overview: A sibling of **SysTableLookup** that is used to generate Grid-style custom lookups for Reference Controls.

Interesting overrides: The **performFormLookup** method, unlike its **SysTableLookup** sibling, returns the record selected in the lookup form.

Quick fact: **SysReferenceTableLookup** is a helper class used to generate Grid-style custom lookups for Reference Controls.

FormRun

X++ name: FormRun

Overview: The **FormRun** class represents a run-time instance of a Form.

Interesting methods: The **selectRecordMode** method places the Form into “record selection mode,” which is similar to the classic select mode, but with the added behavior that an entire record is selected instead of a value from an individual field. The **closeSelectRecord** method is called when a record has been selected on the Form, and the **selectRecordModeSelectedRecord** method lets the developer retrieve the record that was selected.

Quick facts: FormRun:

- Is a run-time representation of a Form.
- Provides the **selectRecordMode** method for placing the Form into select record mode (that is, for making it a lookup Form).
- Provides the **closeSelectRecord** method, which is called when a Form is closing during select record mode.
- Provides the **selectRecordModeSelectedRecord** method for retrieving the record that was selected in the lookup Form.

Examples

Example 1: Basic Reference Group end-to-end scenario

Task: Given the data model shown in Figure 10, create a Form that presents **_Employee** to the user without displaying the raw RecId data behind the **Job** field. Figures 11 and 12 show some sample data for the tables that are entered into Microsoft Dynamics AX.

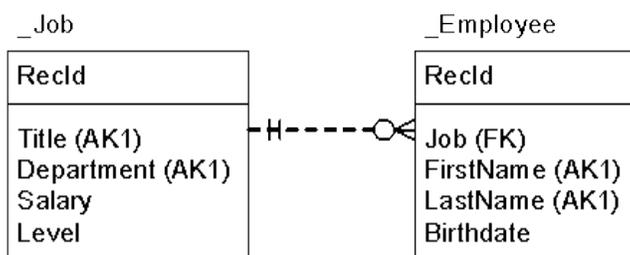


Figure 10—The arbitrary data model used for this example. **_Job represents a list of jobs within a company, whereas **_Employee** represents the employees within a company.**

| | FirstName | LastName | Birthdate | Job | dataAreaId | RecId |
|------------------|-----------|-----------|------------|------------|------------|------------|
| _Employee | John | Doe | 1/1/1980 | 5637144578 | dmo | 5637144580 |
| | Ken | Silverman | 1/1/1976 | 5637144577 | dmo | 5637144579 |
| | Carl | Gauss | 4/30/19... | 5637144579 | dmo | 5637144578 |
| | Bruce | Banner | 5/1/1962 | 5637144581 | dmo | 5637144577 |
| | Kal | El | 6/30/19... | 5637144576 | dmo | 5637144576 |

Figure 11—Arbitrary data entered into the `_Employee` table

| | Title | Salary | Level | Department | dataAreaId | RecId |
|-------------|------------|------------|-------|------------|------------|------------|
| _Job | CEO | 600,000.00 | 90 | EOD | dmo | 5637144576 |
| | Engineer 1 | 60,000.00 | 50 | Software | dmo | 5637144577 |
| | Engineer 1 | 60,000.00 | 50 | EE | dmo | 5637144578 |
| | Engineer 2 | 70,000.00 | 55 | EE | dmo | 5637144579 |
| | Physicist | 70,000.00 | 55 | EE | dmo | 5637144581 |
| | Lead 1 | 80,000.00 | 60 | EOD | dmo | 5637144580 |

Figure 12 Arbitrary data entered into the `_Job` table

Pre-step 1: Create tables corresponding to the data model shown in Figure 10.

Code that you can copy to create the XPO file is provided in [Appendix A: SharedProject ReferenceGroupSample 1](#).

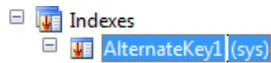
Pre-step 2: Add some data to the new tables by using the table browser, or run the `CreateDataForReferenceGroupSample` job included in the example `ShareProject_ReferenceGroupSample`.

Step 1: In the data model, the `_Job` table is the primary key table, whereas `_Employee` is the foreign key table. The `Job` field on the `_Employee` table forms the surrogate foreign key between the `_Employee` and `_Job` tables. The **AutoIdentification** field group on the `_Job` table will provide the default list of fields used to replace the `Job` field on the `_Employee` table in the UI. Therefore, the first step is to verify that the **AutoIdentification** field group is set up correctly on the `_Job` table.

Open the `_Job` table in the AOT. Verify that the **Field Groups > AutoIdentification** node is populated with the fields that form the alternate key of `_Job`, as shown in Figure 10. (Hint: These fields are **Title** and **Department**.) If it is not, note that the **AutoPopulate** property on the **AutoIdentification** field group should be set to **Yes** as a best practice—meaning that a natural key needs to be set on the table. If a natural key does not exist, create a unique index for the **Title** and **Department** fields. Set the **AlternateKey** property to **Yes** on the index.

Open the property sheet of the `_Job` table, and set the **ReplacementKey** (natural key) property to the name of the new index (**AlternateKey1** in the example). Verify that **AutoIdentification** contains the **Title** and **Department** fields.

Note: AutoIdentification is set to **AutoPopulate = No** in advanced scenarios where an appropriate natural key does not exist.



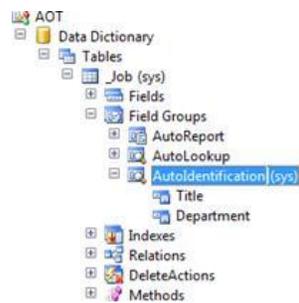
| Name | AlternateKey1 |
|------------------------|---------------|
| AllowDuplicates | No |
| Enabled | Yes |
| ConfigurationKey | |
| AlternateKey | Yes |
| ValidTimeStateKey | No |
| ValidTimeStateMode | |

Figure 13—Alternate key defined on _Job



| | |
|-----------------------|----------------------|
| TableGroup | Miscellaneous |
| PrimaryIndex | SurrogateKey |
| ClusterIndex | SurrogateKey |
| ReplacementKey | AlternateKey1 |
| IsLookup | No |

Figure 14—Natural key defined on _Job



| Name | AutoIdentification |
|--------------|--------------------|
| Label | |
| AutoPopulate | Yes |

Figure 15—Expected contents of AutoIdentification

Step 2: Create a new form that will display the _Employee table with no visible surrogate foreign key fields.

Create a new form named **_Employee**. Add a new data source, and set its **Table** property to **_Employee**. Add a grid to the design. Within the grid, add the **FirstName**, **LastName**, and **Birthdate** fields. Now add the Reference Group control either by dragging the **Job** field onto the design, or by right-clicking the grid and clicking **New Control > ReferenceGroup**. Set the **DataSource** and **ReferenceField** properties to **_Employee** and **Job**, respectively. Open the form, and you should see something similar to Figure 17. Note that the Reference Data Source is added by the kernel automatically (also shown in Figure 17.1).

| Name | ReferenceGroup |
|-----------------------|--------------------|
| AutoDeclaration | No |
| Visible | Yes |
| AllowEdit | Yes |
| Securable | No |
| Enabled | Yes |
| Skip | No |
| HTMLHelpFile | |
| HTMLHelpTopic | |
| ConfigurationKey | |
| SecurityKey | |
| DataSource | _Employee |
| ReferenceField | Job |
| ReplacementFieldGroup | Autoidentification |
| Value | 0 |
| AutoDataGroup | Yes |
| AllowUserSetup | Yes |
| NeededAccessLevel | View |
| ViewEditMode | Auto |

Figure 16—Configuration of the _Employee form

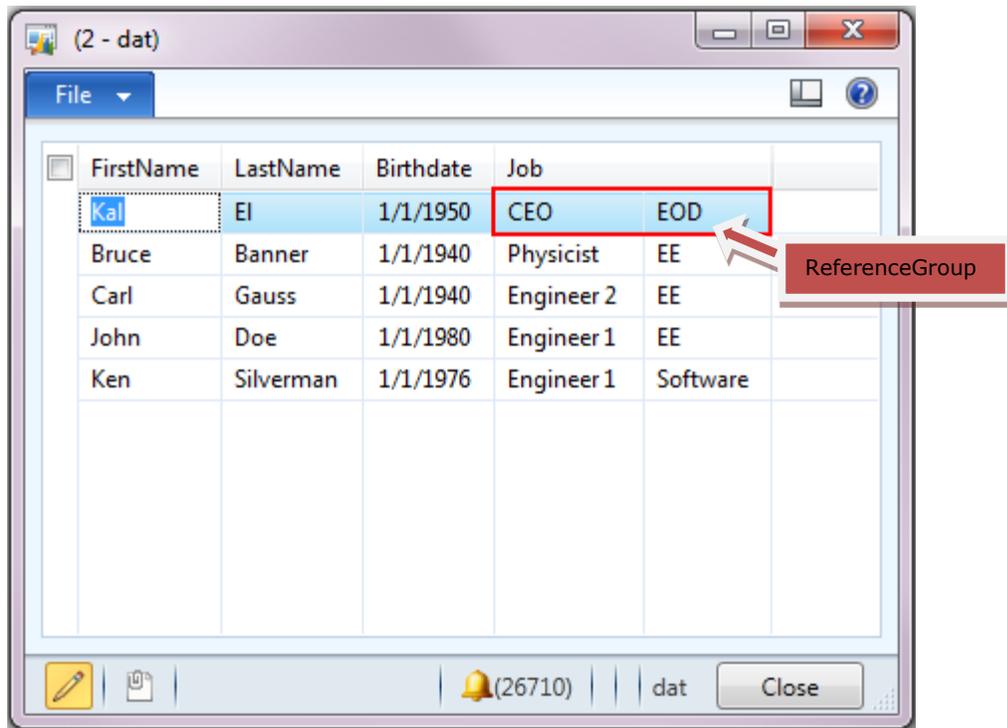


Figure 17—Visualization of the Reference Group

Key points

- **AutoIdentification** should be modeled on the primary key table before you add the surrogate foreign key field to the foreign key table.
- Reference Group has **two** data bindings: **ReferenceField** and **ReplacementFieldGroup**.
- Reference Data Sources are automatically added by the kernel during initialization of the form. Reference Data Sources need to be manually modeled only in advanced scenarios.

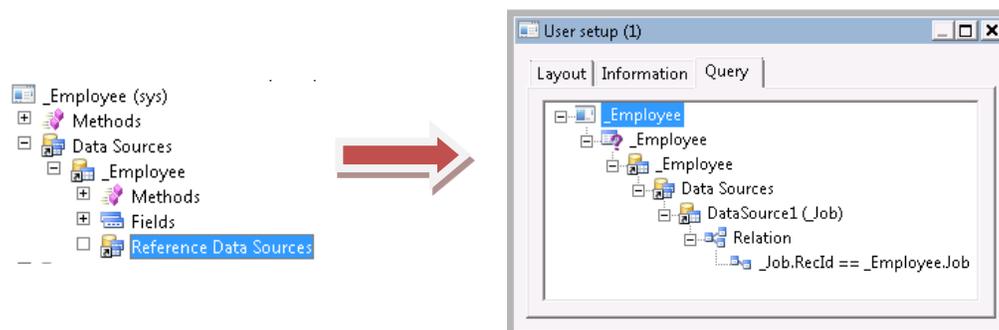


Figure 17.1—Reference Data Sources are added automatically by the kernel.

Note: A replacement field group other than **AutoIdentification** can be chosen. However, this should be reserved for advanced scenarios where multiple alternate keys exist.

Example 2: Ambiguous data entry

Task: Given the data model from Example 1, create a Form named **_Employee** that presents **_Employee** to the user without displaying the raw RecId data behind the **Job** field. Additionally, support the requirement that the user sees the **_Job** table's **Title** field in place of the Job surrogate foreign key only when viewing and editing data in a grid.

Example code for an XPO file is provided in [Appendix B: SharedProject ReferenceGroupSample 2](#).

Pre-step 1: Perform the pre-steps from Example 1 if they have not been done already. This will add the requisite data and metadata.

Step 1: The first thing to note from this example is that the **Title** field alone does **not** form a natural key. Therefore, the user's entered values are not guaranteed to uniquely identify a record (see **Title = Engineer 1** in Figure 12 for an example)—meaning that the kernel's disambiguation functionality might be used during data entry.

Because Reference Controls can bind to any field group, add a new field group named **AutoIdentification_2** to the **_Job** table (a better name should be chosen in real development scenarios); add the **Title** field to the new field group.

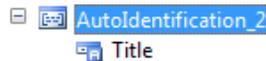


Figure 18—New **AutoIdentification_2** field group

Step 2: Open the **_Employee** form in the AOT, and browse to the Reference Group control named **ReferenceGroup**, under **Designs > Design > Grid**. Change the **ReplacementFieldGroup** property to **AutoIdentification_2**, as shown in Figure 19.

| | |
|-----------------------|----------------------|
| DataSource | _Employee |
| ReferenceField | Job |
| ReplacementFieldGroup | AutoIdentification_2 |

Figure 19—Reference Group configured to use an alternate field group

Step 3: Open the **_Employee** form, and verify that it looks similar to Figure 20. Try creating a new **_Employee** record, and see what happens when you enter **Engineer 1** as the job. Note that **this type of behavior should be leveraged only in advanced scenarios** where requirements specifically call for it.

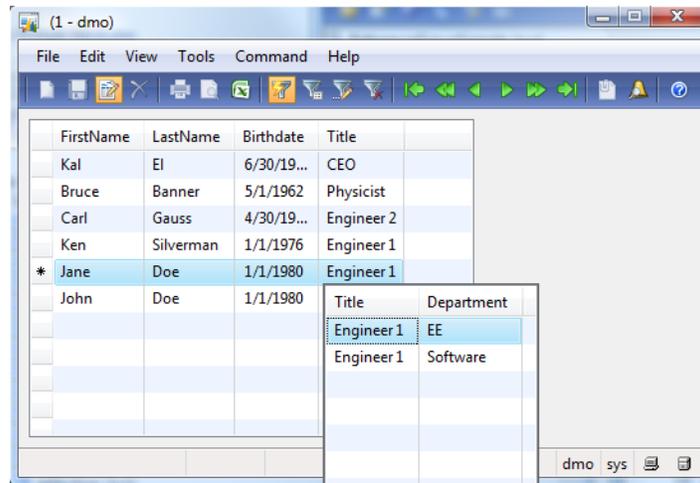


Figure 20—The modified **_Employee form displaying a lookup to let the user disambiguate the entry of “Engineer 1”**

Key points

- A Reference Control can have its **ReplacementFieldGroup** (logical) binding made between field groups other than **AutoIdentification**.
- Reference Group will display a lookup to let the user disambiguate an ambiguous entry.

Note: A Reference Control should have its **ReplacementFieldGroup** binding made to a field group that does not correspond exactly to an alternate key only in advanced scenarios.

Example 3: Overriding the default reference resolving behavior

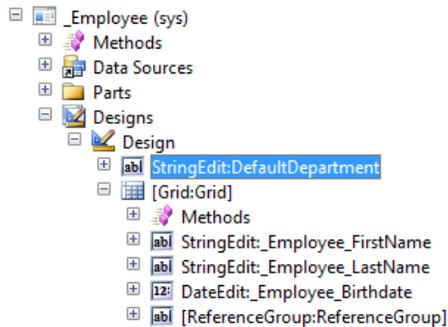
Task: Given the data model from Example 1 and the additional requirements from Example 2 (see the task for Example 2), create a Form named **_Employee** that presents **_Employee** to the user without displaying the raw **RecId** data behind the **Job** field. Additionally, prevent the disambiguation lookup form (shown in Figure 20) from appearing by overriding the **resolveReference** method on the **Job** field and leveraging additional context from the form.

Example code for an XPO file is provided in [Appendix C: SharedProject ReferenceGroupSample 3](#).

Pre-step 1: Perform all steps from Example 2. This will add the requisite data and metadata.

Step 1: Unlike the last example, the requirements now explicitly dictate that the **_Employee** form cannot leverage the default disambiguation behavior. The main problem here is that some additional context or data is required to determine which records the user is meaning to enter. In this case, the entered data will be ambiguous over the **Department** field, so a string control, a combo box, or some other type of control should be provided to let the user provide a default department. For the sake of brevity, a String Edit control will be used in this example.

Open the **_Employee** form in the AOT, and add a new String Edit control named **DefaultDepartment** to the design. Set the **AutoDeclaration** property to **Yes** and the **Label** property to **Default Department**, as shown in Figure 22.



| Name | DefaultDepartment |
|------------------------|---------------------------|
| AutoDeclaration | Yes |
| Visible | Yes |
| LabelForegroundColor | Window text |
| ShowLabel | Yes |
| Label | Default Department |

Figure 21—The new DefaultDepartment String Edit control

Figure 22—The property sheet of the DefaultDepartment control

Step 2: Leaving the **_Employee** form open in the AOT, browse to the **Data Sources > _Employee > Fields > Job > Methods** node, and override the **resolveReference** method. Copy and paste the following code into the overridden method, and then recompile the form.

```
public Common resolveReference(FormReferenceControl _formReferenceControl)
{
    _Job resolvedRecord;
    str 100 title = '', defaultDepartmentStr = '';

    // Do not call super as we're providing our own disambiguation logic.
    // resolvedRecord = super(_formReferenceControl);

    title = _formReferenceControl.filterValue(AbsoluteFieldBinding::construct(fieldStr(_Job,
    Title), tableStr(_Job))).value();

    defaultDepartmentStr = DefaultDepartment.valueStr();

    select firstly
        resolvedRecord
    where
        resolvedRecord.Title == title &&
        resolvedRecord.Department == defaultDepartmentStr;

    return resolvedRecord;
}
```

Listing 1 A – custom implementation of resolveReference that resolve’s a user’s typed-in data to a record from the _Job (primary key) table.

Again, note that extra care should be taken when you implement the **resolveReference** method. It should be overridden only in advanced scenarios.

Step 3: Open the **_Employee** form, and enter **Software** into the DefaultDepartment string control. Now create a new record, type **Engineer 1** into the **Title** field, and then press Tab to move away from

the field. Note how the custom **resolveReference** method automatically resolved the ambiguous entry by using the department specified in the DefaultDepartment string control.

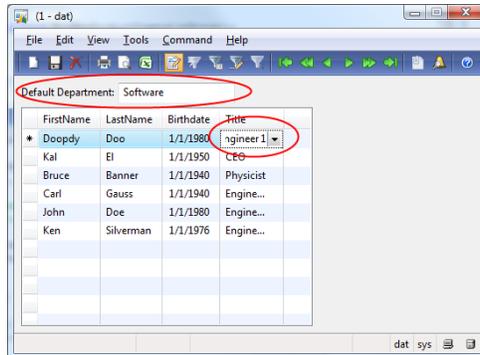


Figure 23—The `_Employee` form using custom disambiguation logic from Listing 1

Key points

- The new **resolveReference** override on the Reference Field lets the X++ developer provide a custom means of retrieving a record from the primary key table.
- Override the **resolveReference** method **only** in advanced scenarios, as dictated by application requirements.

Note: When the super call is removed from the **resolveReference** method, the default disambiguation behavior of presenting a lookup form will no longer occur.

Example 4: Adding a custom lookup

Task: Given the data model from Example 1, and the additional requirements from Examples 2 and 3, create a Form named **_Employee** that presents `_Employee` to the user without displaying the raw `RecId` data behind the **Job** field. Additionally, provide a custom lookup form that filters on the `DefaultDepartment` control's current value.

Example code for an XPO file is provided in [Appendix D: SharedProject_ReferenceGroupSample_4](#).

Pre-step 1: Perform all steps from Example 3. This will add the requisite data and metadata.

Step 1: The new **lookupReference** override on the **Reference** field provides a means of presenting custom lookups for Reference Controls.

Open the **_Employee** form in the AOT, browse to the **Data Sources > _Employee > Fields > Job > Methods** node, and override the **lookupReference** method. Copy and paste the following code into the overridden method, and then recompile the form.

```
public Common lookupReference(FormReferenceControl _formReferenceControl)
{
    _Job selectedRecord;
    SysReferenceTableLookup sysTableLookup =
    SysReferenceTableLookup::newParameters(tablenum(_Job),
    _formReferenceControl, true);
    Query lookupQuery;
    str 100 defaultDepartmentStr = '';

    // Do not call super as we're providing our own custom lookup logic.
    // selectedRecord = super(_formReferenceControl);
}
```

```

defaultDepartmentStr = DefaultDepartment.valueStr();

// Display the Title and Department fields in the lookup form.
sysTableLookup.addLookupfield(fieldnum(_job, Title));
sysTableLookup.addLookupfield(fieldnum(_job, Department));

// Create a custom Query that filters on the Department field.
lookupQuery = new Query();
lookupQuery.addDataSource(tablenum(_Job)).addRange(fieldnum(_Job,
Department)).value(defaultDepartmentStr);
sysTableLookup.parmQuery(lookupQuery);

// Return the record selected by the user.
selectedRecord = sysTableLookup.performFormLookup();

return selectedRecord;
}

```

Listing 2—A custom implementation of lookupReference that presents a custom lookup form filtered by the default department specified on the main form.

The custom lookup code leverages the new **SysReferenceTableLookup** class, which provides support for creating “standard”-looking custom lookups that use a single Grid.

Step 2: Open the **_Employee** form, and use a lookup to pick a job. Note that if no default department is specified, the lookup will present all records to the user. In an actual application scenario, you will probably want to update the **resolveReference** method to expose the same set of records to the user through typing. Figures 24 and 25 show the lookup that is presented to the user when no default department is specified and when EE is the default department, respectively.

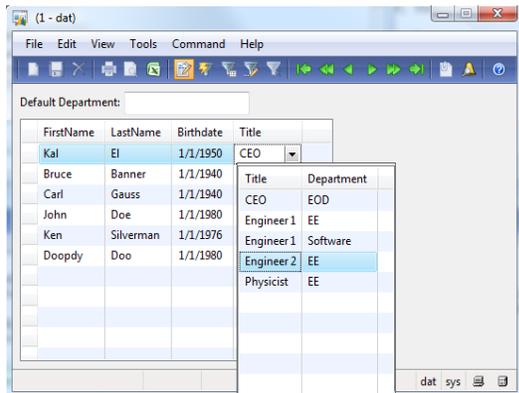


Figure 24—The custom lookup with no default department specified

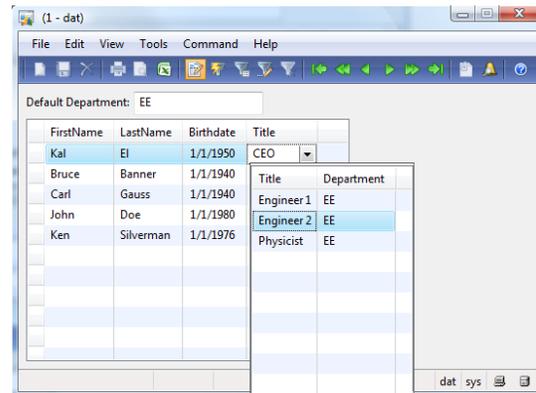


Figure 25—The custom lookup form with “EE” specified as the default department

Key points

- The **SysReferenceTableLookup** X++ class provides basic support for creating custom lookups that display data in a grid.
- The **lookupReference** method should return the record that the user selected within the displayed lookup form.

- In most cases, the **lookupReference** and **resolveReference** overrides should provide the same set of “selectable” records to the user.

Note: There are additional methods on **FormRun** that provide support for even more advanced scenarios where **SysReferenceTableLookup** is not sufficient.

Example 5: Using a Reference Data Source to display additional read-only data from a related record

Task: Given the data model and requirements from Example 1, create a Form named **_Employee** that presents **_Employee** to the user without displaying the raw RecId data behind the **Job** field. Additionally, add a tab page named **Job** that displays read-only versions of the **Title**, **Department**, **Level**, and **Salary** fields.

Example code for an XPO file is provided in [Appendix E: SharedProject ReferenceGroupSample 5](#).

Pre-step 1: Perform all steps from Example 1. This will add the requisite data and metadata.

Step 1: Add a Tab control with two tab pages to the root of the design. Name the first tab page **Overview** and the second tab page **Job**. Move the existing Ggrid control inside the **Overview** tab page.

Step 2: Browse to the **_Employee** data source, expand its **Reference Data Sources** node, and add a new Reference Data Source named **_Job**. Open property sheet for the new data source, and note that its **Table** property is read-only. Set the **JoinRelation** property to **_Job**; note that the **Table** property is now set to **_Job**.

Step 3: Open the **Fields** node of the **_Job** data source, and verify that the **Title**, **Department**, **Level**, and **Salary** fields are present. To the **Job** tab page created in step 1, add controls that map to those fields.

Step 4: Open the **_Employee** form, and verify that the controls on the **Job** tab page are read-only. Note how the fields are automatically updated when a different **_Job** record is selected.

Key points

- Binding directly to fields on a manually modeled Reference Data Source lets you provide read-only information about a related record, beyond what is displayed in a Reference Control.
- Disclosing additional information about a related record is one of the few scenarios where a developer should manually model Reference Data Sources.

Appendix A: SharedProject_ReferenceGroupSample_1

Exportfile for AOT version 1.0 or later

Formatversion: 1

***Element: JOB

; Microsoft Dynamics AX Job: CreateDataForReferenceGroupSample unloaded

; -----

JOBVERSION 1

```
SOURCE #CreateDataForReferenceGroupSample
#static void CreateDataForReferenceGroupSample(Args _args)
#{
#   _Employee _employee;
#   _Job _job;
#
#   ttsbegin;
#
#   delete_from _employee;
#   delete_from _job;
#
#   // Insert the "CEO"/"EDO" job and all assigned employees
#   _job.Title = 'CEO';
#   _job.Salary = 600000;
#   _job.Level = 90;
#   _job.Department = 'EOD';
#   _job.insert();
#
#   _employee.FirstName = 'Kal';
#   _employee.LastName = 'El';
#   _employee.Birthdate = 1\1\1950;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Physicist"/"EE" job and all assigned employees
#   _job.Title = 'Physicist';
#   _job.Salary = 70000;
#   _job.Level = 55;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'Bruce';
#   _employee.LastName = 'Banner';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 2"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 2';
#   _job.Salary = 70000;
#   _job.Level = 55;
```

```

#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'Carl';
#   _employee.LastName = 'Gauss';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 1"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 1';
#   _job.Salary = 60000;
#   _job.Level = 50;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'John';
#   _employee.LastName = 'Doe';
#   _employee.Birthdate = 1\1\1980;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 1"/"Software" job and all assigned employees
#   _job.Title = 'Engineer 1';
#   _job.Salary = 60000;
#   _job.Level = 50;
#   _job.Department = 'Software';
#   _job.insert();
#
#   _employee.FirstName = 'Ken';
#   _employee.LastName = 'Silverman';
#   _employee.Birthdate = 1\1\1976;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   ttscommit;
#}

```

ENDSOURCE

***Element: FRM

; Microsoft Dynamics AX Forms unloaded

; -----

FRMVERSION 6

FORM #_Employee

PROPERTIES

 Name #_Employee

ENDPROPERTIES

METHODS

 Version: 3

```

SOURCE #classDeclaration
    #public class FormRun extends ObjectRun
    #{
    #}
ENDSOURCE
ENDMETHODS
OBJECTBANK
PROPERTIES
ENDPROPERTIES

DATASOURCE
    OBJECTPOOL
        PROPERTIES
            Name                #_Employee
            Table                #_Employee
        ENDPROPERTIES

        FIELDLIST
        ENDFIELDLIST
    ENDOBJECTPOOL
    METHODS
        Version: 3
    ENDMETHODS
    ENDDATASOURCE
ENDOBJECTBANK

REFERENCEDATASOURCES
ENDREFERENCEDATASOURCES

JOINS
ENDJOINS

PARTREFERENCES
ENDPARTREFERENCES

DESIGN
PROPERTIES
ENDPROPERTIES

CONTAINER
    CONTROL GRID
        PROPERTIES
            Name                #Grid
            Width                #Column width
            DataSource            #_Employee
        ENDPROPERTIES

        METHODS
            Version: 3
        ENDMETHODS
    CONTAINER
        CONTROL STRINGEDIT

```

```

PROPERTIES
    Name                #_Employee_FirstName
    DataSource           #_Employee
    DataField            #FirstName
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL STRINGEDIT
PROPERTIES
    Name                #_Employee_LastName
    DataSource           #_Employee
    DataField            #LastName
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL DATEEDIT
PROPERTIES
    Name                #_Employee_Birthdate
    DataSource           #_Employee
    DataField            #Birthdate
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL REFERENCEGROUP
PROPERTIES
    Name                #ReferenceGroup
    DataSource           #_Employee
    ReferenceField       #Job
    BackgroundColor     #Window background
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
CONTAINER
ENDCONTAINER

ENDCONTROL

ENDCONTAINER

```

```

ENDCONTROL

ENDCONTAINER

ENDDSIGN

ENDFORM

***Element: DBT

; Microsoft Dynamics AX Table : _Employee unloaded
; -----
TABLEVERSION 1

TABLE #_Employee
    EnforceFKRelation 1
    PROPERTIES
        Name                #_Employee
        CreateRecIdIndex    #Yes
        PrimaryIndex        #SurrogateKey
        ClusterIndex        #SurrogateKey
    ENDPROPERTIES

    FIELDS
        FIELD #FirstName
            STRING
            PROPERTIES
                Name                #FirstName
                Table                #_Employee
            ENDPROPERTIES

        FIELD #LastName
            STRING
            PROPERTIES
                Name                #LastName
                Table                #_Employee
            ENDPROPERTIES

        FIELD #Birthdate
            DATE
            PROPERTIES
                Name                #Birthdate
                Table                #_Employee
            ENDPROPERTIES

        FIELD #Job
            INT64
            PROPERTIES
                Name                #Job
                Table                #_Employee
    
```

```

        ENDPROPERTIES

    ENDFIELDS
    GROUPS
    ENDGROUPS

    INDICES
    ENDINDICES
    REFERENCES
    REFERENCE #_Job
        PROPERTIES
            Name                #_Job
            Table                #_Job
        ENDPROPERTIES

    FIELDREFERENCES
    REFERENCE TYPE PKFK
    PROPERTIES
        Field                  #Job
        RelatedField           #RecId
    ENDPROPERTIES

    ENDFIELDREFERENCES
    ENDREFERENCE
    ENDREFERENCES

    DELETEACTIONS
    ENDELETEACTIONS

    METHODS
    Version: 3
    ENDMETHODS
    ENDTABLE

```

***Element: DBT

; Microsoft Dynamics AX Table : _Job unloaded

; -----

```

TABLEVERSION 1

TABLE #_Job
    EnforceFKRelation 1
    PROPERTIES
        Name                #_Job
        CreateRecIdIndex    #Yes
        PrimaryIndex        #SurrogateKey
        ClusterIndex        #SurrogateKey
        ReplacementKey      #AlternateKey1
    ENDPROPERTIES

    FIELDS

```

```

FIELD #Title
STRING
PROPERTIES
    Name          #Title
    Table         #_Job
ENDPROPERTIES

FIELD #Salary
REAL
PROPERTIES
    Name          #Salary
    Table         #_Job
ENDPROPERTIES

FIELD #Level
INT
PROPERTIES
    Name          #Level
    Table         #_Job
ENDPROPERTIES

FIELD #Department
STRING
PROPERTIES
    Name          #Department
    Table         #_Job
ENDPROPERTIES

ENDFIELDS
GROUPS
GROUP #AutoIdentification
PROPERTIES
    Name          #AutoIdentification
ENDPROPERTIES

GROUPFIELDS
    #Title
    #Department
ENDGROUPFIELDS
ENDGROUP
ENDGROUPS

INDICES
#AlternateKey1
PROPERTIES
    Name          #AlternateKey1
    AllowDuplicates #No
    AlternateKey   #Yes
ENDPROPERTIES

INDEXFIELDS
    #Title

```

```

        #Department
    ENDINDEXFIELDS

ENDINDICES
REFERENCES
ENDREFERENCES

DELETEACTIONS
ENDDELETEACTIONS

METHODS
    Version: 3
ENDMETHODS
ENDTABLE

***Element: PRN

; Microsoft Dynamics AX Project : ReferenceGroupSample unloaded
; -----
PROJECTVERSION 2

PROJECT #ReferenceGroupSample
    SHARED
PROPERTIES
    Name                #ReferenceGroupSample
ENDPROPERTIES

PROJECTCLASS ProjectNode
BEGINNODE
    FILETYPE 0
    UTILTYPE 5
    UTILOBJECTID 0
    NODETYPE 215
    NAME #CreateDataForReferenceGroupSample
ENDNODE
BEGINNODE
    FILETYPE 0
    UTILTYPE 11
    UTILOBJECTID 0
    NODETYPE 201
    NAME #_Employee
ENDNODE
BEGINNODE
    FILETYPE 0
    UTILTYPE 44
    UTILOBJECTID 6187
    NODETYPE 204
    NAME #_Employee
ENDNODE
BEGINNODE
    FILETYPE 0

```

```
UTILTYPE 44
UTILOBJECTID 6188
NODETYPE 204
NAME #_Job
ENDNODE
ENDPROJECT
```

```
***Element: END
```

Appendix B: SharedProject_ReferenceGroupSample_2

Exportfile for AOT version 1.0 or later

Formatversion: 1

**Element: JOB

; Microsoft Dynamics AX Job: CreateDataForReferenceGroupSample unloaded

; -----JOBVERSION 1

```
SOURCE #CreateDataForReferenceGroupSample
#static void CreateDataForReferenceGroupSample(Args _args)
#{
#   _Employee _employee;
#   _Job _job;
#
#   ttsbegin;
#
#   delete_from _employee;
#   delete_from _job;
#
#   // Insert the "CEO"/"EDO" job and all assigned employees
#   _job.Title = 'CEO';
#   _job.Salary = 600000;
#   _job.Level = 90;
#   _job.Department = 'EOD';
#   _job.insert();
#
#   _employee.FirstName = 'Kal';
#   _employee.LastName = 'El';
#   _employee.Birthdate = 1\1\1950;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Physicist"/"EE" job and all assigned employees
#   _job.Title = 'Physicist';
#   _job.Salary = 70000;
#   _job.Level = 55;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'Bruce';
#   _employee.LastName = 'Banner';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 2"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 2';
#   _job.Salary = 70000;
#   _job.Level = 55;
#   _job.Department = 'EE';
```

```

#   _job.insert();
#
#   _employee.FirstName = 'Carl';
#   _employee.LastName = 'Gauss';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 1"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 1';
#   _job.Salary = 60000;
#   _job.Level = 50;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'John';
#   _employee.LastName = 'Doe';
#   _employee.Birthdate = 1\1\1980;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 1"/"Software" job and all assigned employees
#   _job.Title = 'Engineer 1';
#   _job.Salary = 60000;
#   _job.Level = 50;
#   _job.Department = 'Software';
#   _job.insert();
#
#   _employee.FirstName = 'Ken';
#   _employee.LastName = 'Silverman';
#   _employee.Birthdate = 1\1\1976;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   ttscommit;
#}
ENDSOURCE

```

***Element: FRM

; Microsoft Dynamics AX Forms unloaded

; -----

FRMVERSION 6

FORM #_Employee

PROPERTIES

Name #_Employee

ENDPROPERTIES

METHODS

Version: 3

SOURCE #classDeclaration

```

#public class FormRun extends ObjectRun
#{
#}
ENDSOURCE
ENDMETHODS
OBJECTBANK
PROPERTIES
ENDPROPERTIES

DATASOURCE
OBJECTPOOL
PROPERTIES
    Name                #_Employee
    Table                #_Employee
ENDPROPERTIES

FIELDLIST
ENDFIELDLIST
ENDOBJECTPOOL
METHODS
    Version: 3
ENDMETHODS
ENDDATASOURCE
ENDOBJECTBANK

REFERENCEDATASOURCES
ENDREFERENCEDATASOURCES

JOINS
ENDJOINS

PARTREFERENCES
ENDPARTREFERENCES

DESIGN
PROPERTIES
ENDPROPERTIES

CONTAINER
CONTROL GRID
PROPERTIES
    Name                #Grid
    Width                #Column width
    DataSource           #_Employee
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
CONTAINER
CONTROL STRINGEDIT
PROPERTIES

```

```

        Name                #_Employee_FirstName
        DataSource           #_Employee
        DataField            #FirstName
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
ENDCONTROL

CONTROL STRINGEDIT
    PROPERTIES
        Name                #_Employee_LastName
        DataSource           #_Employee
        DataField            #LastName
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
ENDCONTROL

CONTROL DATEEDIT
    PROPERTIES
        Name                #_Employee_Birthdate
        DataSource           #_Employee
        DataField            #Birthdate
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
ENDCONTROL

CONTROL REFERENCEGROUP
    PROPERTIES
        Name                #ReferenceGroup
        DataSource           #_Employee
        ReferenceField       #Job
        ReplacementFieldGroup #AutoIdentification_2
        BackgroundColor      #Window background
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
    CONTAINER
    ENDCONTAINER

ENDCONTROL

ENDCONTAINER

```

```

ENDCONTROL

ENDCONTAINER

ENDDSIGN

ENDFORM

***Element: DBT

; Microsoft Dynamics AX Table : _Employee unloaded
; -----
TABLEVERSION 1

TABLE #_Employee
    EnforceFKRelation 1
    PROPERTIES
        Name                #_Employee
        CreateRecIdIndex    #Yes
        PrimaryIndex        #SurrogateKey
        ClusterIndex        #SurrogateKey
    ENDPROPERTIES

    FIELDS
        FIELD #FirstName
            STRING
            PROPERTIES
                Name                #FirstName
                Table                #_Employee
            ENDPROPERTIES

        FIELD #LastName
            STRING
            PROPERTIES
                Name                #LastName
                Table                #_Employee
            ENDPROPERTIES

        FIELD #Birthdate
            DATE
            PROPERTIES
                Name                #Birthdate
                Table                #_Employee
            ENDPROPERTIES

        FIELD #Job
            INT64
            PROPERTIES
                Name                #Job
                Table                #_Employee
    
```

```

        ENDPROPERTIES

    ENDFIELDS
    GROUPS
    ENDGROUPS

    INDICES
    ENDINDICES
    REFERENCES
        REFERENCE #_Job
            PROPERTIES
                Name          #_Job
                Table         #_Job
            ENDPROPERTIES

        FIELDREFERENCES
            REFERENCE TYPE PKFK
            PROPERTIES
                Field          #Job
                RelatedField   #RecId
            ENDPROPERTIES

        ENDFIELDREFERENCES
    ENDREFERENCE
    ENDMETHODS
    ENDDDELETEACTIONS
    ENDDDELETEACTIONS

    METHODS
        Version: 3
    ENDMETHODS
ENDTABLE

```

***Element: DBT

; Microsoft Dynamics AX Table : _Job unloaded

; -----

```

TABLEVERSION 1

TABLE #_Job
    EnforceFKRelation 1
    PROPERTIES
        Name          #_Job
        CreateRecIdIndex #Yes
        PrimaryIndex   #SurrogateKey
        ClusterIndex   #SurrogateKey
        NaturalKey     #AlternateKey1
    ENDPROPERTIES

    FIELDS

```

```

FIELD #Title
STRING
PROPERTIES
    Name          #Title
    Table         #_Job
ENDPROPERTIES

FIELD #Salary
REAL
PROPERTIES
    Name          #Salary
    Table         #_Job
ENDPROPERTIES

FIELD #Level
INT
PROPERTIES
    Name          #Level
    Table         #_Job
ENDPROPERTIES

FIELD #Department
STRING
PROPERTIES
    Name          #Department
    Table         #_Job
ENDPROPERTIES

ENDFIELDS
GROUPS
GROUP #AutoIdentification
PROPERTIES
    Name          #AutoIdentification
ENDPROPERTIES

GROUPFIELDS
    #Title
    #Department
ENDGROUPFIELDS
ENDGROUP
GROUP #AutoIdentification_2
PROPERTIES
    Name          #AutoIdentification_2
ENDPROPERTIES

GROUPFIELDS
    #Title
ENDGROUPFIELDS
ENDGROUP
ENDGROUPS

INDICES

```

```

#AlternateKey1
PROPERTIES
    Name                #AlternateKey1
    AllowDuplicates      #No
    AlternateKey         #Yes
ENDPROPERTIES

INDEXFIELDS
    #Title
    #Department
ENDINDEXFIELDS

ENDINDICES
REFERENCES
ENDREFERENCES

DELETEACTIONS
ENDDELETEACTIONS

METHODS
    Version: 3
ENDMETHODS
ENDTABLE

```

***Element: PRN

; Microsoft Dynamics AX Project : ReferenceGroupSample unloaded

; -----

```

PROJECTVERSION 2

PROJECT #ReferenceGroupSample
SHARED
PROPERTIES
    Name                #ReferenceGroupSample
ENDPROPERTIES

PROJECTCLASS ProjectNode
BEGINNODE
    FILETYPE 0
    UTILTYPE 5
    UTILOBJECTID 0
    NODETYPE 215
    NAME #CreateDataForReferenceGroupSample
ENDNODE
BEGINNODE
    FILETYPE 0
    UTILTYPE 11
    UTILOBJECTID 0
    NODETYPE 201
    NAME #_Employee
ENDNODE

```

```
BEGINNODE
  FILETYPE 0
  UTILTYPE 44
  UTILOBJECTID 6187
  NODETYPE 204
  NAME #_Employee
ENDNODE
BEGINNODE
  FILETYPE 0
  UTILTYPE 44
  UTILOBJECTID 6188
  NODETYPE 204
  NAME #_Job
ENDNODE
ENDPROJECT
```

```
***Element: END
```

Appendix C: SharedProject_ReferenceGroupSample_3

Exportfile for AOT version 1.0 or later

Formatversion: 1

***Element: JOB

; Microsoft Dynamics AX Job: CreateDataForReferenceGroupSample unloaded

; -----

JOBVERSION 1

```
SOURCE #CreateDataForReferenceGroupSample
#static void CreateDataForReferenceGroupSample(Args _args)
#{
#   _Employee _employee;
#   _Job _job;
#
#   ttsbegin;
#
#   delete_from _employee;
#   delete_from _job;
#
#   // Insert the "CEO"/"EDO" job and all assigned employees
#   _job.Title = 'CEO';
#   _job.Salary = 600000;
#   _job.Level = 90;
#   _job.Department = 'EOD';
#   _job.insert();
#
#   _employee.FirstName = 'Kal';
#   _employee.LastName = 'El';
#   _employee.Birthdate = 1\1\1950;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Physicist"/"EE" job and all assigned employees
#   _job.Title = 'Physicist';
#   _job.Salary = 70000;
#   _job.Level = 55;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'Bruce';
#   _employee.LastName = 'Banner';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 2"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 2';
#   _job.Salary = 70000;
#   _job.Level = 55;
```

```

#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'Carl';
#   _employee.LastName = 'Gauss';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 1"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 1';
#   _job.Salary = 60000;
#   _job.Level = 50;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'John';
#   _employee.LastName = 'Doe';
#   _employee.Birthdate = 1\1\1980;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 1"/"Software" job and all assigned employees
#   _job.Title = 'Engineer 1';
#   _job.Salary = 60000;
#   _job.Level = 50;
#   _job.Department = 'Software';
#   _job.insert();
#
#   _employee.FirstName = 'Ken';
#   _employee.LastName = 'Silverman';
#   _employee.Birthdate = 1\1\1976;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   ttscommit;
#}
ENDSOURCE

```

***Element: FRM

; Microsoft Dynamics AX Forms unloaded

; -----

FRMVERSION 6

FORM #_Employee

PROPERTIES

 Name #_Employee

ENDPROPERTIES

METHODS

 Version: 3

```

SOURCE #classDeclaration
    #public class FormRun extends ObjectRun
    #{
    #}
ENDSOURCE
ENDMETHODS
OBJECTBANK
PROPERTIES
ENDPROPERTIES

DATASOURCE
OBJECTPOOL
PROPERTIES
    Name                #_Employee
    Table                #_Employee
ENDPROPERTIES

FIELDLIST
REFERENCEFIELD Job
PROPERTIES
ENDPROPERTIES

METHODS
    Version: 3
    SOURCE #resolveReference
        #public Common resolveReference(FormReferenceControl _formReferenceControl)
        #{
        #     _Job resolvedRecord;
        #     str 100 title = '', defaultDepartmentStr = '';
        #
        #     // Do not call super as we're providing our own disambiguation logic.
        #     // resolvedRecord = super(_formReferenceControl);
        #
        #     title =
        #_formReferenceControl.filterValue(AbsoluteFieldBinding::construct(fieldStr(_Job, Title),
        tableStr(_Job))).value();
        #     defaultDepartmentStr = DefaultDepartment.valueStr();
        #
        #     select firstonly
        #         resolvedRecord
        #     where
        #         resolvedRecord.Title == title &&
        #         resolvedRecord.Department == defaultDepartmentStr;
        #
        #     return resolvedRecord;
        #}
        #
    ENDSOURCE
    ENDMETHODS
    ENDREFERENCEFIELD

ENDFIELDLIST

```

```

ENDOBJECTPOOL
METHODS
    Version: 3
ENDMETHODS
ENDDATASOURCE
ENDOBJECTBANK

REFERENCEDATASOURCES
ENDREFERENCEDATASOURCES

JOINS
ENDJOINS

PARTREFERENCES
ENDPARTREFERENCES

DESIGN
PROPERTIES
ENDPROPERTIES

CONTAINER
CONTROL STRINGEDIT
PROPERTIES
    Name                #DefaultDepartment
    AutoDeclaration     #Yes
    Label               #Default Department
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL GRID
PROPERTIES
    Name                #Grid
    Width               #Column width
    DataSource          #_Employee
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
CONTAINER
CONTROL STRINGEDIT
PROPERTIES
    Name                #_Employee_FirstName
    DataSource          #_Employee
    DataField           #FirstName
ENDPROPERTIES

METHODS

```

```

        Version: 3
    ENDMETHODS
ENDCONTROL

CONTROL STRINGEDIT
    PROPERTIES
        Name                #_Employee_LastName
        DataSource           #_Employee
        DataField            #LastName
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
ENDCONTROL

CONTROL DATEEDIT
    PROPERTIES
        Name                #_Employee_Birthdate
        DataSource           #_Employee
        DataField            #Birthdate
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
ENDCONTROL

CONTROL REFERENCEGROUP
    PROPERTIES
        Name                #ReferenceGroup
        DataSource           #_Employee
        ReferenceField       #Job
        ReplacementFieldGroup #AutoIdentification_2
        BackgroundColor     #Window background
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
    CONTAINER
    ENDCONTAINER

ENDCONTROL

ENDCONTAINER

ENDCONTROL

ENDCONTAINER

ENDDESIGN

```

ENDFORM

***Element: DBT

; Microsoft Dynamics AX Table : _Employee unloaded

; -----

TABLEVERSION 1

TABLE #_Employee

EnforceFKRelation 1

PROPERTIES

| | |
|------------------|---------------|
| Name | #_Employee |
| CreateRecIdIndex | #Yes |
| PrimaryIndex | #SurrogateKey |
| ClusterIndex | #SurrogateKey |

ENDPROPERTIES

FIELDS

FIELD #FirstName

STRING

PROPERTIES

| | |
|-------|------------|
| Name | #FirstName |
| Table | #_Employee |

ENDPROPERTIES

FIELD #LastName

STRING

PROPERTIES

| | |
|-------|------------|
| Name | #LastName |
| Table | #_Employee |

ENDPROPERTIES

FIELD #Birthdate

DATE

PROPERTIES

| | |
|-------|------------|
| Name | #Birthdate |
| Table | #_Employee |

ENDPROPERTIES

FIELD #Job

INT64

PROPERTIES

| | |
|-------|------------|
| Name | #Job |
| Table | #_Employee |

ENDPROPERTIES

ENDFIELDS

GROUPS

ENDGROUPS

```

INDICES
ENDINDICES
REFERENCES
  REFERENCE #_Job
  PROPERTIES
    Name #_Job
    Table #_Job
  ENDPROPERTIES

  FIELDREFERENCES
  REFERENCETYPE PKFK
  PROPERTIES
    Field #Job
    RelatedField #RecId
  ENDPROPERTIES

  ENDFIELDREFERENCES
ENDREFERENCE
ENDREFERENCES

DELETEACTIONS
ENDELETEACTIONS

METHODS
  Version: 3
ENDMETHODS
ENDTABLE

```

***Element: DBT

; Microsoft Dynamics AX Table : _Job unloaded

; -----

TABLEVERSION 1

TABLE #_Job

EnforceFKRelation 1

PROPERTIES

```

  Name #_Job
  CreateRecIdIndex #Yes
  PrimaryIndex #SurrogateKey
  ClusterIndex #SurrogateKey
  ReplacementKey #AlternateKey1

```

ENDPROPERTIES

FIELDS

FIELD #Title

STRING

PROPERTIES

```

  Name #Title
  Table #_Job

```

ENDPROPERTIES

```

FIELD #Salary
REAL
PROPERTIES
    Name          #Salary
    Table         #_Job
ENDPROPERTIES

FIELD #Level
INT
PROPERTIES
    Name          #Level
    Table         #_Job
ENDPROPERTIES

FIELD #Department
STRING
PROPERTIES
    Name          #Department
    Table         #_Job
ENDPROPERTIES

ENDFIELDS
GROUPS
GROUP #AutoIdentification
PROPERTIES
    Name          #AutoIdentification
ENDPROPERTIES

GROUPFIELDS
    #Title
    #Department
ENDGROUPFIELDS
ENDGROUP
GROUP #AutoIdentification_2
PROPERTIES
    Name          #AutoIdentification_2
ENDPROPERTIES

GROUPFIELDS
    #Title
ENDGROUPFIELDS
ENDGROUP
ENDGROUPS

INDICES
#AlternateKey1
PROPERTIES
    Name          #AlternateKey1
    AllowDuplicates #No
    AlternateKey   #Yes
ENDPROPERTIES

```

```
INDEXFIELDS
  #Title
  #Department
ENDINDEXFIELDS
```

```
ENDINDICES
REFERENCES
ENDREFERENCES
```

```
DELETEACTIONS
ENDDELETEACTIONS
```

```
METHODS
  Version: 3
ENDMETHODS
ENDTABLE
```

***Element: PRN

; Microsoft Dynamics AX Project : ReferenceGroupSample unloaded

; -----
PROJECTVERSION 2

```
PROJECT #ReferenceGroupSample
  SHARED
PROPERTIES
  Name          #ReferenceGroupSample
ENDPROPERTIES
```

```
PROJECTCLASS ProjectNode
BEGINNODE
  FILETYPE 0
  UTILTYPE 5
  UTILOBJECTID 0
  NODETYPE 215
  NAME #CreateDataForReferenceGroupSample
ENDNODE
```

```
BEGINNODE
  FILETYPE 0
  UTILTYPE 11
  UTILOBJECTID 0
  NODETYPE 201
  NAME #_Employee
ENDNODE
```

```
BEGINNODE
  FILETYPE 0
  UTILTYPE 44
  UTILOBJECTID 6187
  NODETYPE 204
  NAME #_Employee
```

```
ENDNODE
BEGINNODE
  FILETYPE 0
  UTILTYPE 44
  UTILOBJECTID 6188
  NODETYPE 204
  NAME #_Job
ENDNODE
ENDPROJECT
```

```
***Element: END
```

Appendix D: SharedProject_ReferenceGroupSample_4

Exportfile for AOT version 1.0 or later

Formatversion: 1

***Element: JOB

; Microsoft Dynamics AX Job: CreateDataForReferenceGroupSample unloaded

; -----

JOBVERSION 1

```
SOURCE #CreateDataForReferenceGroupSample
#static void CreateDataForReferenceGroupSample(Args _args)
#{
#   _Employee _employee;
#   _Job _job;
#
#   ttsbegin;
#
#   delete_from _employee;
#   delete_from _job;
#
#   // Insert the "CEO"/"EDO" job and all assigned employees
#   _job.Title = 'CEO';
#   _job.Salary = 600000;
#   _job.Level = 90;
#   _job.Department = 'EOD';
#   _job.insert();
#
#   _employee.FirstName = 'Kal';
#   _employee.LastName = 'El';
#   _employee.Birthdate = 1\1\1950;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Physicist"/"EE" job and all assigned employees
#   _job.Title = 'Physicist';
#   _job.Salary = 70000;
#   _job.Level = 55;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'Bruce';
#   _employee.LastName = 'Banner';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 2"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 2';
#   _job.Salary = 70000;
#   _job.Level = 55;
```

```

#     _job.Department = 'EE';
#     _job.insert();
#
#     _employee.FirstName = 'Carl';
#     _employee.LastName = 'Gauss';
#     _employee.Birthdate = 1\1\1940;
#     _employee.Job = _job.RecId;
#     _employee.insert();
#
#     // Insert the "Engineer 1"/"EE" job and all assigned employees
#     _job.Title = 'Engineer 1';
#     _job.Salary = 60000;
#     _job.Level = 50;
#     _job.Department = 'EE';
#     _job.insert();
#
#     _employee.FirstName = 'John';
#     _employee.LastName = 'Doe';
#     _employee.Birthdate = 1\1\1980;
#     _employee.Job = _job.RecId;
#     _employee.insert();
#
#     // Insert the "Engineer 1"/"Software" job and all assigned employees
#     _job.Title = 'Engineer 1';
#     _job.Salary = 60000;
#     _job.Level = 50;
#     _job.Department = 'Software';
#     _job.insert();
#
#     _employee.FirstName = 'Ken';
#     _employee.LastName = 'Silverman';
#     _employee.Birthdate = 1\1\1976;
#     _employee.Job = _job.RecId;
#     _employee.insert();
#
#     ttscommit;
#}

```

ENDSOURCE

***Element: FRM

; Microsoft Dynamics AX Forms unloaded

; -----

FRMVERSION 6

FORM #_Employee

PROPERTIES

 Name #_Employee

ENDPROPERTIES

METHODS

 Version: 3

```

SOURCE #classDeclaration
    #public class FormRun extends ObjectRun
    #{
    #}
ENDSOURCE
ENDMETHODS
OBJECTBANK
PROPERTIES
ENDPROPERTIES

DATASOURCE
OBJECTPOOL
PROPERTIES
    Name                #_Employee
    Table                #_Employee
ENDPROPERTIES

FIELDLIST
REFERENCEFIELD Job
PROPERTIES
ENDPROPERTIES

METHODS
    Version: 3
    SOURCE #lookupReference
        #public Common lookupReference(FormReferenceControl _formReferenceControl)
        #{
        #    _Job selectedRecord;
        #    SysReferenceTableLookup sysTableLookup =
SysReferenceTableLookup::newParameters(tablenum(_Job),
        #
        #_formReferenceControl, true);
        #    Query lookupQuery;
        #    str 100 defaultDepartmentStr = '';
        #
        #    // Do not call super as we're providing our own custom lookup logic.
        #    // selectedRecord = super(_formReferenceControl);
        #
        #    defaultDepartmentStr = DefaultDepartment.valueStr();
        #
        #    // Display the Title and Department fields in the lookup form.
        #    sysTableLookup.addLookupfield(fieldnum(_job, Title));
        #    sysTableLookup.addLookupfield(fieldnum(_job, Department));
        #
        #    // Create a custom Query that filters on the Department field.
        #    lookupQuery = new Query();
        #    lookupQuery.addDataSource(tablenum(_Job)).addRange(fieldnum(_Job,
Department)).value(defaultDepartmentStr);
        #    sysTableLookup.parmQuery(lookupQuery);
        #
        #    // Return the record selected by the user.
        #    selectedRecord = sysTableLookup.performFormLookup();
        #
        #

```

```

        #     return selectedRecord;
    # }
    #
ENDSOURCE
SOURCE #resolveReference
    #public Common resolveReference(FormReferenceControl _formReferenceControl)
    # {
    #     _Job resolvedRecord;
    #     str 100 title = '', defaultDepartmentStr = '';
    #
    #     // Do not call super as we're providing our own disambiguation logic.
    #     // resolvedRecord = super(_formReferenceControl);
    #
    #     title =
    _formReferenceControl.filterValue(AbsoluteFieldBinding::construct(fieldStr(_Job, Title),
    tableStr(_Job))).value();
    #     defaultDepartmentStr = DefaultDepartment.valueStr();
    #
    #     select firstonly
    #         resolvedRecord
    #     where
    #         resolvedRecord.Title == title &&
    #         resolvedRecord.Department == defaultDepartmentStr;
    #
    #     return resolvedRecord;
    # }
    #
ENDSOURCE
ENDMETHODS
ENDREFERENCEFIELD

    ENDFIELDLIST
    ENDOBJECTPOOL
    METHODS
    Version: 3
    ENDMETHODS
    ENDDATASOURCE
    ENDOBJECTBANK

    REFERENCEDATASOURCES
    ENDREFERENCEDATASOURCES

    JOINS
    ENDJOINS

    PARTREFERENCES
    ENDPARTREFERENCES

    DESIGN
    PROPERTIES
    ENDPROPERTIES

```

```

CONTAINER
CONTROL STRINGEDIT
PROPERTIES
    Name                #DefaultDepartment
    AutoDeclaration     #Yes
    Label               #Default Department
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL GRID
PROPERTIES
    Name                #Grid
    Width               #Column width
    DataSource          #_Employee
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
CONTAINER
CONTROL STRINGEDIT
PROPERTIES
    Name                #_Employee_FirstName
    DataSource          #_Employee
    DataField           #FirstName
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL STRINGEDIT
PROPERTIES
    Name                #_Employee_LastName
    DataSource          #_Employee
    DataField           #LastName
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL DATEEDIT
PROPERTIES
    Name                #_Employee_Birthdate
    DataSource          #_Employee

```

```

        DataField          #Birthdate
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
ENDCONTROL

CONTROL REFERENCEGROUP
PROPERTIES
    Name                  #ReferenceGroup
    DataSource             #_Employee
    ReferenceField         #Job
    ReplacementFieldGroup #AutoIdentification_2
    BackgroundColor       #Window background
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
CONTAINER
ENDCONTAINER

ENDCONTROL

ENDCONTAINER

ENDCONTROL

ENDCONTAINER

ENDDSIGN

ENDFORM

***Element: DBT

; Microsoft Dynamics AX Table : _Employee unloaded
; -----
TABLEVERSION 1

TABLE #_Employee
    EnforceFKRelation 1
    PROPERTIES
        Name                #_Employee
        CreateRecIdIndex    #Yes
        PrimaryIndex        #SurrogateKey
        ClusterIndex        #SurrogateKey
    ENDPROPERTIES

    FIELDS

```

```

FIELD #FirstName
STRING
PROPERTIES
    Name                #FirstName
    Table                #_Employee
ENDPROPERTIES

FIELD #LastName
STRING
PROPERTIES
    Name                #LastName
    Table                #_Employee
ENDPROPERTIES

FIELD #Birthdate
DATE
PROPERTIES
    Name                #Birthdate
    Table                #_Employee
ENDPROPERTIES

FIELD #Job
INT64
PROPERTIES
    Name                #Job
    Table                #_Employee
ENDPROPERTIES

ENDFIELDS
GROUPS
ENDGROUPS

INDICES
ENDINDICES
REFERENCES
REFERENCE #_Job
PROPERTIES
    Name                #_Job
    Table                #_Job
ENDPROPERTIES

FIELDREFERENCES
REFERENCETYPE PKFK
PROPERTIES
    Field                #Job
    RelatedField         #RecId
ENDPROPERTIES

ENDFIELDREFERENCES
ENDREFERENCE
ENDREFERENCES

```

```

DELETEACTIONS
ENDELETEACTIONS

METHODS
    Version: 3
ENDMETHODS
ENDTABLE

***Element: DBT

; Microsoft Dynamics AX Table : _Job unloaded
; -----
TABLEVERSION 1

TABLE #_Job
    EnforceFKRelation 1
    PROPERTIES
        Name                #_Job
        CreateRecIdIndex    #Yes
        PrimaryIndex        #SurrogateKey
        ClusterIndex        #SurrogateKey
        ReplacementKey      #AlternateKey1
    ENDPROPERTIES

    FIELDS
        FIELD #Title
            STRING
            PROPERTIES
                Name          #Title
                Table         #_Job
            ENDPROPERTIES

        FIELD #Salary
            REAL
            PROPERTIES
                Name          #Salary
                Table         #_Job
            ENDPROPERTIES

        FIELD #Level
            INT
            PROPERTIES
                Name          #Level
                Table         #_Job
            ENDPROPERTIES

        FIELD #Department
            STRING
            PROPERTIES
                Name          #Department
                Table         #_Job
    
```

```

        ENDPROPERTIES

    ENDFIELDS
    GROUPS
        GROUP #AutoIdentification
            PROPERTIES
                Name                #AutoIdentification
            ENDPROPERTIES

            GROUPFIELDS
                #Title
                #Department
            ENDGROUPFIELDS
        ENDGROUP
        GROUP #AutoIdentification_2
            PROPERTIES
                Name                #AutoIdentification_2
            ENDPROPERTIES

            GROUPFIELDS
                #Title
            ENDGROUPFIELDS
        ENDGROUP
    ENDGROUPS

    INDICES
        #AlternateKey1
        PROPERTIES
            Name                    #AlternateKey1
            AllowDuplicates         #No
            AlternateKey            #Yes
        ENDPROPERTIES

        INDEXFIELDS
            #Title
            #Department
        ENDINDEXFIELDS

    ENDINDICES
    REFERENCES
    ENDMETHODS

    DELETEDACTIONS
    ENDELETEDACTIONS

    METHODS
        Version: 3
    ENDMETHODS
ENDTABLE

```

***Element: PRN

```

; Microsoft Dynamics AX Project : ReferenceGroupSample unloaded
; -----
PROJECTVERSION 2

PROJECT #ReferenceGroupSample
  SHARED
PROPERTIES
  Name #ReferenceGroupSample
ENDPROPERTIES

PROJECTCLASS ProjectNode
BEGINNODE
  FILETYPE 0
  UTILTYPE 5
  UTILOBJECTID 0
  NODETYPE 215
  NAME #CreateDataForReferenceGroupSample
ENDNODE
BEGINNODE
  FILETYPE 0
  UTILTYPE 11
  UTILOBJECTID 0
  NODETYPE 201
  NAME #_Employee
ENDNODE
BEGINNODE
  FILETYPE 0
  UTILTYPE 44
  UTILOBJECTID 6187
  NODETYPE 204
  NAME #_Employee
ENDNODE
BEGINNODE
  FILETYPE 0
  UTILTYPE 44
  UTILOBJECTID 6188
  NODETYPE 204
  NAME #_Job
ENDNODE
ENDPROJECT

***Element: END

```

Appendix E: SharedProject_ReferenceGroupSample_5

Exportfile for AOT version 1.0 or later

Formatversion: 1

***Element: JOB

; Microsoft Dynamics AX Job: CreateDataForReferenceGroupSample unloaded

; -----

JOBVERSION 1

```
SOURCE #CreateDataForReferenceGroupSample
#static void CreateDataForReferenceGroupSample(Args _args)
#{
#   _Employee _employee;
#   _Job _job;
#
#   ttsbegin;
#
#   delete_from _employee;
#   delete_from _job;
#
#   // Insert the "CEO"/"EDO" job and all assigned employees
#   _job.Title = 'CEO';
#   _job.Salary = 600000;
#   _job.Level = 90;
#   _job.Department = 'EOD';
#   _job.insert();
#
#   _employee.FirstName = 'Kal';
#   _employee.LastName = 'El';
#   _employee.Birthdate = 1\1\1950;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Physicist"/"EE" job and all assigned employees
#   _job.Title = 'Physicist';
#   _job.Salary = 70000;
#   _job.Level = 55;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'Bruce';
#   _employee.LastName = 'Banner';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 2"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 2';
#   _job.Salary = 70000;
#   _job.Level = 55;
```

```

#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'Carl';
#   _employee.LastName = 'Gauss';
#   _employee.Birthdate = 1\1\1940;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 1"/"EE" job and all assigned employees
#   _job.Title = 'Engineer 1';
#   _job.Salary = 60000;
#   _job.Level = 50;
#   _job.Department = 'EE';
#   _job.insert();
#
#   _employee.FirstName = 'John';
#   _employee.LastName = 'Doe';
#   _employee.Birthdate = 1\1\1980;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   // Insert the "Engineer 1"/"Software" job and all assigned employees
#   _job.Title = 'Engineer 1';
#   _job.Salary = 60000;
#   _job.Level = 50;
#   _job.Department = 'Software';
#   _job.insert();
#
#   _employee.FirstName = 'Ken';
#   _employee.LastName = 'Silverman';
#   _employee.Birthdate = 1\1\1976;
#   _employee.Job = _job.RecId;
#   _employee.insert();
#
#   ttscommit;
#}

```

ENDSOURCE

***Element: FRM

; Microsoft Dynamics AX Forms unloaded

; -----

FRMVERSION 6

FORM #_Employee

PROPERTIES

 Name #_Employee

ENDPROPERTIES

METHODS

 Version: 3

```

SOURCE #classDeclaration
    #public class FormRun extends ObjectRun
    #{
    #}
ENDSOURCE
ENDMETHODS
OBJECTBANK
PROPERTIES
ENDPROPERTIES

DATASOURCE
    OBJECTPOOL
        PROPERTIES
            Name                #_Employee
            Table                #_Employee
        ENDPROPERTIES

        FIELDLIST
        ENDFIELDLIST
    ENDOBJECTPOOL
    METHODS
        Version: 3
    ENDMETHODS
ENDDATASOURCE
ENDOBJECTBANK

REFERENCEDATASOURCES
    DATASOURCE
        OBJECTPOOL
            PROPERTIES
                Name            #_Job
                Table            #_Job
                JoinSource       #_Employee
                JoinRelation     #_Job
            ENDPROPERTIES

            FIELDLIST
            ENDFIELDLIST
        ENDOBJECTPOOL
        METHODS
            Version: 3
        ENDMETHODS
    ENDDATASOURCE
ENDREFERENCEDATASOURCES

JOINS
    _Job _Employee
ENDJOINS

PARTREFERENCES
ENDPARTREFERENCES

```

```

DESIGN
  PROPERTIES
ENDPROPERTIES

CONTAINER
  CONTROL TAB
    PROPERTIES
      Name          #Tab
      Width         #Column width
    ENDPROPERTIES

    METHODS
      Version: 3
    ENDMETHODS

  CONTAINER
    CONTROL TABPAGE
      PROPERTIES
        Name          #Overview
        Width         #Column width
        Caption       #Overview
      ENDPROPERTIES

      METHODS
        Version: 3
      ENDMETHODS

    CONTAINER
      CONTROL GRID
        PROPERTIES
          Name          #Grid
          Width         #Column width
          DataSource    #_Employee
        ENDPROPERTIES

        METHODS
          Version: 3
        ENDMETHODS

      CONTAINER
        CONTROL STRINGEDIT
          PROPERTIES
            Name          #_Employee_FirstName
            DataSource    #_Employee
            DataField     #FirstName
          ENDPROPERTIES

          METHODS
            Version: 3
          ENDMETHODS
        ENDCONTROL

      CONTROL STRINGEDIT
        PROPERTIES
          Name          #_Employee_LastName

```

```

        DataSource      #_Employee
        DataField      #LastName
    ENDPROPERTIES

    METHODS
        Version: 3
    ENDMETHODS
ENDCONTROL

CONTROL DATEEDIT
PROPERTIES
    Name                #_Employee_Birthdate
    DataSource          #_Employee
    DataField           #Birthdate
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL REFERENCEGROUP
PROPERTIES
    Name                #ReferenceGroup
    DataSource          #_Employee
    ReferenceField      #Job
    BackgroundColor    #Window background
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
CONTAINER
ENDCONTAINER

ENDCONTROL

ENDCONTAINER

ENDCONTROL

ENDCONTAINER

ENDCONTROL

CONTROL TABPAGE
PROPERTIES
    Name                #JobDetails
    Caption             #Job Details
ENDPROPERTIES

METHODS

```

```

Version: 3
ENDMETHODS
CONTAINER
CONTROL STRINGEDIT
PROPERTIES
    Name                #_Job_Title
    DataSource           #_Job
    DataField            #Title
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL REALEDIT
PROPERTIES
    Name                #_Job_Salary
    DataSource           #_Job
    DataField            #Salary
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL INTEDIT
PROPERTIES
    Name                #_Job_Level
    DataSource           #_Job
    DataField            #Level
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

CONTROL STRINGEDIT
PROPERTIES
    Name                #_Job_Department
    DataSource           #_Job
    DataField            #Department
ENDPROPERTIES

METHODS
    Version: 3
ENDMETHODS
ENDCONTROL

ENDCONTAINER

```

```

        ENDCONTROL

    ENDCONTAINER

ENDCONTROL

ENDCONTAINER

ENDDSIGN

ENDFORM

***Element: DBT

; Microsoft Dynamics AX Table : _Employee unloaded
; -----
TABLEVERSION 1

TABLE #_Employee
    EnforceFKRelation 1
    PROPERTIES
        Name                #_Employee
        CreateRecIdIndex    #Yes
        PrimaryIndex        #SurrogateKey
        ClusterIndex        #SurrogateKey
    ENDPROPERTIES

    FIELDS
        FIELD #FirstName
            STRING
            PROPERTIES
                Name                #FirstName
                Table                #_Employee
            ENDPROPERTIES

        FIELD #LastName
            STRING
            PROPERTIES
                Name                #LastName
                Table                #_Employee
            ENDPROPERTIES

        FIELD #Birthdate
            DATE
            PROPERTIES
                Name                #Birthdate
                Table                #_Employee
            ENDPROPERTIES

        FIELD #Job

```

```

INT64
PROPERTIES
  Name          #Job
  Table         #_Employee
ENDPROPERTIES

```

```

ENDFIELDS
GROUPS
ENDGROUPS

```

```

INDICES
ENDINDICES
REFERENCES

```

```

REFERENCE #_Job
PROPERTIES
  Name          #_Job
  Table         #_Job
ENDPROPERTIES

```

```

FIELDREFERENCES
REFERENCETYPE PKFK
PROPERTIES
  Field          #Job
  RelatedField   #RecId
ENDPROPERTIES

```

```

ENDFIELDREFERENCES
ENDREFERENCE
ENDREFERENCES

```

```

DELETEACTIONS
ENDDELETEACTIONS

```

```

METHODS
  Version: 3
ENDMETHODS
ENDTABLE

```

***Element: DBT

; Microsoft Dynamics AX Table : _Job unloaded

; -----

TABLEVERSION 1

```

TABLE #_Job
  EnforceFKRelation 1
PROPERTIES
  Name          #_Job
  CreateRecIdIndex #Yes
  PrimaryIndex   #SurrogateKey
  ClusterIndex   #SurrogateKey

```

```

ReplacementKey          #AlternateKey1
ENDPROPERTIES

FIELDS
FIELD #Title
STRING
PROPERTIES
Name                    #Title
Table                   #_Job
ENDPROPERTIES

FIELD #Salary
REAL
PROPERTIES
Name                    #Salary
Table                   #_Job
ENDPROPERTIES

FIELD #Level
INT
PROPERTIES
Name                    #Level
Table                   #_Job
ENDPROPERTIES

FIELD #Department
STRING
PROPERTIES
Name                    #Department
Table                   #_Job
ENDPROPERTIES

ENDFIELDS
GROUPS
GROUP #AutoIdentification
PROPERTIES
Name                    #AutoIdentification
ENDPROPERTIES

GROUPFIELDS
#Title
#Department
ENDGROUPFIELDS
ENDGROUP
GROUP #AutoIdentification_2
PROPERTIES
Name                    #AutoIdentification_2
ENDPROPERTIES

GROUPFIELDS
#Title
ENDGROUPFIELDS

```

```
ENDGROUP
ENDGROUPS
```

```
INDICES
#AlternateKey1
PROPERTIES
  Name #AlternateKey1
  AllowDuplicates #No
  AlternateKey #Yes
ENDPROPERTIES
```

```
INDEXFIELDS
#Title
#Department
ENDINDEXFIELDS
```

```
ENDINDICES
REFERENCES
ENDREFERENCES
```

```
DELETEACTIONS
ENDDELETEACTIONS
```

```
METHODS
  Version: 3
ENDMETHODS
ENDTABLE
```

```
***Element: PRN
```

```
; Microsoft Dynamics AX Project : ReferenceGroupSample unloaded
```

```
; -----
```

```
PROJECTVERSION 2
```

```
PROJECT #ReferenceGroupSample
SHARED
PROPERTIES
  Name #ReferenceGroupSample
ENDPROPERTIES
```

```
PROJECTCLASS ProjectNode
BEGINNODE
  FILETYPE 0
  UTILTYPE 5
  UTILOBJECTID 0
  NODETYPE 215
  NAME #CreateDataForReferenceGroupSample
ENDNODE
BEGINNODE
  FILETYPE 0
  UTILTYPE 11
```

```
    UTILOBJECTID 0
    NODETYPE 201
    NAME #_Employee
ENDNODE
BEGINNODE
    FILETYPE 0
    UTILTYPE 44
    UTILOBJECTID 6187
    NODETYPE 204
    NAME #_Employee
ENDNODE
BEGINNODE
    FILETYPE 0
    UTILTYPE 44
    UTILOBJECTID 6188
    NODETYPE 204
    NAME #_Job
ENDNODE
ENDPROJECT
```

```
***Element: END
```

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989
Worldwide +1-701-281-6500
www.microsoft.com/dynamics

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

© 2012 Microsoft Corporation. All rights reserved.

Microsoft