Overview

Number sequences are unique identifiers that can be associated with a master record so that they can be individually distinguished. They can be either formatted as alpha-numeric strings or simply as numbers.

Microsoft Dynamics AX 2012 provides an easy to implement framework to generate custom number sequences.

Scenario

As part of this tutorial, a custom number sequence will be generated for the Customer Groups setup form (**Accounts receivable  Setup  Customers  Customer groups**)

Steps

1. First create a new **Extended Data Type (EDT)**. Open **AOT Data Dictionary  Extended Data Types**
2. Right Click on **Extended Data Types** and create a new EDT **NumSeqDemoCustGroupNum**of type **String**
3. Set the properties as shown below

1. Now go to **AOT  Classes** and open the **NumberSeqModuleCustomer** class by right clicking it and selecting **View Code**

1. In the **loadModule** method, add the following code after the last line of code

*//customer group number*

*//define the EDT*

datatype.parmDatatypeId(**extendedTypeNum**(NumSeqDemoCustGroupNum));

*//define its default properties*datatype.parmReferenceHelp(**literalStr**(“Unique number for customer group”));  
datatype.parmWizardIsContinuous(**true**);  
datatype.parmWizardIsManual(NoYes::No);  
datatype.parmWizardIsChangeDownAllowed(NoYes::No);  
datatype.parmWizardIsChangeUpAllowed(NoYes::No);  
datatype.parmWizardHighest(**999999**);  
datatype.parmSortField(**27**);

*//define its scope*

datatype.addParameterType(NumberSeqParameterType::DataArea, **true**, **false**);  
this.create(datatype);

1. Now, go to **AOT  Jobs** and create a new job **loadNumSeqCustDemo**

Write the following code in the job and then run it

**static void** loadNumSeqCustDemo(Args \_args)  
{  
*//define the class variable*NumberSeqModuleCustomer seqMod = **new** NumberSeqModuleCustomer();

*//load the number sequences that were not generated*seqMod.load();  
}

1. Now, go to **Organization administration  Common  Number sequences  Number sequences**
2. Click on **Generate** button in the **New** button group
3. In the Setup number sequences wizard, Press **Next**
4. In the Setup set different values for the number sequence like the format, highest value and lowest value
5. Click **Next**
6. In the last step, Click **Finish** to generate the number sequences
7. The number sequence is generated and can be used on the Customer Groups form
8. Open **AOT  Data Dictionary  Tables  CustGroup**
9. Add a new **String** field and set the properties as follows
10. Add the newly added field in the Overview field group
11. Now go to **Forms CustGroup** and restore the form. It will add the newly added field in the grid
12. Write the following code on the Class declaration node

 NumberSeqFormHandler numberSeqFormHandler;

1. Create a new method on the form and write the following code

NumberSeqFormHandler numberSeqFormHandler()  
{  
**if** (!numberSeqFormHandler)  
{

*//create a reference of number sequence form handler class specifying the         EDT, Data source name and the field of the table*

numberSeqFormHandler =NumberSeqFormHandler::newForm(NumberSeqReference::findReference(**extendedtypenum**(NumSeqDemoCustGroupNum)).NumberSequenceId, element,CustGroup\_DS,**fieldnum**(CustGroup,CustGroupNumber));  
}  
**return** numberSeqFormHandler;  
}

1. Override the **close** method of the form and write the following code

**public void** close()  
{  
**if** (numberSeqFormHandler)  
{  
numberSeqFormHandler.formMethodClose();  
}

**super**();  
}

1. Override the **create** method on the **CustGroup** data source and add the following code

**public void** create(**boolean** \_append = **false**)  
{  
element.numberSeqFormHandler().formMethodDataSourceCreatePre();

**super**(\_append);

element.numberSeqFormHandler().formMethodDataSourceCreate(**true**);  
}

1. Override the **write** method on the **CustGroup** data source and add the following code

**public void** write()  
{  
**super**();

element.numberSeqFormHandler().formMethodDataSourceWrite();  
}

1. Override the **validateWrite** method on the **CustGroup** data source and add the following code

**public boolean** validateWrite()  
{  
**boolean** ret;

ret = **super**();

ret = element.numberSeqFormHandler().formMethodDataSourceValidateWrite(ret) && ret;

**return** ret;  
}

1. Override the **delete** method on the **CustGroup** data source and add the following code

**public**  
**void** delete()  
{

element.numberSeqFormHandler().formMethodDataSourceDelete();

**super**();  
}

1. Override the **linkActive** method on the **CustGroup** data source and add the following code

**public**  
**void** linkActive()  
{

element.numberSeqFormHandler().formMethodDataSourceLinkActive();

**super**();  
}

1. Now go to **Accounts receivable  Setup  Customers  Customer groups**
2. Create a new record. The number sequence is generated according to the format defined as shown below