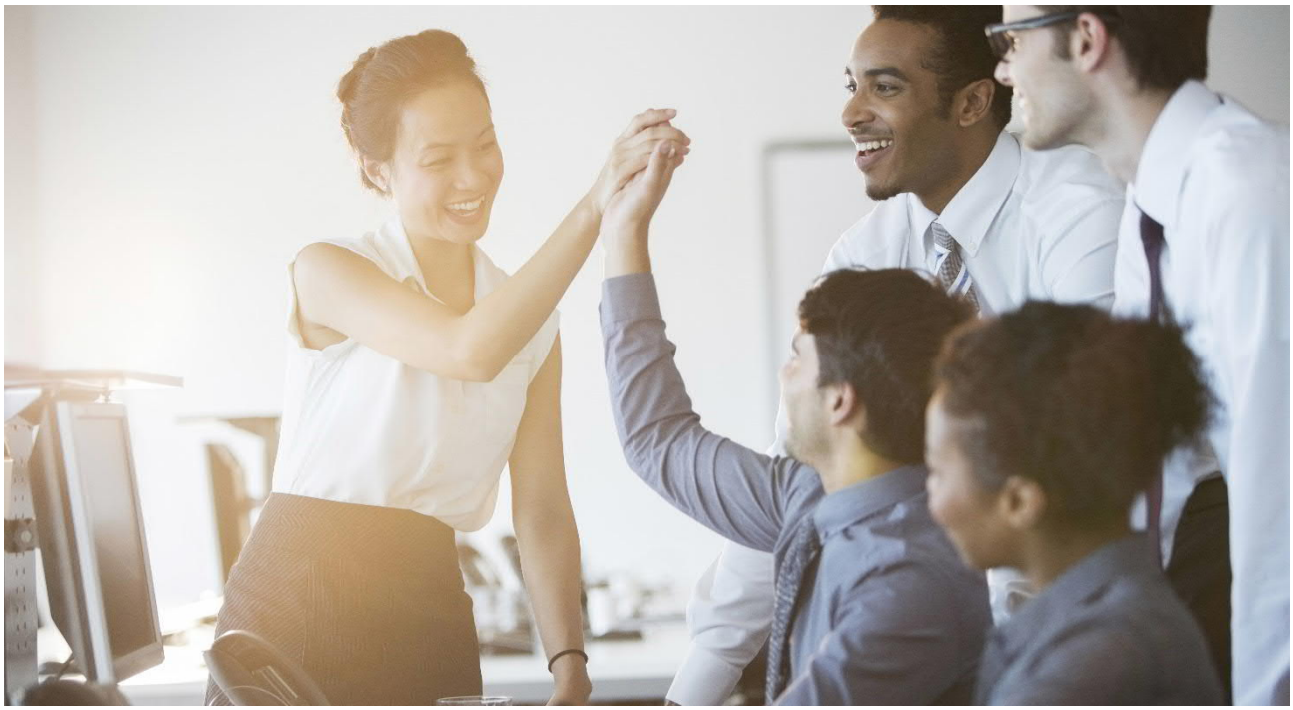


Exercise Solution: Meta Data

Data Interface API

PUBLIC

INTRODUCTION

In this exercise, you will perform the following tasks:

1. Create new buttons called “Add UDF and UDT”
2. Create User Defined Table
3. Remove User Defined Table
4. Enhance the User Defined Table with User Defined Fields
5. Insert records to the User Defined Table

PREREQUISITE:

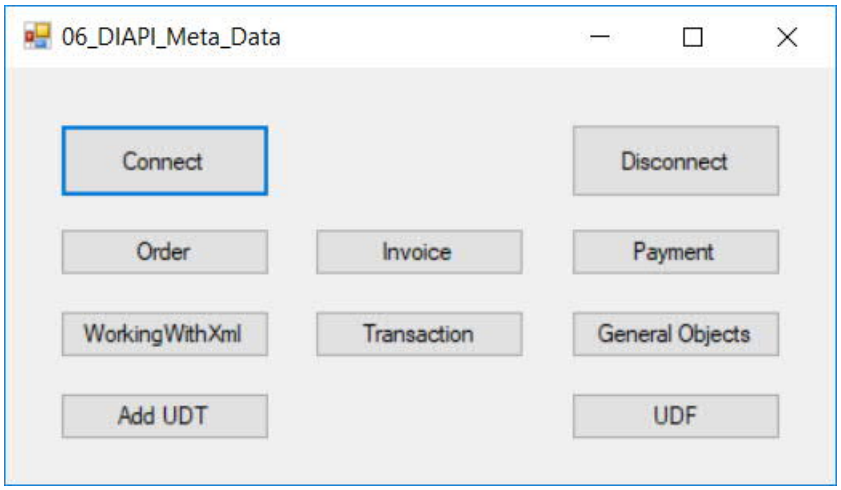
- This document is using the **C Sharp** (C#) language
- This document is using the Microsoft Visual Studio 2015
- Continue to work with the project finalized in previous exercise.
- Use the demo database for SAP Business One, version for SAP HANA or SAP Business One
- Credentials: User code: **manager**

GUIDELINES:

The screenshots provided here are for your reference only and may differ from the actual screenshots in your system.

1. TASK - CREATE A NEW BUTTONS CALLED “ADD UDF AND UDT”

1.1. On your Visual Studio project create a new button called “Add UDF and UDT”



2. TASK - CREATE USER FIELDS

2.1. As a first small exercise add a User-Defined Field to the item table (OITM) through DI API.

Use namespace “TB1_” as a prefix.

Use the UserTableMD Object to create User Tables.

```
SAPbobsCOM.UserFieldsMD oUDF;
oUDF = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.oUserFields);

oUDF.TableName = "OITM";
oUDF.Name = "TB1_Course";
oUDF.Description = "Course UDF";
oUDF.Type = SAPbobsCOM.BoFieldTypes.db_Alpha;
oUDF.EditSize = 20;

int ret = oUDF.Add();
if (ret == 0)
       MessageBox.Show("UDF " + oUDF.Name + " added.");
Else
       MessageBox.Show("Failed to add UDF " + oUDF.Name + ": " +
oCompany.GetLastErrorDescription());
```

2.2. Release the User Defined Field Object

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(oUDT);
GC.Collect();
```

3. TASK - CREATE USER TABLE

3.1. Add a User-Defined Table (use namespace “TB1_” as a prefix…), but do not add any fields to the table yet.
Table name: TB1_VIDS
Table description: Video Management

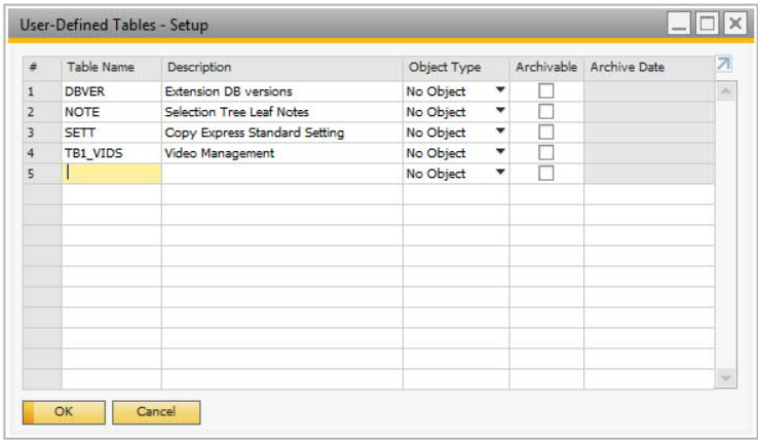
You will need to create an instance of the UserTablesMD object in order to add a field to the User Table. It is recommended that after you create your table you set this object variable to “Nothing” so that its properties do not inadvertently carry forward to the next table or field you are creating.

```
SAPbobsCOM.UserTablesMD oUDT;
string MyTableName;

oUDT =
(SAPbobsCOM.UserTablesMD)oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.oUserTables);
MyTableName = "TB1_VIDS";
if (oUDT.GetByKey(MyTableName) == false)
{
        oUDT.TableName = MyTableName;
        oUDT.TableDescription = "Video Management";
        int ret = oUDT.Add();

              if (ret == 0)
                  MessageBox.Show("Add Table: " + oUDT.TableName + " successfull");
              else
                  MessageBox.Show("Add Table error: " +
oCompany.GetLastErrorDescription());
}
Else
        MessageBox.Show("Table: " + MyTableName + " already exists");
```

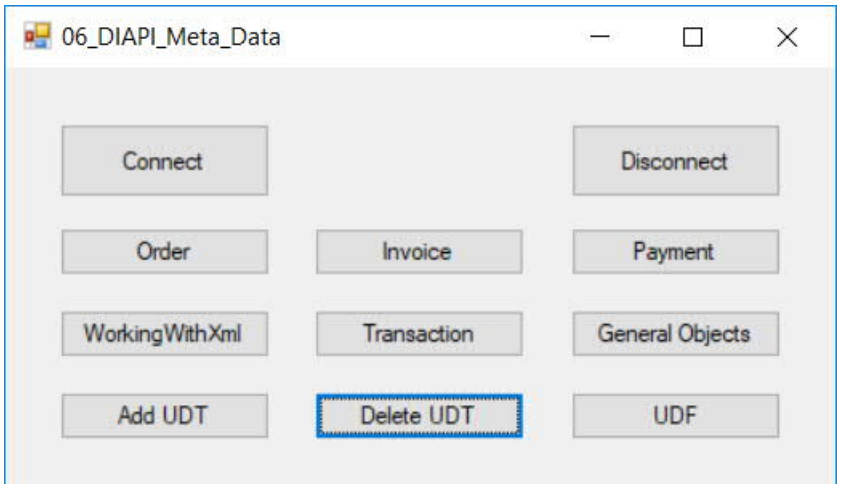
3.2. Release the User Defined Table Object after the creation operation.

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(oUDT);
GC.Collect();
```

3.3. Test your application by opening the “Manage User Fields” window in SAP Business One. Check to see that the table was added.



3.4. On your Visual Studio project create a new button called “Delete UDT”.



3.5. Remove the User-Defined Table (in the SAP Business One application) you just created before. Enhance your application with the capability to remove the User-Defined Table through DI API – and then test your application to see that you can also add and delete the User-Defined Table in SAP Business One.

```
SAPbobsCOM.UserTablesMD oUDT;
string MyTableName;

oUDT =
(SAPbobsCOM.UserTablesMD)oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.oUserTables);
MyTableName = "TB1_VIDS";
if (oUDT.GetByKey(MyTableName) == true)
{
      int ret = oUDT.Remove();

      if (ret == 0)
            MessageBox.Show("Remove Table: " + oUDT.TableName + " successfull");
      else
            MessageBox.Show("Remove Table error: " +
      oCompany.GetLastErrorDescription());
}
else
      MessageBox.Show("Table: " + MyTableName + " not exists");
```

3.6. Release the User Defined Table Object after the deletion operation.

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(oUDT);
GC.Collect();
```

4. TASK - ENHANCE THE USER DEFINED TABLE WITH USER DEFINED FIELDS

4.1. Create a Generic function to handle the User Defined Field creation.

```
        private void CreateField (string MyTableName, string MyFieldName, string
MyFieldDescrition, SAPbobsCOM.BoFieldTypes MyFieldType, int MyFieldSize)
        {
            try
            {
                SAPbobsCOM.UserFieldsMD oUDF;
                oUDF = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.oUserFields);

                oUDF.TableName = MyTableName;
                oUDF.Name = MyFieldName;
                oUDF.Description = MyFieldDescrition;
                oUDF.Type = MyFieldType;
                oUDF.EditSize = MyFieldSize;

                if (oUDF.Name == "RENTED")
                {
                    oUDF.ValidValues.Value = "Y";
                    oUDF.ValidValues.Description = "Yes";
                    oUDF.ValidValues.Add();
                    oUDF.ValidValues.Value = "N";
                    oUDF.ValidValues.Description = "No";
                    oUDF.ValidValues.Add();
                }

                int ret = oUDF.Add();
                if (ret == 0)
                    MessageBox.Show("UDF " + oUDF.Name + " added.");
                else
                    MessageBox.Show("Failed to add UDF " + oUDF.Name + ": " +
oCompany.GetLastErrorDescription());

                System.Runtime.InteropServices.Marshal.ReleaseComObject(oUDF);
                GC.Collect();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Exception: " + ex.Message);
            }

            }
```

4.2. Add the following User-Defined Fields to your new User-Defined Table.

You will need to create an instance of the UserTablesMD object in order to add a field to the User Table. It is recommended that after you create your table you set this object variable to “Nothing” so that its properties do not inadvertently carry forward to the next table or field you are creating.

Field Comment	Field Name	Field Description	Field Type	Field EditSize
Indicates in which aisle the movie is stored	AISLE	Aisle Number	db_Numeric	2
Indicates the section the movie is store in the aisle	SECTION	Section Number	db_Alpha	20
Indicates weather the movie is rented or not. Holds 2 “valid values”: Y/N.	RENTED	Rented/Available	db_Alpha	1
In case the movie is “Rented”. This field will hold the CardCode of the customer who rented it.Otherwise it will be empty. CardCode	CARDCODE	Card Code	db_Alpha	20

```
private void GenerateFields(string MyTableName)
 {
                  CreateField(MyTableName, "AISLE", "Aisle Number",
       SAPbobsCOM.BoFieldTypes.db_Numeric, 1);
                  CreateField(MyTableName, "SECTION", "Aisle Number",
       SAPbobsCOM.BoFieldTypes.db_Alpha, 20);
                  CreateField(MyTableName, "RENTED", "Aisle Number",
       SAPbobsCOM.BoFieldTypes.db_Alpha, 1);
                  CreateField(MyTableName, "CARDCODE", "Card Code",
       SAPbobsCOM.BoFieldTypes.db_Alpha, 20);
 }
```

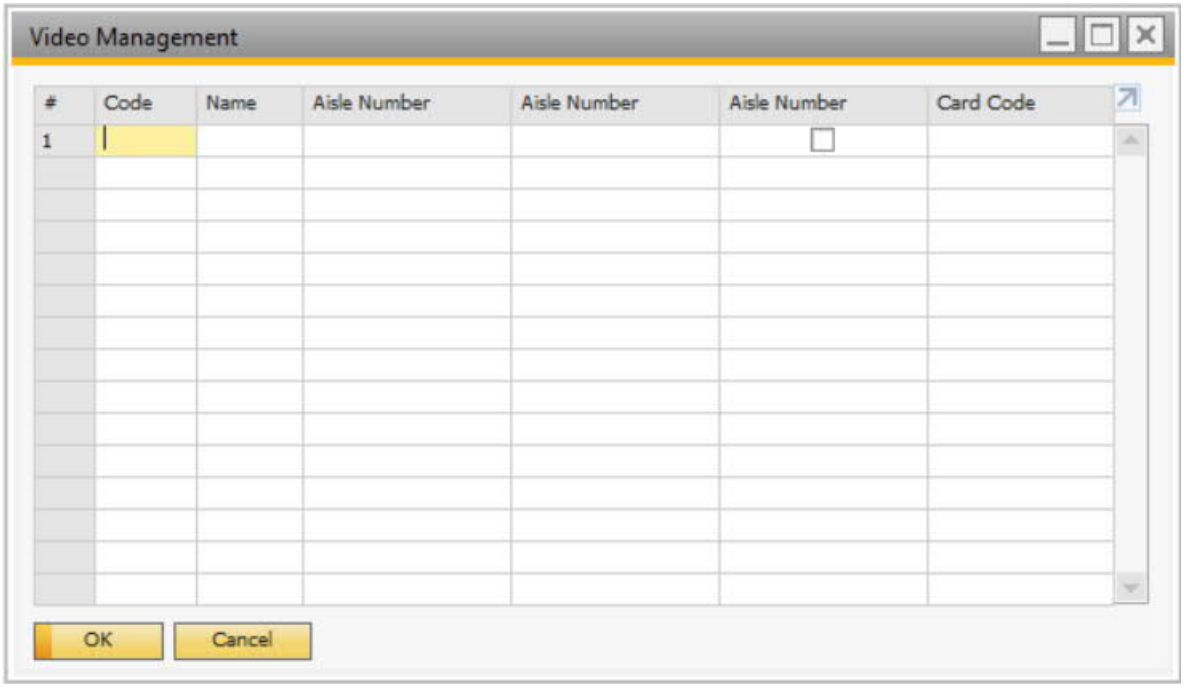
4.3. Place the *GenerateFields* function into the table creation function.

Release the User Defined Table Object before you will start to work with User Defined Fields. Otherwise you might get the error message documented in the SAP note 723420.

```
if (ret == 0){
       MessageBox.Show("Add Table: " + oUDT.TableName + " successfull");
       System.Runtime.InteropServices.Marshal.ReleaseComObject(oUDT);
       GC.Collect();
       GenerateFields(MyTableName);}
```

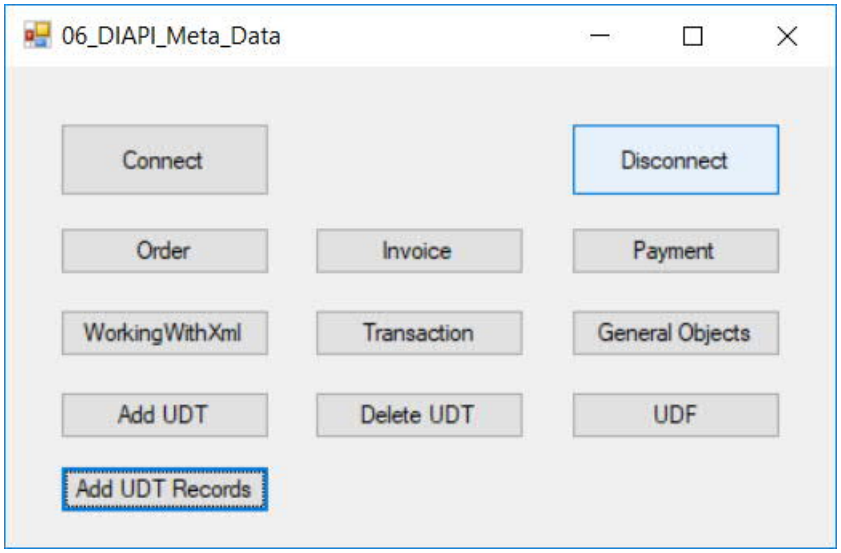
4.4. Test your application and make sure all your fields were added successfully.

Go to SAP Business One client → *Tools* → *User Defined Windows* → *TB1_VIDS*



5. TASK – INSERT RECORDS TO THE USER DEFINED TABLE

5.1. On your Visual Studio project create a new button called “Add UDT Records”.



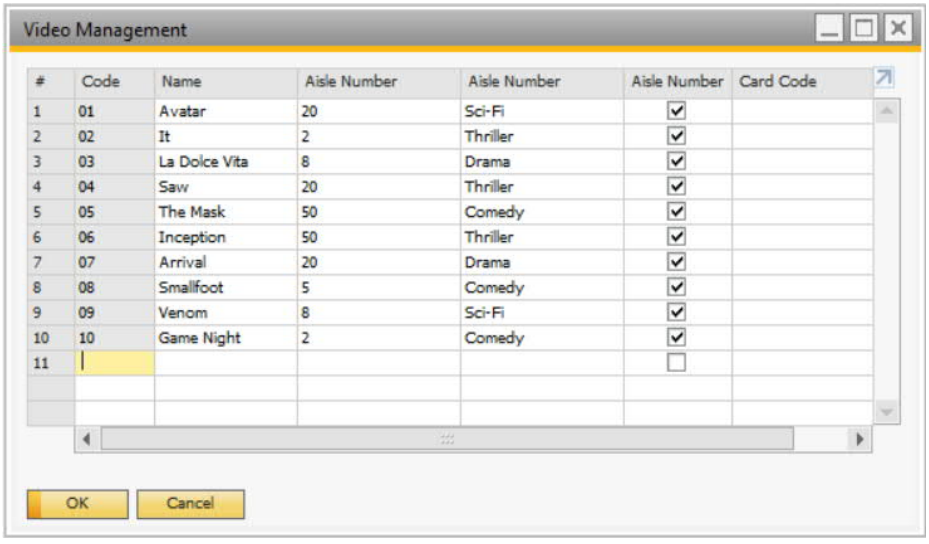
5.2. Create a generic function to insert data into the User Defined Field.

```
private void InsertRecordUDT(string Code, string Name, int Aisle, string Section,
string Rented)
        {
            SAPbobsCOM.UserTable oUDT;
            oUDT = oCompany.UserTables.Item("TB1_VIDS");
            oUDT.Code = Code;
            oUDT.Name = Name;
            oUDT.UserFields.Fields.Item("U_AISLE").Value = Aisle;
            oUDT.UserFields.Fields.Item("U_SECTION").Value = Section;
            oUDT.UserFields.Fields.Item("U_RENTED").Value = Rented;

            int ret = oUDT.Add();
            if (ret != 0)
                MessageBox.Show("Entry cannot be added: " +
oCompany.GetLastErrorDescription());

            System.Runtime.InteropServices.Marshal.ReleaseComObject(oUDT);
            GC.Collect();
        }
```

5.3. Add about 10 records to your new User-Defined Table.

```
InsertRecordUDT("01", "Avatar", 15, "Sci-Fi", "Y");
InsertRecordUDT("02", "It", 2, "Thriller", "Y");
InsertRecordUDT("03", "La Dolce Vita", 8, "Drama", "Y");
InsertRecordUDT("04", "Saw", 21, "Thriller", "Y");
InsertRecordUDT("05", "The Mask", 47, "Comedy", "Y");
InsertRecordUDT("06", "Inception", 53, "Thriller", "Y");
InsertRecordUDT("07", "Arrival", 20, "Drama", "Y");
InsertRecordUDT("08", "Smallfoot", 5, "Comedy", "Y");
InsertRecordUDT("09", "Venom", 8, "Sci-Fi", "Y");
InsertRecordUDT("10", "Game Night", 2, "Comedy", "Y");
```

5.4. Your User-Defined Table could look like this:



A similar solution can be found in the SDK samples (in the SDK Folder – see Appendix "SDK Installations" for more information), …\COM UI DI\CSharp\UIDIBasicApp\CreateUserTables

www.sap.com