

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:

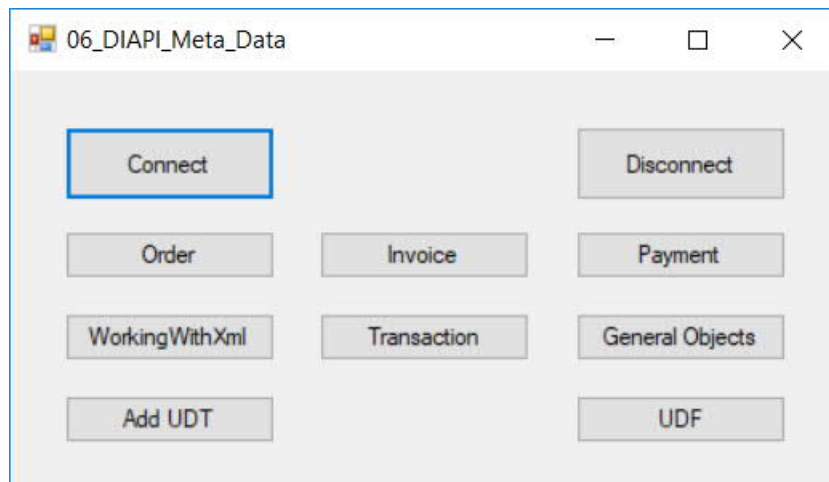
- This document is using the **C Sharp** (C#) language
- This document is using the Microsoft Visual Studio 2015
- Continue to work with the project finalized in previous exercise.
- Use the demo database for SAP Business One, version for SAP HANA or SAP Business One
- Credentials: User code: **manager**

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. UserTableMD oUDT;  
string MyTableName;
```

```

oUDT =
(SAPbobsCOM.UserTableMD)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 + " successful!");
    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();
```

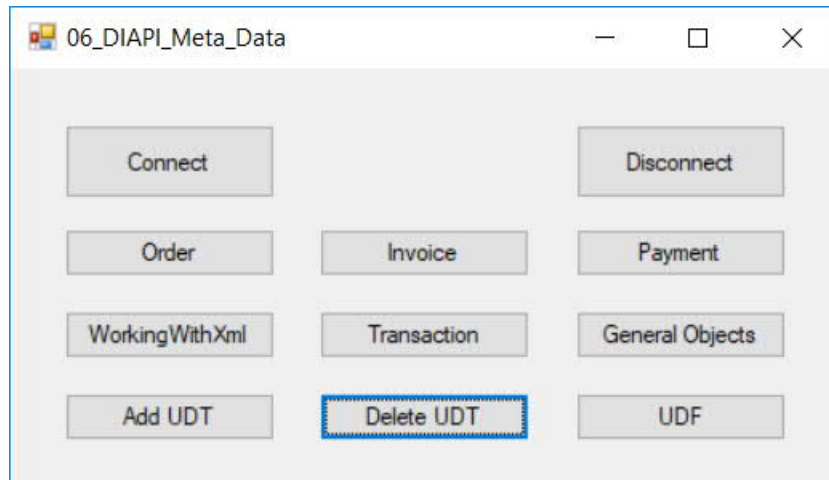
- 3.3. Test your application by opening the “Manage User Fields” window in SAP Business One. Check to see that the table was added.

The screenshot shows a window titled "User-Defined Tables - Setup". It contains a table with five columns: "#", "Table Name", "Description", "Object Type", "Archivable", and "Archive Date". There are four rows of data listed:

#	Table Name	Description	Object Type	Archivable	Archive Date
1	DVER	Extension DB versions	No Object ▼	<input type="checkbox"/>	
2	NOTE	Selection Tree Leaf Notes	No Object ▼	<input type="checkbox"/>	
3	SETT	Copy Express Standard Setting	No Object ▼	<input type="checkbox"/>	
4	TB1_VIDS	Video Management	No Object ▼	<input type="checkbox"/>	

The fifth row has a yellow background. At the bottom left are two buttons labeled "OK" and "Cancel". The top right corner has standard window control icons.

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 + " successful!");
    else
        MessageBox.Show("Remove Table error: " +
            oCompany. GetLastErrorMessage());
}
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
MyFieldDescription, SAPbobsCOM.BoFieldTypes MyFieldType, int MyFieldSize)
{
    try
    {
        SAPbobsCOM.UserFieldsMD oUDF;
        oUDF = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.oUserFields);

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

        if (oUDF.Name == "RENTED")
        {
            oUDF.ValueList.Value = "Y";
            oUDF.ValueList.Description = "Yes";
            oUDF.ValueList.Add();
            oUDF.ValueList.Value = "N";
            oUDF.ValueList.Description = "No";
            oUDF.ValueList.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 + " successful!");
    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*

Video Management

#

Code

Name

Aisle Number

Aisle Number

Aisle Number

Card Code

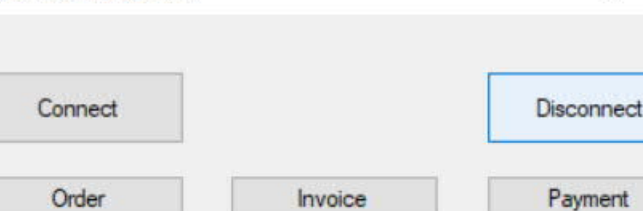
1

OK

Cancel

5. TASK – INSERT RECORDS TO THE USER DEFINED TABLE

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



06_DIAPI_Meta_Data

Connect Disconnect

Order Invoice Payment

WorkingWithXml Transaction General Objects

Add UDT Delete UDT UDF

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:

#	Code	Name	Aisle Number	Aisle Number	Aisle Number	Card Code
1	01	Avatar	20	Sci-Fi	<input checked="" type="checkbox"/>	
2	02	It	2	Thriller	<input checked="" type="checkbox"/>	
3	03	La Dolce Vita	8	Drama	<input checked="" type="checkbox"/>	
4	04	Saw	20	Thriller	<input checked="" type="checkbox"/>	
5	05	The Mask	50	Comedy	<input checked="" type="checkbox"/>	
6	06	Inception	50	Thriller	<input checked="" type="checkbox"/>	
7	07	Arrival	20	Drama	<input checked="" type="checkbox"/>	
8	08	Smallfoot	5	Comedy	<input checked="" type="checkbox"/>	
9	09	Venom	8	Sci-Fi	<input checked="" type="checkbox"/>	
10	10	Game Night	2	Comedy	<input checked="" type="checkbox"/>	
11					<input type="checkbox"/>	

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



© 2018 SAP SE or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.
These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.