

Exercise Solution: XML

Data Interface API

PUBLIC

INTRODUCTION

In this exercise, you will perform the following tasks:

1. Create new buttons in your project called "Working with XML"
2. Save the Invoice created in the Documents exercise as XML
3. Modify the XML data obtained before and add it to the SAP Business One database

PREREQUISITE:

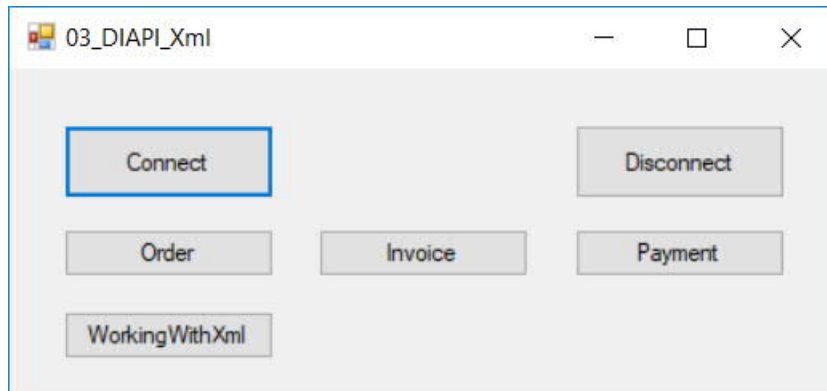
- This document is using the **C Sharp** (C#) language
- This document is using the Microsoft Visual Studio 2015
- Continue to work with the project finalized in previous exercise.
- Use the demo database for SAP Business One, version for SAP HANA or SAP Business One
- Credentials: User code: **manager**

GUIDELINES:

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

1. TASK - CREATE NEW BUTTONS IN YOUR PROJECT CALLED "WORKING WITH XML"

1.1. On your Visual Studio project create new buttons called "Working with XML"



2. TASK - CREATE S SALES ORDER

2.1. Try all settings for *XmlExportType* property on the *Company* object and find the differences.

The 4 property types are

XML ExportType	Definition
<i>oCompany.XmlExportType = SAPbobsCOM.BoXmlExportTypes.xet_AllNodes</i>	Export to XML all fields (both read only and read/write fields) from the database.
<i>oCompany.XmlExportType = SAPbobsCOM.BoXmlExportTypes.xet_ExportImportMode</i>	Export to XML only valid fields that support XML import (read/write fields only) from the database.
<i>oCompany.XmlExportType = SAPbobsCOM.BoXmlExportTypes.xet_NodesAsProperties</i>	Export to XML all fields as properties from the database.
<i>SAPbobsCOM.BoXmlExportTypes.xet_ValidNodesOnly</i>	Export to XML only valid fields that support XML import and export (read/write fields only that do not contain null values) from the database.

2.2. Save the Invoice document created in the previous exercise in Xml format.

```
oCompany.XmlExportType = SAPbobsCOM.BoXmlExportTypes.xet_ExportImportMode;
SAPbobsCOM.Documents olnv;
olnv = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.olnvoices);
if (olnv.GetByKey(Int32.Parse(MySalesInvoice)) == true)
{
    olnv.SaveXML("c:\\temp\\invoice_" + MySalesInvoice + ".xml");
    MessageBox.Show("Invoice " + MySalesInvoice + " exported to XML");
}
Else
    MessageBox.Show("Get Invoice failed: " + MySalesInvoice + " - " +
oCompany.GetLastErrorDescription());
```

2.3. Test also the method *GetBusinessObjectXmlSchema* of the Company object. What kind of information does it save?

```
string schema;
schema = oCompany.GetBusinessObjectXmlSchema(SAPbobsCOM.BoObjectTypes.olnvoices);
MessageBox.Show(schema);
```

This method retrieves the XML schema used to define the structure and content of the object.

3. TASK - 3. MODIFY THE XML DATA OBTAINED BEFORE AND ADD IT TO THE SAP BUSINESS ONE DATABASE

3.1. Try all files generated above and check the errors (exceptions) for details

```
olnv = oCompany.GetBusinessObjectFromXML("c:\\temp\\invoice_" + MySalesInvoice +
".xml", 0);
olnv.Comments = "invoice loaded from xml";
int ret = olnv.Add();
if (ret == 0)
{
    oCompany.GetNewObjectCode(out MySalesInvoice);
    MessageBox.Show("Invoice " + MySalesInvoice + " added from xml.");
}
Else
    MessageBox.Show("Adding invoice " + MySalesInvoice + " from XML failed: " +
oCompany.GetLastErrorDescription());
```

Similar exercises can be found in the SDK samples (in the SDK Folder – see Appendix “SDK Installations” for more information), COM DI/7.SaveXML and COM DI/8.LoadFromXML



© 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.