

**Everest Advanced**

# Everest Add-ins



**The only software you need to run your entire business... from bricks to clicks**

## Developer Guide



**Everest Software**  
YOUR BUSINESS OPERATING SYSTEM

**Copyright**

© 2008, Everest Software, Inc. All Rights Reserved.

V5.0

[www.everestsoftwareinc.com](http://www.everestsoftwareinc.com)

Email: [info@everestsoftwareinc.com](mailto:info@everestsoftwareinc.com)

**Notices**

No part of this developer guide may be used or reproduced in any form or by any means, or stored in a database or retrieval system, or translated, or otherwise duplicated in or any medium without the prior written permission of Everest Software, Inc. Making copies of any part of this book for any purpose other than for your personal use, which rights you cannot assign, sell, loan or otherwise distribute, is a violation of United States copyright laws and international treaty provisions and laws.

This book is sold/provided as is, without warranty of any kind, either expressed or implied, respecting the contents of this book, including but not limited to implied warranties for quality, performance, merchantability, or fitness for any particular purpose. Neither Everest Software, Inc., nor its dealers, resellers, or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, damage caused or alleged to have been caused directly or indirectly by this developer guide.

Information in this developer guide is subject to change without notice and does not represent a commitment on the part of Everest Software, Inc.

Use of the software programs described herein is subject to the Everest Software, Inc. License and Services Agreement enclosed with the software.

**Trademarks** Everest is a registered trademark of Everest Software, Inc. Windows, Microsoft, Visual Basic, and MSDN are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Borland, the Borland Logo, and Delphi are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. All other product names and services referenced herein are trademarks or service marks of their respective companies. Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.

**Restricted Rights** The Product is provided with restricted rights. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in the applicable License and Services Agreement and in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable.



# Table of Contents

<b>Introduction .....</b>	<b>1</b>
Introduction to Everest Add-Ins .....	3
Features of Everest Add-Ins.....	3
Assumptions.....	5
Document Structure .....	5
Document Conventions .....	6
<b>Developing Add-Ins for Everest .....</b>	<b>9</b>
About Everest Add-Ins .....	11
Getting Started .....	11
Quick Start .....	12
Everest Add-Ins.....	16
Overview .....	16
Add-In Workflow Example .....	17
Why XML? .....	18
Creating an Add-In for Everest .....	19
Step-by-Step Procedures for Developing Everest Add-Ins .....	26
Prerequisites.....	26
Creating Everest Add-Ins Using Visual Studio 2005 (C#) .....	27
Creating Everest Add-Ins Using Visual Basic 6.0.....	40
Creating Everest Add-Ins Using Delphi 7.....	46
Add-In Interface.....	58
IEverestAddin.....	58
Everest Public Interface .....	61
Everest Form .....	62
Everest Session .....	99
Everest Browser.....	102
Everest Profile.....	104
Everest Document .....	106
Everest Document Line Item .....	113
Everest Payment.....	119
Creating an Add-In Configuration File.....	123
Add-In Information Dialog Box .....	125
Installing Your Add-Ins.....	131
Add-In Validation .....	131
Add-In Registration.....	131
Add-In Execution .....	132
Everest Client Menu Hooks .....	133
Everest Client Event Hooks .....	133
<b>Everest Add-In Schemas.....</b>	<b>139</b>
Everest Add-In Schemas.....	141
Add-In Configuration File Schema .....	142
SessionData Schema.....	142
EverestForm.ExecuteAction Schema.....	142
sControlData Schema .....	143
PurchaseDocument Schema .....	144
PurchaseDocument Schema Continued .....	145

---

SalesDocument Schema.....	146
SalesDocument Schema Continued.....	147
LineItem Schema.....	148
LineItem Schema Continued.....	149
DeleteLineItem Schema.....	150
POSPayment Schema.....	151
POSPayment Schema Continued.....	152
Everest Add-In Samples.....	153
Working with Sample Add-ins.....	153
XML Samples.....	154
<b>Managing Everest Add-Ins.....</b>	<b>167</b>
Add-Ins.....	169
Add-In Manager Utility.....	170
Register Add-Ins.....	170
Manage Registered Add-Ins.....	171
View Registered Add-ins.....	172
Enable Registered Add-Ins.....	173
Disable Registered Add-Ins.....	173
Configure Add-Ins.....	173
Remove Add-Ins.....	174
Installing Add-Ins.....	174

# 1. Introduction

---

---

Introduction to Everest Add-Ins

Features of Everest Add-Ins

Assumptions

Document Structure

Document Conventions





## Introduction to Everest Add-Ins

To further enhance the power of **Everest**, **Everest Software** provides an **Everest Software Development Kit (SDK)** for developers, contractors, and value-added resellers (VARs). **Everest SDK** supports the development of custom-fit, practical solutions without requiring in-depth knowledge of **Everest's** architecture. SDK programs use the regular architecture for **Everest** transactions. **Everest SDK** contains the following components:

- **Everest API** - Allows programmatic access and integrates with **Everest Business Objects**
- **EDM Case Tool** - Contains supporting documentation used to integrate with the **Everest Database**
- **Everest Forms Designer** - Allows customization of the layout and design of **Everest forms**
- **Everest Add-Ins** - Allows programs to integrate with the **Everest Client**

The **Everest Add-Ins** framework is part of the SDK and is used to enhance **Everest Client** forms (documents, profiles, and browsers) by allowing third-party add-ins to attach to them. An **Add-In** is a supplemental program that extends the functionality of the main **Everest** application. **Everest Add-Ins** consists of COM Components and XML configuration files that are created by developers. XML configuration files provide information about the location of the COM DLL and show how the add-in is launched. Add-ins can be launched from **Everest** menu items or events.

After the add-in is developed, it is installed into the **Everest Client** application. When enabled, the end-users can use the add-in in **Everest**. End users with the appropriate security right can manage **Everest Add-Ins** using the **Add-In Manager** utility.

## Features of Everest Add-Ins

**Everest Add-Ins** allow you to develop software that supports the **Everest Client** interface:

- Customize and extend **Everest** functionality
  - Helps you provide “end-to-end” solutions
  - Add specialized tools and programs to **Everest Client** forms
- Integrate with external applications
  - Eliminate the need to open and operate separate applications by launching third-party add-in applications within **Everest**
  - Enable and disable the add-ins automatically or allow the end users to make this determination
  - Launch the add-ins from **Everest** menu items or events
  - Retrieve and send data from the add-ins to **Everest** forms
  - Display configurations and information specific to the add-in in the **Everest Add-Ins** utility which is used by **Everest** end users to manage add-ins
- Use proven code developed by the **Everest** Development Team
  - Validates, registers, and executes add-ins to **Everest**

- Helps you concentrate on your requirements
- Safely integrate with **Everest**:
  - No database corruption
  - No impact to the stability of the central system

## Assumptions

- The target audience for **Everest Add-Ins** is Resellers, System Managers, and Developers.
- A good working knowledge of the following is required to understand and use **Everest Add-Ins**:
  - **Everest Client Forms** (documents, profiles, and browsers) in the area related to your specific add-in.
  - Any programming language that supports standards such as COM and XML (Visual Basic .NET, C#, ASP, VB, Delphi™ etc.)
  - Microsoft® Component Object Model (COM)
  - Microsoft® XML (eXtensible Markup Language) Parser
- You must have access to an installation of the **Everest Advanced Edition Server** and **Everest Advanced Edition Client** to test the integration of your Add-In with **Everest**. See the **Everest Advanced Edition Installation Guide** for additional information.

## Document Structure

This developer guide is organized into four chapters as shown in the following table:

Chapter Number	Chapter Name	Description
1	Introduction to <b>Everest Add-Ins</b>	Introduces <b>Everest Add-Ins</b> .
2	Developing Add-Ins for <b>Everest</b>	Provides <b>Quick Start</b> steps to help you get productive quickly and an overview of <b>Everest Add-Ins</b> .  It also contains details of the <b>Add-In Interface</b> and <b>Everest Public Interface</b> along with the associated interface classes and methods.
3	<b>Everest Add-In Schemas</b>	Describes and provides schema samples for clarity and understanding.
4	Managing <b>Everest Add-Ins</b>	Describes how to manage the <b>Everest Add-Ins</b> after they are created. It also details how to use the <b>Add-In Manager</b> utility. You do not have to be a programmer to manage add-ins. Add-ins can be managed by any end user with the necessary security right.

## Document Conventions

The following conventions are used in this developer guide to enable you to navigate easily through the information provided and help you accomplish your tasks quickly:

- Important information is mentioned using the “Note” symbol as seen below:



You will see the icon on the left with the information appearing alongside.

- A “Refer” symbol denotes references to other sections within the document or to other documents of the product documentation suite.



You will see the icon on the left with the text for the cross-reference appearing alongside.

- Examples have been provided to illustrate the information presented. The example is mentioned using the symbol given below:



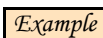
With this convention, you will see the icon on the left with the example appearing alongside.

- Wherever there is a necessity to perform a sequence of tasks for a particular function, the “Steps” icon is used. This is indicated by the symbol given below:



You will see the icon on the left with the text appearing alongside indicating that the steps to carry out a task are listed.

- The keywords related to the software such as menu options, profile/dialog box names and headings are highlighted in **Boldface**.



**Add-In Manager, Everest**

- Words with bold and italic text indicate parameters of the function and are highlighted in ***Boldface + Italic***.



***sControlName, bRecursive***

- Words with blue shading indicate coding.



```
function TUserAddIn.Execute(const oForm:
IEverestForm;
const sCommand, strInstance: WideString):
WordBool;
```

- For HTML rendering of XML schemas and XML examples, words in red indicate markup. Words in bold black indicate content or values.

```
- <address>
- <General>
  <CustomerCode>100100</CustomerCode>
- <AddressInfo>
  <AddressType>5</AddressType>
  <FirstName>Joe</FirstName>
  <LastName>Dimagio</LastName>
```

- Navigation between and through menu options is indicated by the symbol ">".

**Example** To view the **Add-In Manager** utility for a workstation, from the main menu bar you must select **Utility** and then **Add-In Manager**.

This is explained in the document as follows:

From the main menu bar of the appropriate workstation, select **Utility > Add-In Manager**.



## 2. Developing Add-Ins for Everest

---

---

About Everest Add-Ins  
Getting Started  
Quick Start  
Creating an Add-In for Everest  
Step-by-Step Procedures for Developing Everest Add-Ins  
Add-In Interface  
Everest Public Interface  
Creating an Add-In Configuration File  
Add-In Information Dialog Box  
Installing Your Add-Ins  
Add-In Validation  
Add-In Registration  
Add-In Execution  
Everest Client Menu Hooks  
Everest Client Event Hooks





## About Everest Add-Ins

The **Everest Add-Ins Framework** allows you to extend the functionality of the main **Everest** application to better satisfy the requirements of your business environment. This provides you with the following benefits:

- Ability to customize and extend **Everest Client** features
- Ability to seamlessly integrate with custom applications

The **Everest Add-Ins Framework** provides the necessary programmatic access to third-party client extensions, known as **Add-Ins**. In **Everest**, the **Everest Client** is the host application for the add-in. An **Add-In** is basically a supplemental program that extends the functionality of the main application. You can build add-ins using any .COM or .Net development tool including but not limited to Visual Basic, Visual C#, and Visual C++.

The **Everest Add-Ins Framework** is not part of the SDK installation. Because this framework is part of the **Everest** desktop client application, it is installed along with the **Everest Client** components.

Utilize add-ins to the **Everest Client** to do the following:

- Modify menus, menu items, and toolbars
- Add custom property pages
- Write event handler procedures
- Add specialized features

From within the **Everest Client**, you can view a list of the add-ins and enable or disable them as required.

**Everest API** and **Everest Add-Ins** do not provide the same functionality. **Everest API** programs interact with the business logic layer but **Everest Add-Ins** interact with the client interface.

## Getting Started

Use the following **Quick Start** section to get up and running quickly with **Everest Add-Ins**. The remaining sections of the **Everest Add-Ins Developer Guide** provide more detailed information.



For more information, refer to the **Everest\_Add-Ins Help** file located in the `..\Icode\Everest\Client\Help` directory. For example, if you installed **Everest** on the C: drive, it is located in the `C:\Program Files\Icode\Everest\Client\Help` directory.

## Quick Start

Use the following procedure to begin using **Everest Add-Ins** and extend the functionality of the main **Everest** application to better satisfy the requirements of your business environment. A separate installation is not required since the **Everest Client** contains the **Everest Add-Ins Framework**.

 **To begin using Everest Add-Ins, do the following:**

- Create an Add-In for **Everest**:

<b>Create a COM DLL that Conforms to Everest Standards</b>
Add the <b>Everest</b> Reference Libraries to your program: <ul style="list-style-type: none"><li>• Everest</li><li>• Everest Add-In</li></ul>
Implement the <b>IEverestAddIn</b> interface methods.

### Create a COM DLL that Conforms to Everest Standards

Implement this method to display information about your Add-In in the **Add-In Manager** dialog box when the end user clicks the **Options** button:

- **GetOptions**

Make additional calls to the **Everest** form as required using the following object hierarchy:

- **Handle**
- **GetFormData**
- **SetFormData**
- **GetData**
- **QueryAction**
- **ExecuteAction**
- **SetCustomFieldData**
- **Session**
  - **GetData**
  - **SetData**
- **Browser**
  - **GetData**
- **Profile**
  - **AddCustomPage**
  - **GetData**
- **Document**
  - **AddCustomPage**
  - **UpdateDocument**
  - **Line Item**
    - **AddLineItem**
    - **ModifyLineItem**
    - **DeleteLineItem**
  - **Payment**
    - **GetPOSPaymentDetails**
    - **SetPOSPaymentDetails**



[“Creating an Add-In for Everest” on page 19](#)

[“Add-In Interface” on page 58](#)

[“Everest Public Interface” on page 61](#)

- Create an **Everest Add-In Configuration** file. This is a configuration file that provides additional information to **Everest** regarding your Add-In and has an .acf extension. It is used to link particular forms with specific methods at specific attach points or hooks. Use the **Add-In Configuration File Example** as a guide.



[“Creating an Add-In Configuration File” on page 123](#)

[“Add-In Configuration File Example” on page 124](#)

- Use the **Add-In Information** dialog box to locate the toolbars, menus, menu items and events. To access this dialog box, press the **Ctrl + Shift + F11** keys from any add-in supported form (browser, profile, or document).

 [“Add-In Information Dialog Box” on page 125](#)

- Install your Add-In within the **Everest Client**.
  - Place your .dll and .acf files in the Everest Client Addins folder:  
..\**Icode\Everest\Client\Addins**

**Example** C:\Program Files\Icode\Everest\Client\Addins

- If you do not see the “Addins” folder in the Client directory, you must create it.
- If you are migrating to the Microsoft Windows® Vista operating system, you must manually move your **Everest** Add-Ins to the **<ProgramData>\Everest\Addins** folder.

**Example** C:\ProgramData\Everest\Addins

 [“Installing Your Add-Ins” on page 131](#)

- Register your **Add-In Configuration** files.
  - With your Add-In files in the **Addins** folder, open the **Everest Client**. If your **Everest Client** is already open, close it and restart it. This validates and registers your Add-Ins.

 [“Add-In Registration” on page 131](#)

- Enable your Add-In within the **Everest Client** to activate it.

- When starting **Everest**, you will see an Add-In Registration message. Click **Yes** to activate your Add-In. You can also set your Add-In to activate automatically in the Add-In Configuration file.



**Figure 2.1: Add-In Registration Message**

When the **Everest Client** host application is running, your active add-in executes and can send and receive events.

 [“Add-In Execution” on page 132](#)  
[“Everest Add-Ins” on page 16](#)

**Everest Add-In Samples** for different programming languages can be found in the following location:

`..\Icode\Everest\Client\Samples\Addins`


**Example** `C:\Program Files\Icode\Everest\Client\Samples\Addins`

A sample file with its corresponding Add-In Configuration file is available for each development tool.

 [“Working with Sample Add-ins” on page 153](#)

To view Step-by-Step Procedures for Developing Everest Add-Ins in different programming languages, refer to the **Step-by-Step Procedures for Developing Everest Add-Ins** section.

 [“Step-by-Step Procedures for Developing Everest Add-Ins” on page 26](#)

 For more information, refer to the **Everest\_Add-Ins Help** file located in the `..\Icode\Everest\Client\Help` directory. For example, if you installed **Everest** on the C: drive, it is located in the `C:\Program Files\Icode\Everest\Client\Help` directory.

## Everest Add-Ins

An **Add-In** is a supplemental program that extends the functionality of the main **Everest** application. In order to introduce the **Add-In** structure, **Everest** loads libraries from a specific directory at runtime. These libraries add actions to a menu or subscribe to an event on the main application.

An action can do the following:

- Run as a background process
- Update elements of the main application

Actions are executed on subscribed **Everest** events or registered **Add-In** events.

## Overview

The **Everest Client** contains support for add-ins. Each form in **Everest** includes an interface (IEverestForm) that supports add-ins.

Add-Ins can control **Everest** elements with exposed methods of IEverestForm.

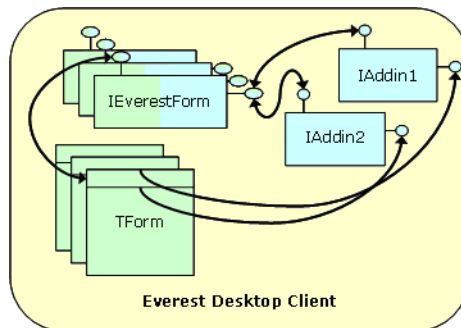


Figure 2.2: Add-In Interaction with Everest

## Add-In Workflow Example

The following figure shows the typical workflow for a registered **Everest Add-In**. When the **Everest** form runs, it reaches the attach point (or launch point) and calls the Create (OnCreate) method. After that, it calls the Execute method which causes the add-in to run. The add-in runs and can then interact with the **Everest** form. The add-in calls a GetData method to the **Everest** form and **Everest** returns data from the form. The add-in calls a SetData method to the **Everest** form and **Everest** returns OK (or another appropriate response) to indicate that the data has been received.

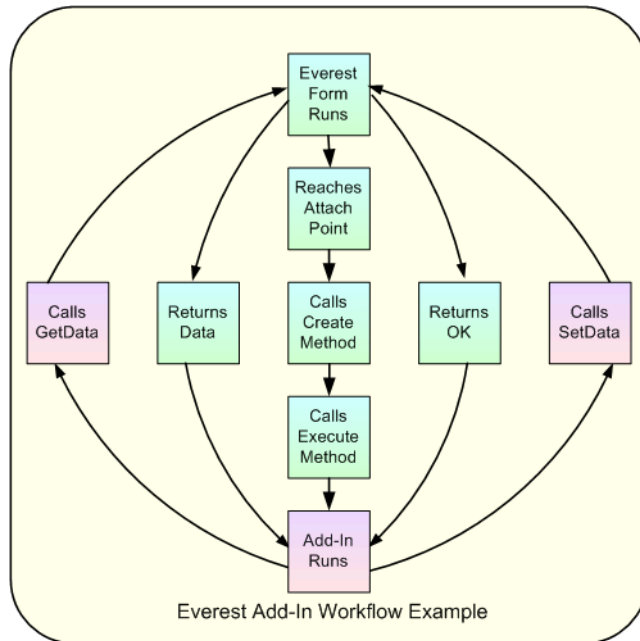


Figure 2.3: Add-In Workflow Example

## Why XML?

Several current technologies exist for sending and receiving data. Some of them include XML strings, Recordset objects, Variant arrays, and Collection objects. The rapid growth of XML usage, together with the industry-wide shift to HTTP and Web services and the efficiency and openness of the method, earn it the choice of best practice.

Advantages of XML
• Disconnected from the database and application server
• Can be persisted to and loaded from files, MSMQ messages, and so on
• Powerful but new navigation semantics
• Easily transformed into different structures with XSL
• Metadata
• When correctly structured, they can be opened as recordsets
• Easily extensible
• Easily integrated with ASP and HTML applications
• Negligible impact on performance when extracted from ADO Recordsets
• Easily incorporated into schema/data validation frameworks
• Easily integrated with XML-based services such as BizTalk

However, with so many technologies coming up each day there is no blanket rule as to what can be the best choice for a particular application. Each technology finds new horizons but on its way binds itself to others, such as DCOM, ADO, XML Parser, etc.



## Creating an Add-In for Everest

In order to create an add-in for **Everest**, you must first create a user-defined DLL that conforms to **Everest** standards. **Everest** will make the following method calls to your Add-In application:

- Execute
- OnCreate
- OnDestroy
- GetOptions

 [“Add-In Interface” on page 58](#)

You must include these methods to use your Add-In with **Everest**. After you add these methods, you can develop your Add-In using this structure. In addition to making your DLL conform to **Everest** standards, you can make the following calls to the **Everest** form using the following object hierarchy:

- Handle - Provides the Window Handle to the **Everest** form
- GetFormData - Gets data from the **Everest** form elements
- SetFormData - Places data in the **Everest** form elements
- GetData - Retrieves the dataset details on any **Everest** form
- QueryAction - Retrieves the status of components available in the **Everest** form
- ExecuteAction - Executes the Click event of a menu item or button available in the **Everest** form or opens any other **Everest** form
- SetCustomFieldData - Sets custom field data
- **Session**
  - GetData - Retrieves session data
  - SetData - Sets up the session details
- **Browser**
  - GetData - Retrieves data from an **Everest** browser
- **Profile**
  - AddCustomPage - Adds your custom forms to any **Everest** profile
  - GetData - Retrieves data from an **Everest** profile
- **Document**
  - AddCustomPage - Adds your custom forms to any **Everest** profile
  - UpdateDocument - Updates documents (sales, purchase, and point of sale)
    - **LineItem**
      - AddLineItem - Adds line items in documents (sales, purchase, and point of sale)
      - ModifyLineItem - Modifies line items in documents (sales, purchase, and point of sale)
      - DeleteLineItem - Removes line items from documents (sales, purchase, and point of sale)
    - **Payment**
      - GetPOSPaymentDetails - Gets the details of POS payments
      - SetPOSPaymentDetails - Places data into the controls on the payment details grid of the **Payment Details** dialog box in **Everest** POS



- Validation on changing the control values may not always happen instantly; it may be handled at the time of saving.
- As a general practice, any new forms that you introduce should be invoked from a custom menu item.

 [“Everest Public Interface” on page 61](#)

Use the following code samples as a guide for the DLL creation.

## Delphi Sample Code

```
function TUserAddIn.Execute(const oForm: IEverestForm;
const sCommand, strInstance: WideString): WordBool;
var
bQueryAction : Boolean;
begin
// Call AddCustomPage form oForm.Profile object
oForm.Profile.AddCustomPage('Item
Image','DelphiUserAddIn.DelphicustomPage');
end;
```

## Visual Basic Sample Code

```
Private Function IEverestAddin_Execute(ByVal oForm As EverestForm,
ByVal sCommand As String) As Boolean
Dim strGetProfileData As String
Dim MyForm As frmGetProfileData
'MsgBox "IEverestAddin_Execute"
IEverestAddin_Execute = True
strGetProfileData = oForm.Profile.GetProfileData
'MsgBox (strGetProfileData)
Set MyForm = New frmGetProfileData
Load MyForm
MyForm.txtOutput.Text = strGetProfileData
MyForm.Show vbModal
Unload MyForm
Set MyForm = Nothing
End Function
```

## Visual Basic .NET Sample Code

```
Public Function Execute(ByVal oForm As Everest.EverestForm, ByVal
sCommand As String, ByVal sInstanceData As String) As Boolean Implements
EverestAddin.IEverestAddin.Execute
Dim strSessionData As String
Dim Sample1 As frmSample
Dim strGetData, strNodeData, sCreditCardNo, strCCNumber As String
Dim strServerName, strDatabaseName, strJurisdiction As String
Dim strDepartment, strCompanyCode, strPassword, strUserCode As String
Dim oResultXML As Xml.XmlDocument
Dim oNode As Xml.XmlNode
MsgBox("VBNetAddin Execute method called.")
*** Calling the GetSessionData method from EverestForm.
strSessionData = oForm.Session.GetData(false)
Try
*** Create an instance of Sample form
Sample1 = New frmSample
*** show Sample form
Sample1.Show()
*** Assing the output XML to output textbox on frmsample
Sample1.txtoutput.Text = strSessionData
*** Create an instance of XML Document
oResultXML = New Xml.XmlDocument
*** Load the session data XML into the XML document calling loadxml()
oResultXML.LoadXml(strSessionData)
*** Get the node for servername from the xml
oNode = oResultXML.SelectSingleNode("SessionData/ServerName")
*** Get the value of the Servername node
strServerName = oNode.InnerText
*** Assign the servername node to textbox.
Sample1.txtServerName.Text = strServerName
*** Get the node for databasename from the xml
oNode = oResultXML.SelectSingleNode("SessionData/DatabaseName")
```

## Visual Basic .NET Sample Code (Continued)

```
**** Get the value of the databasename node
strDatabaseName = oNode.InnerText
**** Assign the databasename node to textbox.
Sample1.txtDBName.Text = strDatabaseName
**** Get the node for usercode from the xml
oNode = oResultXML.SelectSingleNode("SessionData/UserCode")
**** Get the value of the usercode node
strUserCode = oNode.InnerText
**** Assign the usercode node to textbox.
Sample1.txtUserCode.Text = strUserCode
**** Get the node for companycode from the xml
oNode = oResultXML.SelectSingleNode("SessionData/CompanyCode")
**** Get the value of the companycode node
strCompanyCode = oNode.InnerText
**** Assign the companycode node to textbox.
Sample1.txtCompanyCode.Text = strCompanyCode
**** Get the node for department from the xml
oNode = oResultXML.SelectSingleNode("SessionData/Department")
**** Get the value of the department node
strDepartment = oNode.InnerText
**** Assign the departmentcode node to textbox.
Sample1.txtDepartment.Text = strDepartment
**** Get the node for jurisdiction from the xml
oNode = oResultXML.SelectSingleNode("SessionData/Jurisdiction")
**** Get the value of the jurisdiction node
strJurisdiction = oNode.InnerText
**** Assign the jurisdiciton node to textbox.
Sample1.txtJursicode.Text = strJurisdiction
Catch
MsgBox(Err.Description)
End Try
End Function
```

## Visual C++ Sample Code

```
STDMETHOD(Execute)(IEverestForm * oForm, BSTR sCommand,
VARIANT_BOOL * bResult)
{
    BSTR sData;
    _bstr_t strCommand;
    _bstr_t strData;
    strCommand = sCommand;
    MessageBox(NULL, strCommand, "MyVCAddin_Execute", MB_OK);
    oForm->GetFormData(&sData);
    strData = sData;
    MessageBox(NULL, strData, "VCADDIN GetFormData", MB_OK);
    // if (bResult == NULL)
    // return E_POINTER;
    return TRUE;
    // return E_NOTIMPL;
}
```

## C# Sample Code

```
public bool Execute(Everest.EverestForm oForm, string sCommand, string
strInstance)
{
    // TODO: Add Sample.Execute implementation
    //MessageBox.Show("VC.Net Addin Execute method","C# Addin");
    string strFormData = oForm.GetFormData();
    try
    {
        // Create an instance of form
        frmSample sample1 = new frmSample();
        // Show the form
        sample1.Show();
        // Set the form data to output textarea on the form
        sample1.txtOutput.Text = strFormData;
        // Create a new instance of xml document
        XmlDocument oResultXML = new XmlDocument();
        // Load the form data XML into the XML document callingloadxml()
        oResultXML.LoadXml(strFormData);
        /*** Select the node to get the value of Amount.
        string strCode = oResultXML.SelectSingleNode("//
        Object[@Name='TMaskEdit']/Property[@Name='WindowText']/
        @Value").Value;
        // Set the account code to the textbox on form.
        sample1.txtName.Text = strCode;
    }
    catch(NullReferenceException ex)
    {
        MessageBox.Show(ex.Message);
    }
    return true;
}
```

Everest Add-In Samples for different programming languages can be found in the following location:

..\Icode\Everest\Client\Samples\Addins

**Example** C:\Program Files\Icode\Everest\Client\Samples\Addins

A sample file with its corresponding Add-In Configuration file is available for each development tool.



[“Working with Sample Add-ins” on page 153](#)

## Step-by-Step Procedures for Developing Everest Add-Ins

The following examples show how to develop **Everest Add-Ins** using three different programming languages:

- Visual Studio ® 2005 (C#)<sup>1</sup>
- Microsoft ® Visual Basic ® 6.0
- Borland ® Delphi <sup>TM</sup> 7<sup>2</sup>

**Everest COM Add-Ins** presents the developer with a way to extend the functionality of the **Everest** application for custom tasks. **Everest COM Add-Ins** are typically used to automate **Everest** in response to a click of a Menu Item, button, or some other event specific to Everest such as opening or closing profiles and documents or entering data in to the fields.

An **Everest COM Add-In** is an in-process COM server (an ActiveX DLL) that must inherit from the **IEverestAddIn** interface and makes use of the **Everest** Reference Libraries. All COM Add-ins must implement each of the four methods of the **IEverestAddIn** interface: **OnCreate**, **GetOptions**, **Execute**, and **OnDestroy**.

When a COM Add-in is installed on a user's system, registry entries are created for the Add-in.



["Introduction to Everest Add-Ins" on page 3](#)

["About Everest Add-Ins" on page 11](#)

["Add-In Interface" on page 58](#)

## Prerequisites

The Add-In developer should have considerable knowledge of **Everest** functionality, **COM interop** and **XML technology**.



["Assumptions" on page 5](#)

1.Windows, Microsoft, Visual Basic, Visual Studio, and MSDN are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.  
2.Borland, the Borland Logo, and Delphi are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries.



## Creating Everest Add-Ins Using Visual Studio 2005 (C#)

Use the following procedures to create an **Everest Add-In** using Visual Studio 2005 (C#):


- Register the **Everest** Reference Libraries
  - Register the EverestAddin Library and **Everest** Library Components on your machine
  - Create a Project and Add the **Everest** Reference Libraries
- Implement the IEverestAddin Interface Methods
- Make Additional Calls to the **Everest Form** as Required
- Create the Add-In Configuration File
- Install, Register, and Enable your Add-In

 ["Quick Start" on page 12](#)

### Register the Everest Reference Libraries (Visual Studio 2005 - C#)

Before you create your **Everest Add-In**, make sure that the **EverestAddin Library** and **Everest Library** components are registered on your machine.

## Create a Project and Add the Everest Reference Libraries (Visual Studio 2005 - C#)

 To create a new project and add the Everest Reference Libraries, do the following:

- From the Microsoft ® Development Environment main menu bar, select **File > New > Project**.

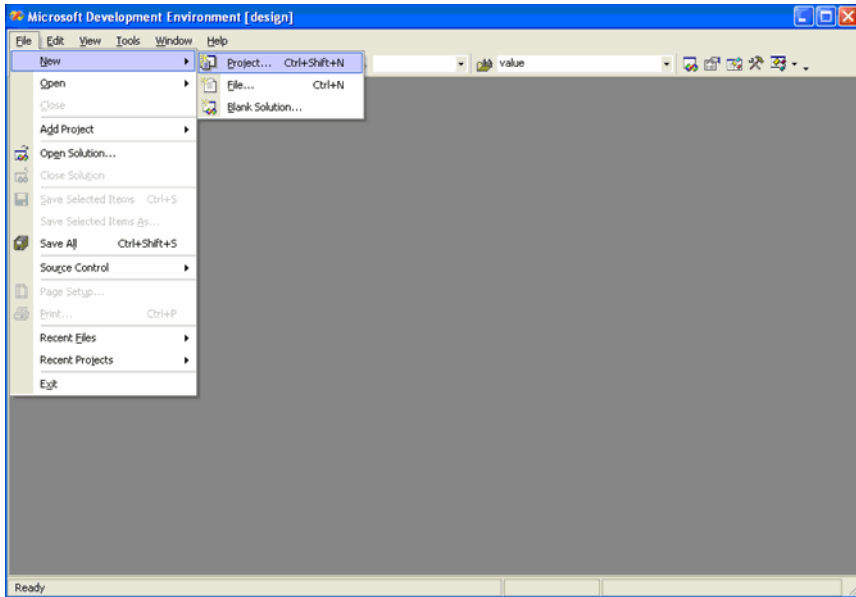
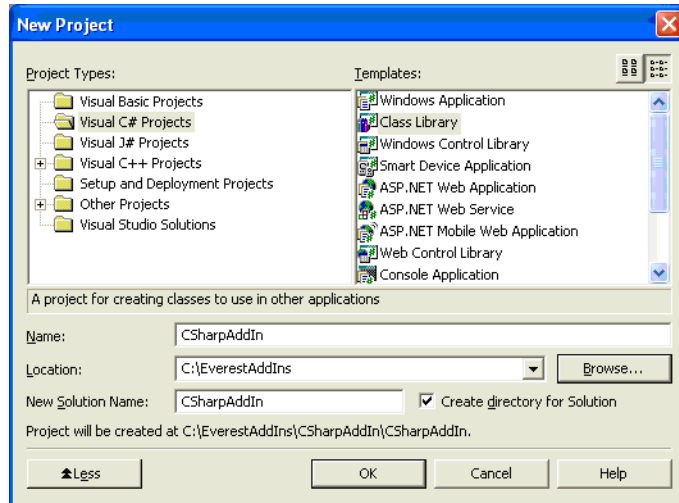


Figure 2.4: Path to New Project

The **New Project** dialog box appears.



**Figure 2.5: New Project Dialog Box**

- In the **New Project** dialog box, select:  
Project Types: **Visual C# Projects**  
Templates: **Class Library**
- Give a proper name for **Name**, **Location**, and **New Solution Name**.
- Click **OK**. The project code editor opens for the design.

- If the Solution Explorer window does not open, select **View > Solution Explorer** as shown below:

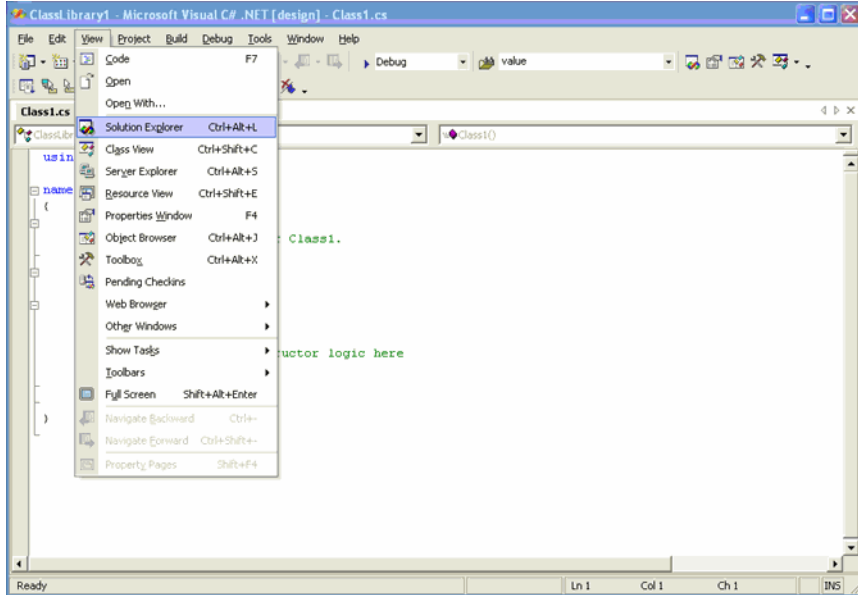


Figure 2.6: Path to Solution Explorer

The Solution Explorer is displayed.

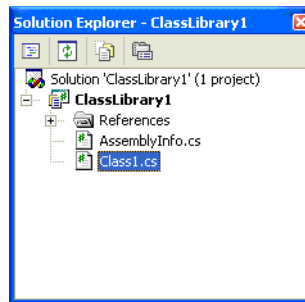


Figure 2.7: Solution Explorer

- Right-click the project name and select **Properties**.

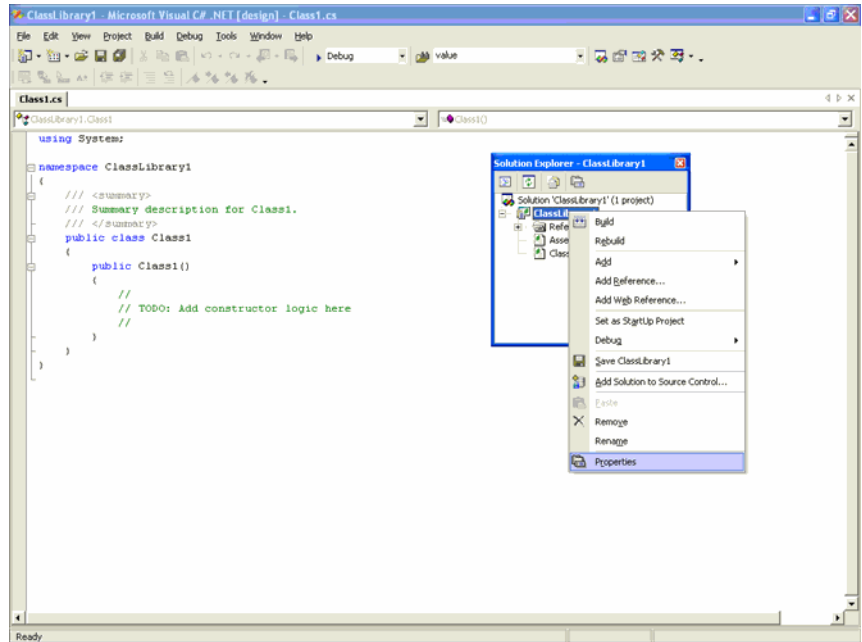
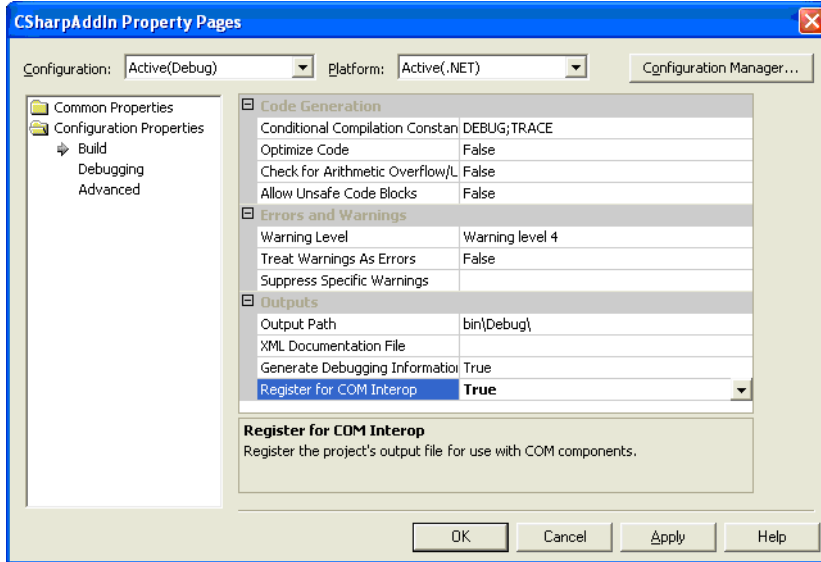


Figure 2.8: Solution Explorer - Properties Selected

The Project properties dialog box is displayed.

- In the left pane, select **Configuration Properties > Build**. It displays all the Build properties on the right side pane.



**Figure 2.9: Project Properties Dialog Box**

- In the right pane, select **Outputs > Register for COM Interop** and set the value to **True**.
- Click **OK**.

- In the **Solution Explorer** dialog box, right-click the **References** option under CSharpAddIn and select **Add Reference** from the options.

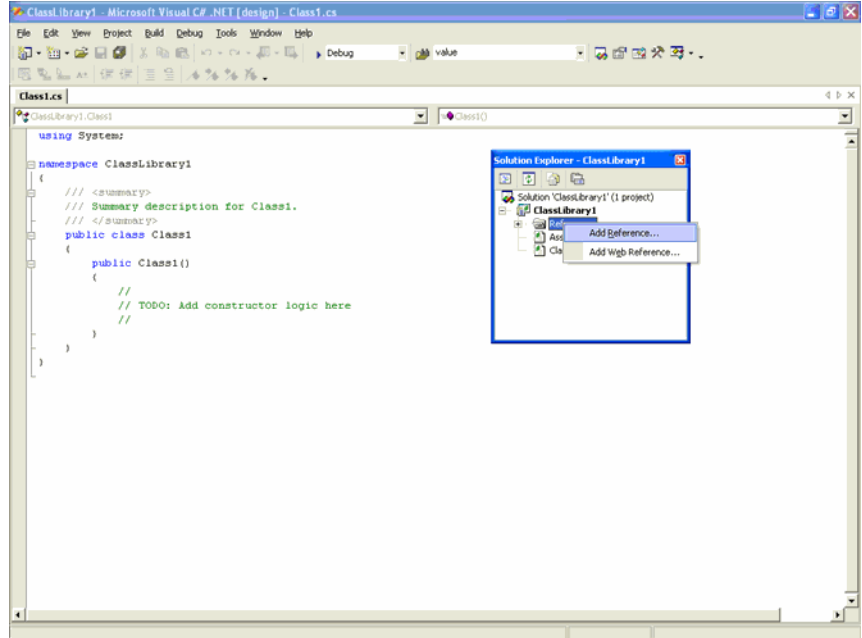


Figure 2.10: Solution Explorer - Add Reference Selected

The **Add Reference** dialog box is displayed.

- Select the COM tab sheet. It displays all the registered COM Interop Libraries.

- Highlight **Everest Library** and click the **Select** button. It gets added to the selected components list.

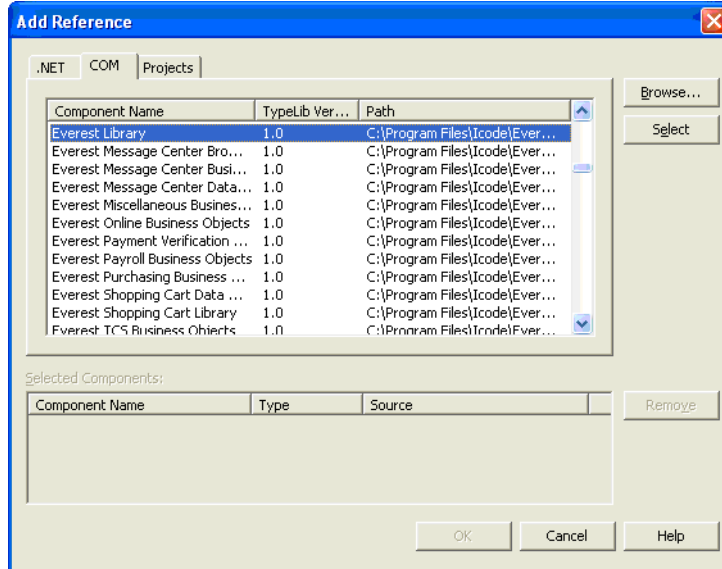


Figure 2.11: Add Reference Dialog Box - Everest Library Highlighted



- Similarly, select **EverestAddin Library** and click the **Select** button. The **Selected Components** section below displays the two selected **Everest** library components.

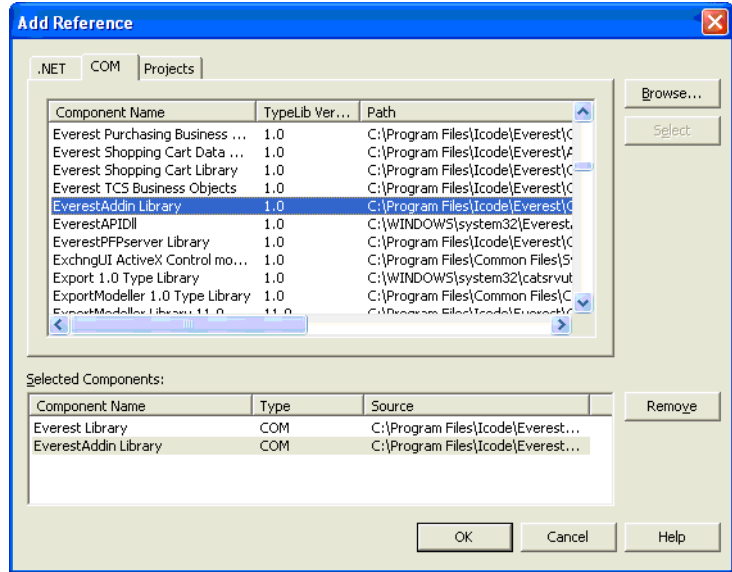


Figure 2.12: Add Reference Dialog Box - Everest Library and EverestAddin Library Selected

- Click **OK**. The **Reference** option in the Solution Explorer window contains all the references.

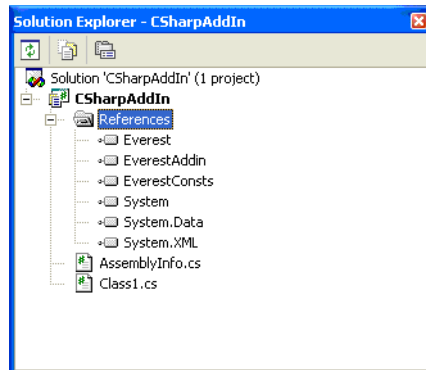


Figure 2.13: Solution Explorer Showing References

## Implement the IEverestAddin Interface Methods (Visual Studio 2005 - C#)

Use the Class “Class1” from the project you just created to implement the IEverestAddin interface methods.

 To implement the IEverestAddin interface, do the following:

- Now extend the Class “Class1” by inheriting from EverestAddin.IEverestAddIn as shown below.

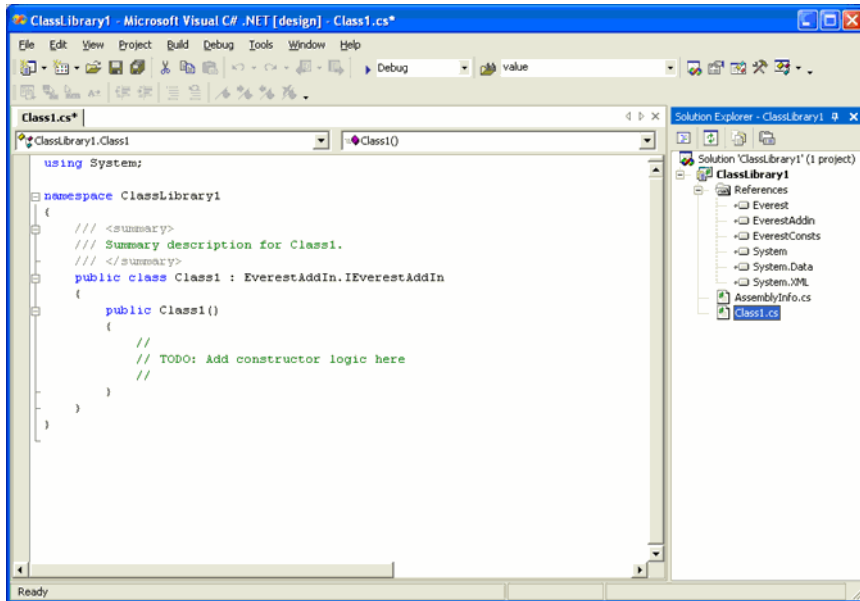


Figure 2.14: Extend the Class by Inheriting from EverestAddin.IEverestAddin

- Add the four required methods (OnCreate, OnDestroy, Execute, and GetOptions) under the class declaration.

```

public void OnCreate()
{
    // TODO: Add OnCreate implementation
}
public void OnDestroy()
{
    // TODO: Add OnDestroy implementation
}

public bool Execute(Everest.EverestForm oForm, string
    SCommand, string strInstance)
{
    // TODO: Add Execute implementation
}

public void GetOptions()
{
    // TODO: Add GetOptions implementation
}

```

Figure 2.15: Code for the Four Required Methods

The class declaration should look like the following example.

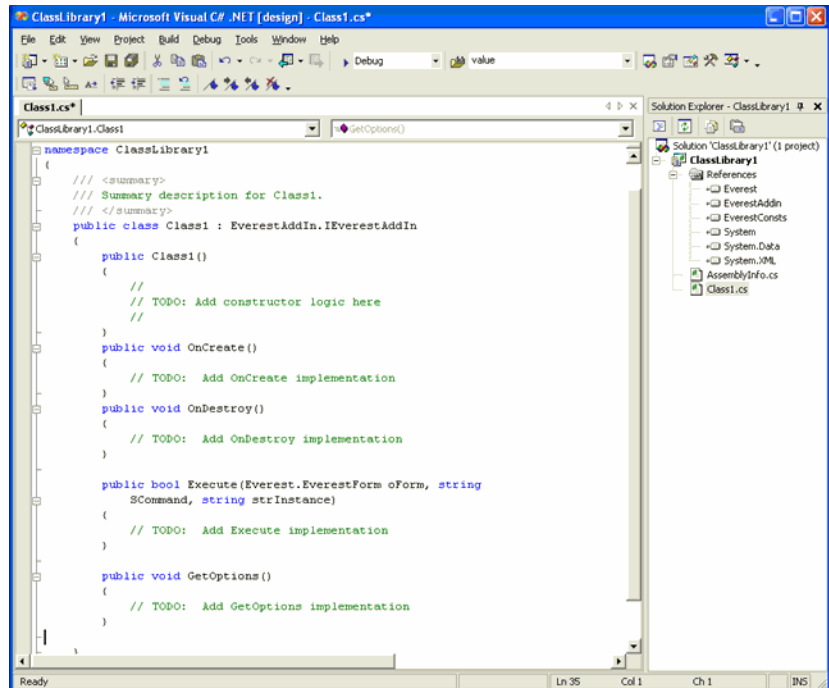


Figure 2.16: Class Declaration Code Example

- Implement the Execute, GetOptions, OnCreate, OnDestroy methods to suit your Add-In needs.



All the Interop DLLs created by Microsoft® Visual Studio must remain in the same folder. Do not delete them.



[“Add-In Interface” on page 58](#)

#### Parameters for the Execute method:

##### *oForm*

It is an `IEverestForm` interface object that contains methods which can be used to operate on **Everest**.

##### *sCommand*

It is an identifier which can be used effectively to change the Add-In behavior depending on `sCommand`, which will be specified in the ACF file.

##### **sInstanceData**

It contains event related data.

- Now build the COM Add-In and register it.

The objectID or ProgID of the COM Add-In will be type **library name.Object name**.

- Register your Addin Dll using the “regasm” command specifying the “codebase” parameter.



[“IEverestAddin.Execute” on page 58](#)

## Make Additional Calls to the Everest Form as Required

You can make additional calls to the **Everest** form as required by your Add-In.



[“Creating an Add-In for Everest” on page 19](#)

[“Everest Public Interface” on page 61](#)

## Create the Add-In Configuration File

Create an **ACF (Add-In Configuration File)** file to specify all the hooks for your Add-In.

 [“Creating an Add-In Configuration File” on page 123](#)

Use the **Add-In Information** dialog box [Path: In **Everest**, open the form (browser, document, or profile) and press the **Ctrl + Shift + F11** keys.] to quickly locate a list of all the menus, toolbars, actions, and events that can be used with add-ins for a specific form.

 [“Add-In Information Dialog Box” on page 125](#)

## Install, Register, and Enable your Add-In

To use your Add-In in **Everest**, it you must install, register, and enable it.

 [“Quick Start” on page 12](#)

## Creating Everest Add-Ins Using Visual Basic 6.0

Use the following procedures to create an **Everest Add-In** using Microsoft® Visual Basic® 6.0:

- Register the **Everest** Reference Libraries
  - Register the **EverestAddin Library** and **Everest Library** Components on your machine
  - Create a Project and Add the **Everest** Reference Libraries
- Implement the **IEverestAddin** Interface Methods
- Make Additional Calls to the **Everest Form** as Required
- Create the Add-In Configuration File
- Install, Register, and Enable your Add-In

 ["Quick Start" on page 12](#)

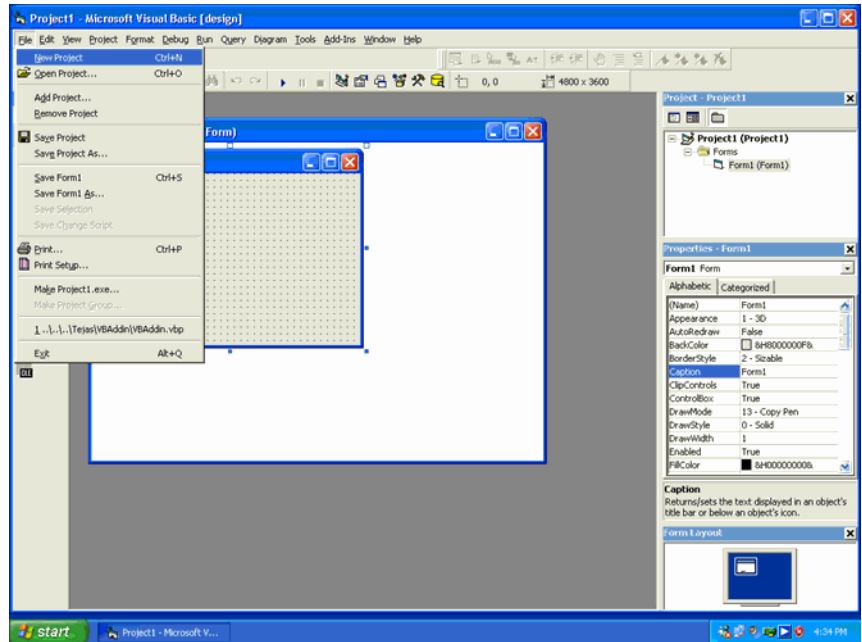
### Register the Everest Reference Libraries (Visual Basic 6.0)

Before you create your **Everest Add-In**, make sure that the **EverestAddin Library** and **Everest Library** components are registered on your machine.

## Create a Project and Add the Everest Reference Libraries (VB 6.0)

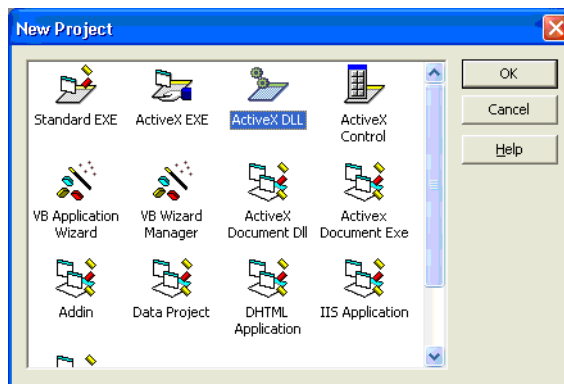
**1** **2** **3** → To create a new project and add the Everest Reference Libraries, do the following:

- From the Microsoft® Visual Basic® 6.0 main menu bar, select **File > New Project**.



**Figure 2.17: Path to New Project Dialog Box**

The New Project dialog box appears.



**Figure 2.18: New Project Dialog Box**

- Select **ActiveX DLL** from the project types.
- Click **OK** to display the code editor.

- From the code editor, select **Project > References**.

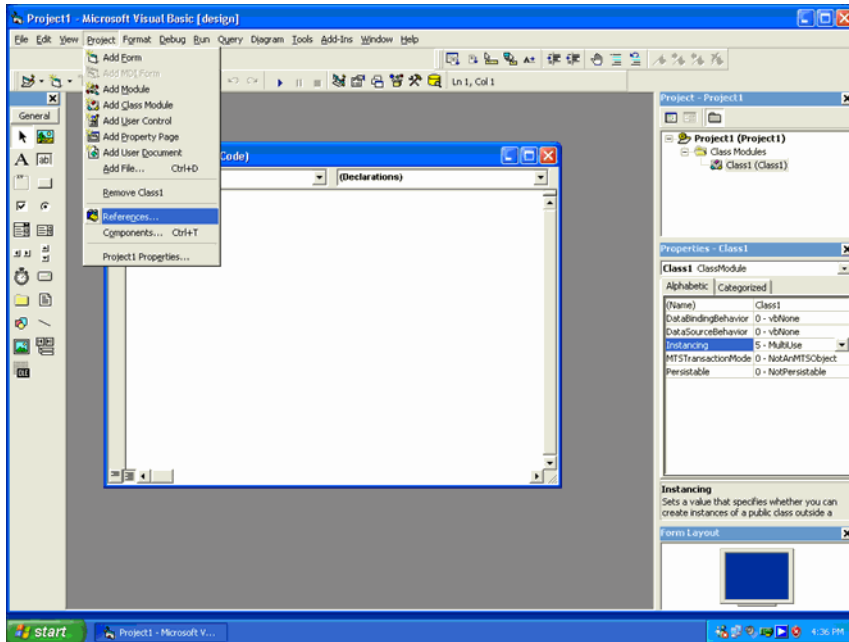


Figure 2.19: Code Editor with Project References Selected

The **References** dialog box is displayed.

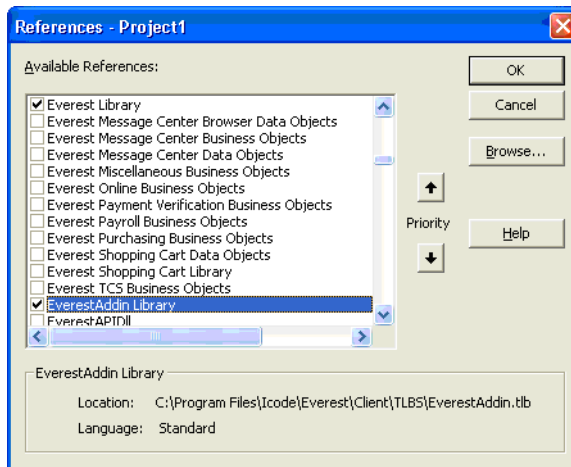


Figure 2.20: References Dialog Box



- In the References dialog box, select **Everest Library** and **EverestAddinLibrary** from the available list as shown above.
- Click **OK** to return to the code editor.

## Implement the **IEverestAddin** Interface Methods (Visual Basic 6.0)

Use the code editor in the project you just created to implement the **IEverestAddin** interface method.

 **To implement the **IEverestAddin** interface, do the following:**

- In the project's code editor, enter the following line:  
**Implements EverestAddin, IEverestAddin**  
As soon as you add the line, the **IEverestAddin** entry adds onto the **General** combo box.
- Click the **IEverestAddin** entry.

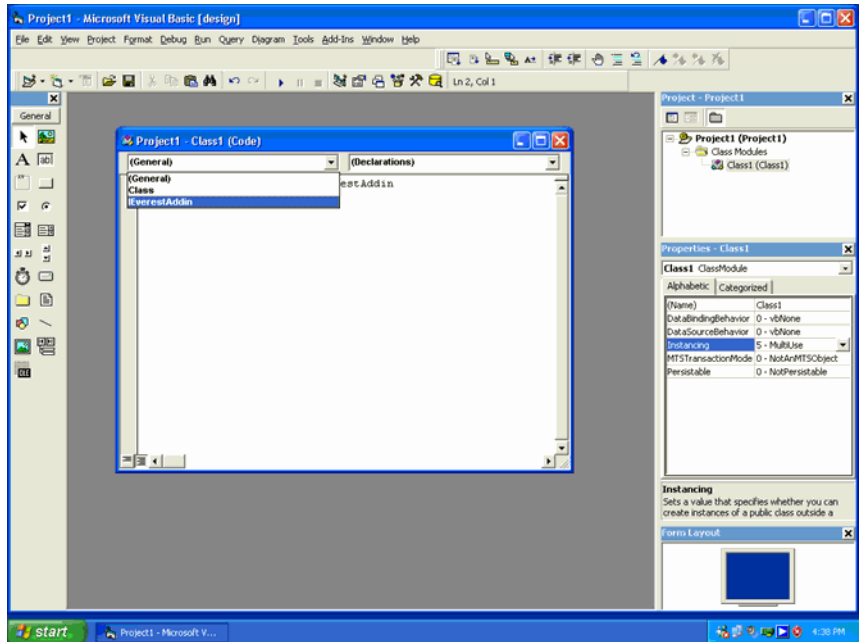


Figure 2.21: Code Editor Showing **IEverestAddin** Selected

The right side combo box gets all the methods of the IEverestAddIn interface as shown below.

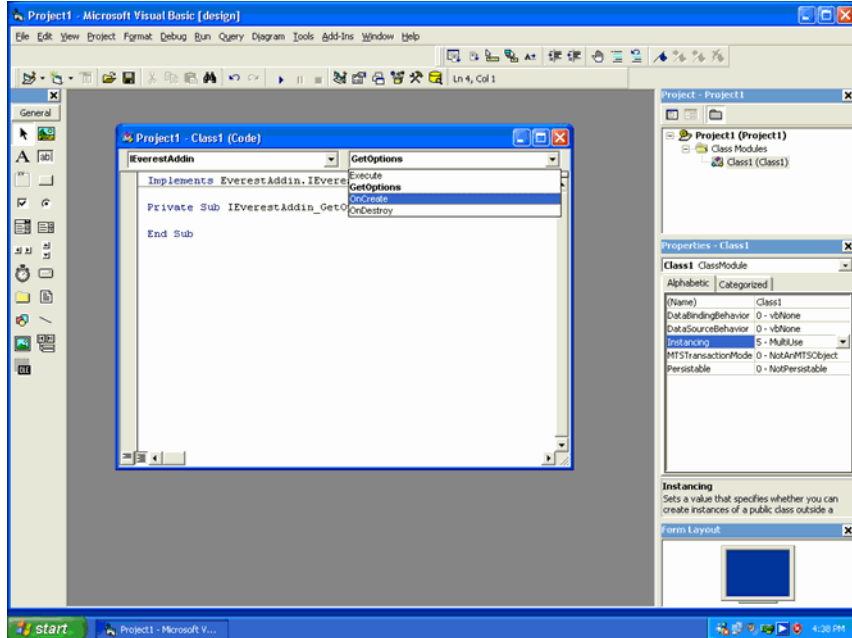


Figure 2.22: Code Editor Showing IEverestAddin Interface Methods

- Click the method name in the combo box to add the definition to the code editor. Add all method definitions.
- Implement the Execute, GetOptions, OnCreate, OnDestroy methods to suit your Add-In needs.

 ["Add-In Interface" on page 58](#)

#### Parameters for the Execute method:

##### *oForm*

It is an IEverestForm interface object that contains methods which can be used to operate on Everest.

##### *sCommand*

It is an identifier which can be used effectively to change the Add-In behavior depending on sCommand, which will be specified in the ACF file.

##### *sInstanceData*

It contains event related data.

- Now build the COM Add-In and register it.

The objectID or ProgID of the COM Add-In will be **library name.Object name**.

- Register the above created Addin Dll using the “regsvr32” command.

 [“IEverestAddin.Execute” on page 58](#)

## Make Additional Calls to the Everest Form as Required

You can make additional calls to the **Everest** form as required by your Add-In.

 [“Creating an Add-In for Everest” on page 19](#)

[“Everest Public Interface” on page 61](#)

## Create the Add-In Configuration File

Create an **ACF (Add-In Configuration File)** file to specify all the hooks for your Add-In.

 [“Creating an Add-In Configuration File” on page 123](#)

Use the **Add-In Information** dialog box [Path: In **Everest**, open the form (browser, document, or profile) and press the **Ctrl + Shift + F11** keys.] to quickly locate a list of all the menus, toolbars, actions, and events that can be used with add-ins for a specific form.

 [“Add-In Information Dialog Box” on page 125](#)

## Install, Register, and Enable your Add-In

To use your Add-In in **Everest**, it you must install, register, and enable it.

 [“Quick Start” on page 12](#)

## Creating Everest Add-Ins Using Delphi 7

Use the following procedures to create an **Everest Add-In** using Borland ® Delphi™ 7:

- Register the **Everest** Reference Libraries
  - Register EverestAddIn\_TLB and Everest\_TLB on your machine
  - Create a New ActiveX Library
  - Create an Automation Object
- Implement the IEverestAddin Interface Methods
- Make Additional Calls to the **Everest Form** as Required
- Create the Add-In Configuration File
- Install, Register, and Enable your Add-In

 [“Quick Start” on page 12](#)

### Register the Everest Reference Libraries (Delphi 7)

Before you create your **Everest Add-In**, make sure that **EverestAddIn\_TLB** and **Everest\_TLB** type libraries are registered on your machine.

## Create a New Active X Library (Delphi 7)

**1** **2** **3** → To create a new Active X library, do the following:

- From the Delphi™ 7 main menu bar, select **File > New > Other**.

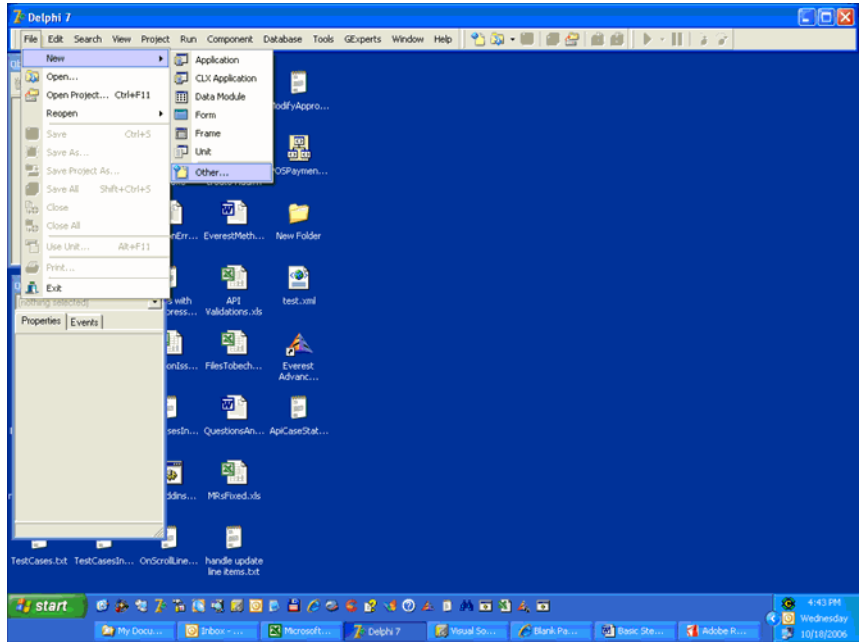


Figure 2.23: Path to New Items

- On the **ActiveX** tab of the **New Items** dialog box, double-click **ActiveX Library**.

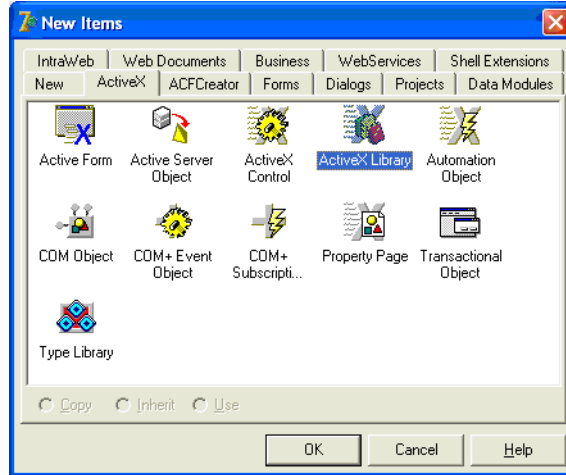


Figure 2.24: New Items - ActiveX Tab (ActiveX Library Selected)

- Save the project with some proper name.

## Create an Automation Object (Delphi 7)

Keep the project open and select the **Automation** object as shown below.

 **To create an Automation Object, do the following:**

- With the project open, select **File > New > Other**. The **New Items** dialog box is displayed.

- On the **ActiveX** tab, double-click the **Automation Object**.

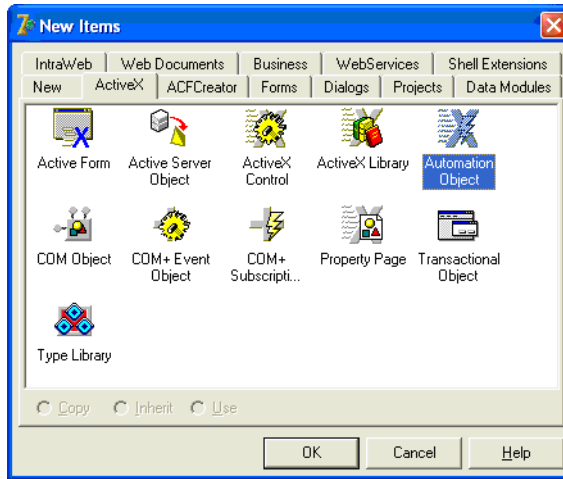


Figure 2.25: New Items - ActiveX Tab (Automation Object Selected)

The **Automation Object Wizard** dialog box pops up.

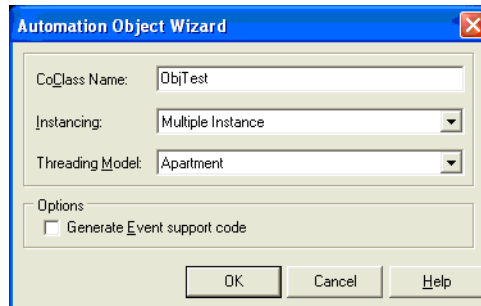


Figure 2.26: Automation Object Wizard Dialog Box

- In the **CoClass Name** field, enter a meaningful name which acts as the name of the COM Add-In object. You may modify the other options such as **Instancing** and **Threading Model** based on the Add-In needs. The default options in the dialog box provide all typical functionality.

- Click OK. The **Type Library** dialog box appears.

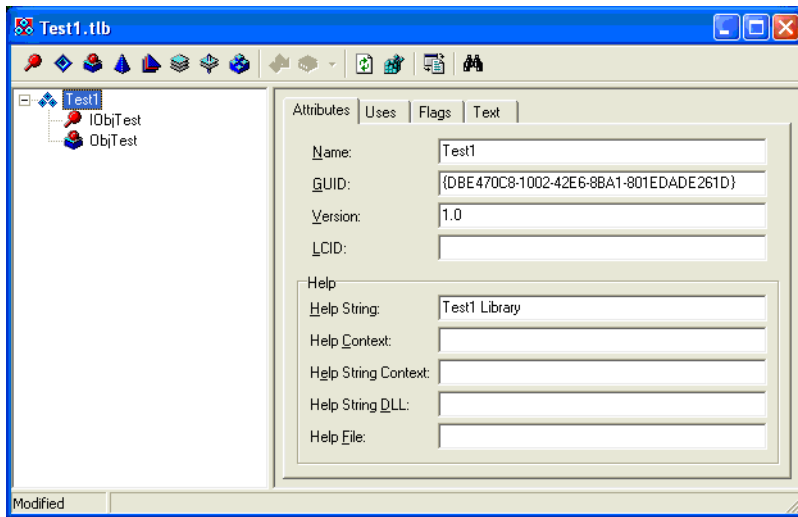


Figure 2.27: Type Library Dialog Box

- **Select the Type Library** in the tree view as shown in the above dialog box.
- In the right pane, click the **Uses** tab sheet. You can see all the type libraries that are used by this type library.



- Right click on the free space and select **Show All Type Libraries**.

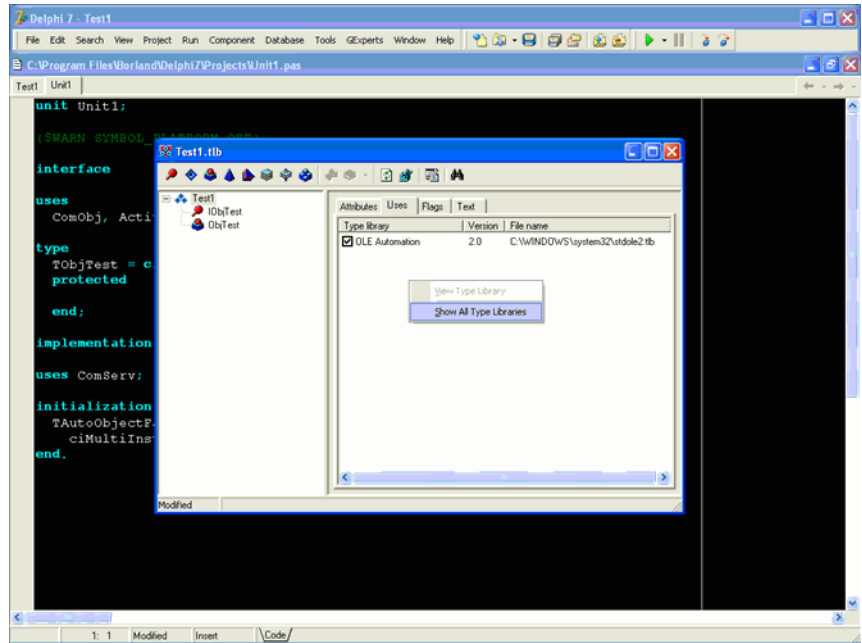


Figure 2.28: Type Library - Uses Tab with “Show All Type Libraries” Option

This lists all type libraries registered on the machine.

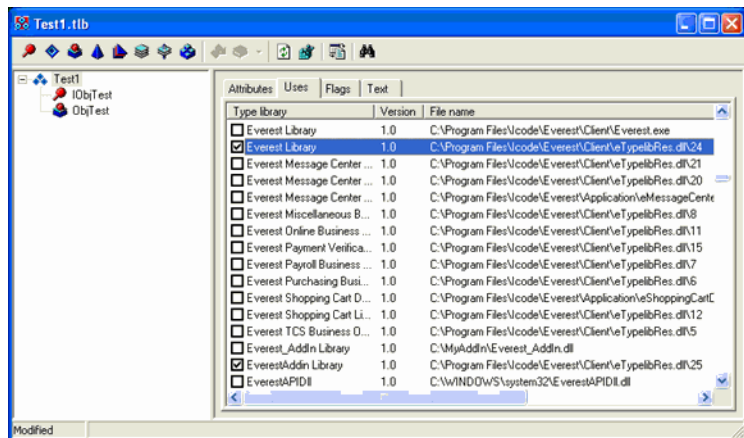



Figure 2.29: Type Library - Uses Tab Showing All Type Libraries Registered on the Machine

- Select and check **Everest Library** and **Everest Addin Library** from the list.
-  These libraries will appear only if you had already registered them on the machine.
- To view only the selected type libraries, right click anywhere on the list and select **Show Selected**.

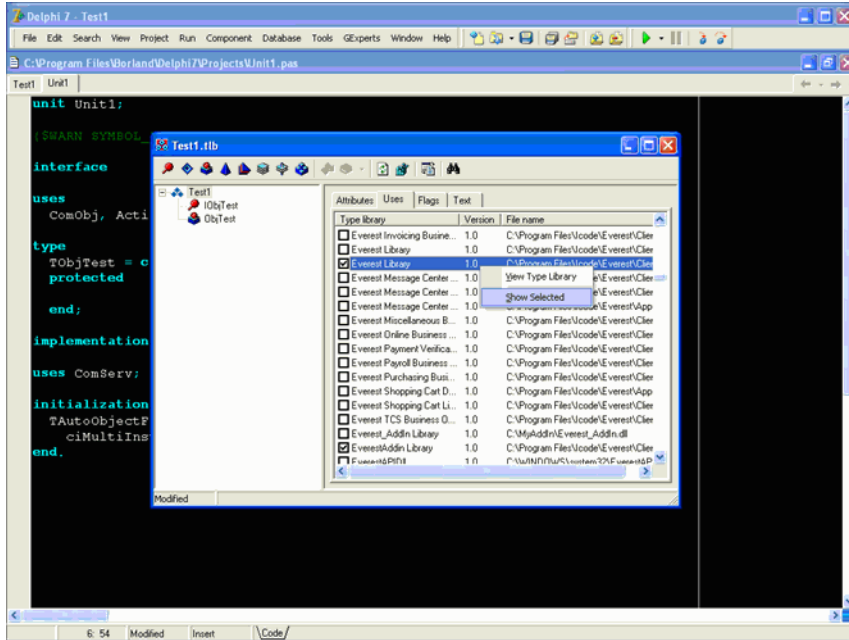


Figure 2.30: Type Library - Uses Tab with “Show Selected” Option

All of the selected type libraries appear in the list. You can choose any number of type libraries for your Add-In but you must select the **Everest** and **Everest Addin** type libraries.

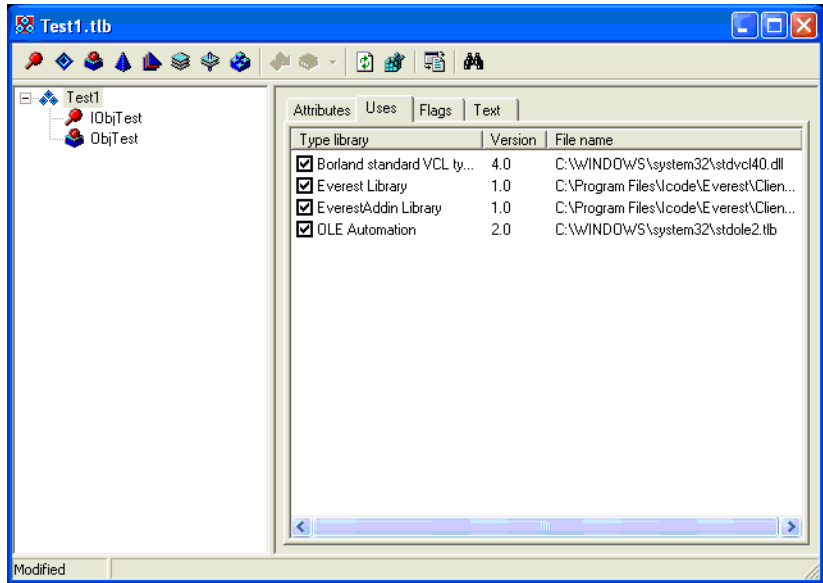


Figure 2.31: Type Library - Uses Tab Showing All Selected Type Libraries

## Implement the IEverestAddin Interface Methods (Delphi 7)

Use the Automation Object you just created to implement the IEverestAddin interface methods.

**1** **2** **3** → To implement the IEverestAddin interface, do the following:

- In the **Type Library** dialog box tree view, select your interface (IObjTest in this example).

- Change the parent interface to IEverestAddIn as shown below.

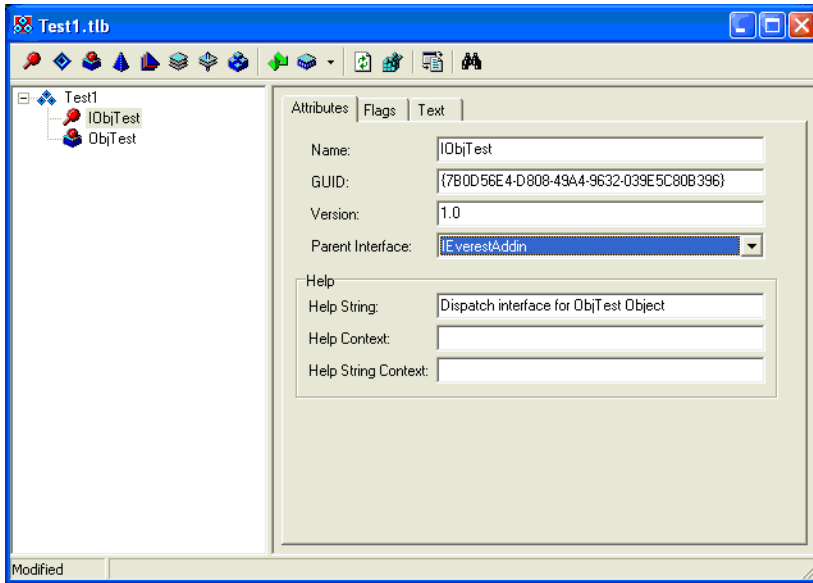


Figure 2.32: Type Library Dialog Box with IEverestAddIn as Parent Interface

- Click the **Refresh** button provided on the type library editor. The **Implementation File Update Wizard** dialog box appears.

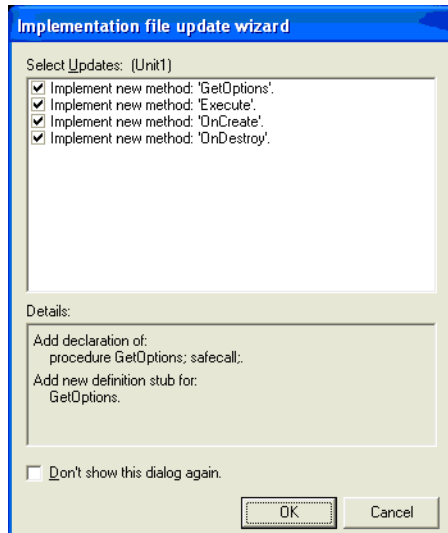
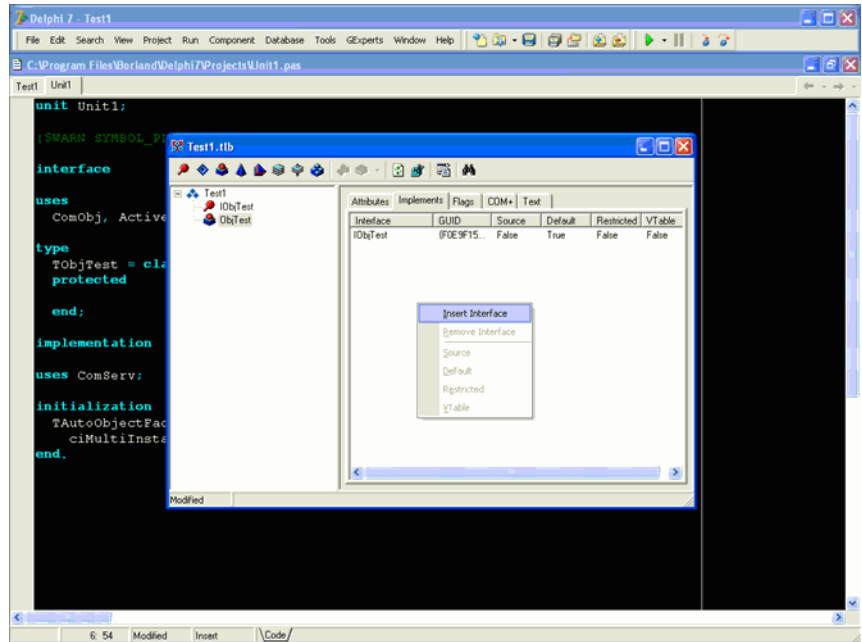


Figure 2.33: Implementation File Update Wizard Dialog Box

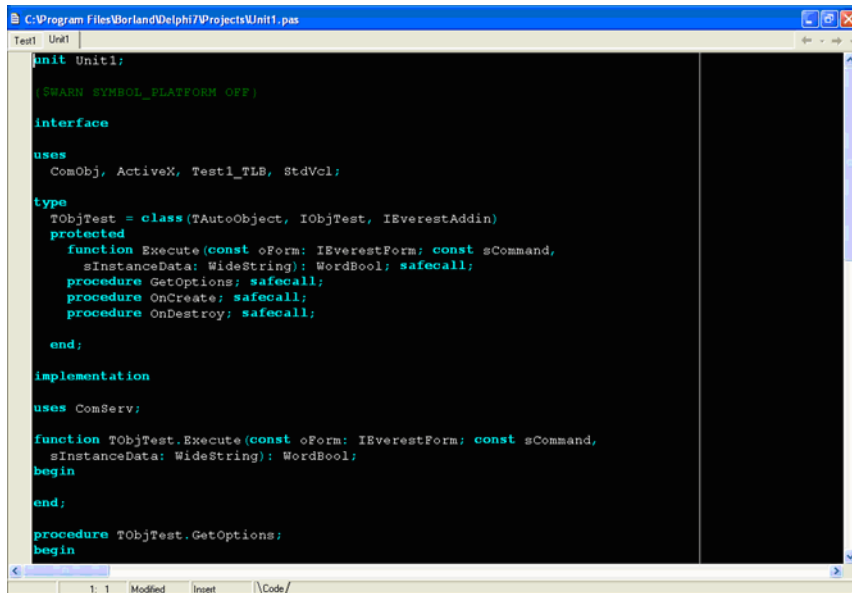
- Click **OK**. This adds all the required code to the implementation unit.

- Select the COM object and select the **Implements** tab as shown below. Right click on the free space and select **Insert Interface**. Choose the **IEverestAddIn** interface.



**Figure 2.34: COM Object Implements Tab with “Insert Interface” Option**

Make sure that Everest\_TLB, EverestAddIn\_TLB got added to the uses class of the implementation unit, and the CoClass implements IEverestAddIn.



```

Unit Unit1;
{$WARN SYMBOL_PLATFORM OFF}

interface

uses
  ComObj, ActiveX, Test1_TLB, StdVcl;

type
  TObjTest = class(TAutoObject, IObjTest, IEverestAddin)
  protected
    function Execute(const oForm: IEverestForm; const sCommand,
      sInstanceData: WideString): WordBool; safecall;
    procedure GetOptions; safecall;
    procedure OnCreate; safecall;
    procedure OnDestroy; safecall;
  end;

implementation

uses ComServ;

function TObjTest.Execute(const oForm: IEverestForm; const sCommand,
  sInstanceData: WideString): WordBool;
begin
end;

procedure TObjTest.GetOptions;
begin
end;

```

Figure 2.35: Code Showing Everest\_TLB and EverestAddIn\_TLB Added

- Start implementing the Execute, GetOptions, OnCreate, OnDestroy methods to suit the Add-In needs.

 [“Add-In Interface” on page 58](#)

#### Parameters for the Execute Method:

##### *oForm*

It is an IEverestForm interface object that contains methods which can be used to operate on Everest.

##### *sCommand*

It is an identifier which can be used effectively to change the Add-In behavior depending on sCommand, which will be specified in the ACF file.

##### *sInstanceData*

It contains event related data.

- Now build the COM Add-In and register it.

The objectID or ProgID of the COM Add-In will be **type library name.Object name**.

In the above figure it is “Test1.ObjTest”.

 [“IEverestAddin.Execute” on page 58](#)

## Make Additional Calls to the Everest Form as Required

You can make additional calls to the **Everest** form as required by your Add-In.

 [“Creating an Add-In for Everest” on page 19](#)  
[“Everest Public Interface” on page 61](#)

## Create the Add-In Configuration File

Create an ACF (**Add-In Configuration File**) file to specify all the hooks for your Add-In.

 [“Creating an Add-In Configuration File” on page 123](#)

Use the **Add-In Information** dialog box [Path: In **Everest**, open the form (browser, document, or profile) and press the **Ctrl + Shift + F11** keys.] to quickly locate a list of all the menus, toolbars, actions, and events that can be used with add-ins for a specific form.

 [“Add-In Information Dialog Box” on page 125](#)

## Install, Register, and Enable your Add-In

To use your Add-In in **Everest**, it you must install, register, and enable it.

 [“Quick Start” on page 12](#)

## Add-In Interface

Use the following interface class to make your DLL conform to **Everest** standards. You must include these required methods to use your Add-In with **Everest**. After you add these methods, you can develop your Add-In using this structure.

Class	Description
IEverestAddin	Implement this interface in your DLL.

### IEverestAddin

IEverestAddin is a middle tier interface that contains the following methods:

Method	Description
IEverestAddin.Execute	This method should be implemented to execute some action at the attached menu or event hook.
IEverestAddin.OnCreate	This method initializes your Add-In.
IEverestAddin.OnDestroy	This method frees the add-in instance and writes to the error handler.
IEverestAddin.GetOptions	This method gets information that is accessed from the <b>Options</b> button in the <b>Add-In Manager</b> utility. Examples include version information and a description of the add-in.

### IEverestAddin.Execute

#### Purpose

This method should be implemented to execute some action at the attached menu or event hook.

#### Syntax

```
procedure Execute(const oForm: IEverestAddinForm; Command: String;
sInstanceData : WideString) : Boolean;
```

#### Parameters

##### *EverestForm (oForm)*

This is a reference to the public interface of the Add-In's parent **Everest** form. See the **Everest Public Interface** section for the methods to use with this parameter. With oForm, you are free to access the methods that oForm provides such as



GetFormData, SetFormData, etc., which allow interacting with data from the Everest Form.

**Example**

If an end user activates a click event from the Customer profile, the end user gets available data from the Customer profile.



[“Everest Public Interface” on page 61](#)

See the EverestForm section for the complete list of EverestForm methods.



[“Everest Form” on page 62](#)

**Command**

Everest passes your command that leads to this execution from the Add-In Configuration file.

**InstanceData**

Where applicable, this parameter holds event data based on the event.

**Boolean**

You must return true, or false, or throw an exception.

**Return Value**

If the return value is False and the event supports “Can stop further processing”, then Everest stops processing any further code for the respective event. In all other situations, the return value must be true.



[“XML Samples” on page 154](#)

[“Everest Add-In Samples” on page 153](#)

**IEverestAddin.OnCreate****Purpose**

This method initializes your Add-In.

**Syntax**

Create

**Parameters**

None

**Return Value**

None



[“XML Samples” on page 154](#)

[“Everest Add-In Samples” on page 153](#)

## IEverestAddin.OnDestroy

### Purpose

This method frees the add-in instance and writes to the error handler.

### Syntax

Destroy

### Parameters

None

### Return Value

None



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## IEverestAddin.GetOptions

### Purpose

This method gets information that is accessed from the **Options** button in the **Add-In Manager** utility. Examples include version information and a description of the add-in.

### Syntax

GetOptions

### Parameters

None

### Return Value

None



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## Everest Public Interface

Use these interface classes to exchange data to and from the **Everest** forms. The methods in this class are called from the Everest.EverestForm (oForm) parameter of the IEverestAddin Execute method.

Class	Description
EverestForm	Use this interface to manage the data exchange between <b>Everest</b> forms and user-defined DLLs.
EverestSession	Use this interface to manage the data exchange specific to <b>Everest</b> sessions.
EverestBrowser	Use this interface to manage the data exchange specific to <b>Everest</b> browsers such as Items Browser, Vendors Browser, etc.
EverestProfile	Use this interface to manage the data exchange specific to <b>Everest</b> profiles such as Item Profile, Customer Profile, etc.
EverestDocument	Use this interface to manage the data exchange specific to <b>Everest</b> documents such as Sales Orders, Purchase Orders, etc.
EverestDocumentLineItem	Use this interface to manage the data exchange specific to line items of <b>Everest</b> documents such as Sales Orders, Purchase Orders, etc.
EverestPayment	Use this interface to manage the data exchange specific to payments related to <b>Everest</b> documents such as Sales Orders, Purchase Orders, etc.

## Everest Form

The EverestForm class is a client side interface that contains the following methods:

Property	Description
EverestForm.Session	This property provides access to the <b>EverestSession</b> interface.
EverestForm.Browser	This property provides access to the <b>EverestBrowser</b> interface.
EverestForm.Profile	This property provides access to the <b>EverestProfile</b> interface.
EverestForm.Document	This property provides access to the <b>EverestDocument</b> interface.
EverestForm.Handle	This property provides the Window Handle to the <b>Everest</b> form.

Method	Description
EverestForm.GetSessionData*	Use this method to retrieve session data.
EverestForm.GetFormData	Use this method to retrieve details of controls or forms.
EverestForm.GetBrowserData*	Use this method to retrieve data from a browser where it is called or invoked.
EverestForm.GetProfileData*	Use this method to retrieve data from a profile where it is called or invoked.
EverestForm.GetData	Use this method to retrieve the dataset details on any <b>Everest</b> form.
EverestForm.QueryAction	Use this method to retrieve the status of components available in the <b>Everest</b> form.
EverestForm.ExecuteAction	Use this method to execute the Click event of a menu item or button available in the <b>Everest</b> form or to open any other <b>Everest</b> form.
EverestForm.AddCustomPage*	Use this method to add your custom forms to any <b>Everest</b> profile.
EverestForm.SetFormData	Use this method to set data to a control.

Method	Description
EverestForm.SetCustomFieldData	Use this method to set custom fields in any profile where custom fields are available.
EverestForm.UpdateDocument*	Use this method to modify documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents. To create new documents, call the <b>ExecuteAction</b> method with a menu item for creating new documents and then call the <b>UpdateDocument</b> interface method for updating the values.
EverestForm.AddLineItem*	Use this method to add line items in documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.
EverestForm.ModifyLineItem*	Use this method to make changes to line items in documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.
EverestForm.DeleteLineItem*	Use this method to delete line items from documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.
EverestForm.SetPOSPaymentDetails*	Use this method to place data into the controls on the payment details grid of the <b>Payment Details</b> dialog box in <b>Everest POS</b> .
EverestForm.GetPOSPaymentDetails*	Use this method to get the details of POS payments.

*\*These methods are deprecated in this **Everest** version and will be removed from a future version of **Everest**.*

## EverestForm.GetSessionData



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to retrieve session data.

### Syntax

```
function GetSessionData(bExtendedInfo: WordBool = False): WideString
```

### Parameters

#### *bExtendedInfo*

Use this optional parameter to include or exclude sensitive information. When *bExtendedInfo* is not passed, it will assume a default value of `False`. When *bExtendedInfo* is passed as `True`, the return value includes sensitive information such as `UID`, `ConnectionString`, `User Password`, etc.

### Return Value

This method returns XML with session information. The information returned depends upon the value of the *bExtendedInfo* parameter. See the following XML samples:

- Session Data XML Sample (*bExtendedInfo* = `False`)
- Session Data XML Sample (*bExtendedInfo* = `True`)

For the **Session Data XML Sample (*bExtendedInfo* = `True`)**, see the **Everest Add-Ins** help files.

## Session Data XML Sample (bExtendedInfo = False)

```

- <SessionData>
  <DBServerName>EVEREST</DBServerName>
  <AppServerName>EVEREST</AppServerName>
  <DatabaseName>EVEREST_SAMPLE</DatabaseName>
  <DBUserName>sa</DBUserName>
  <UserCode>SUP</UserCode>
  <CompanyCode>ASC</CompanyCode>
  <ComputerName>EVEREST</ComputerName>
  <LastErrorCode />
  <LastErrorTimeStamp />
  <SecChkUserCode />
  <LastSecNumber />
  <Department>MAIN</Department>
  <Jurisdiction>VA</Jurisdiction>
  <LoggedInDateTime>08/03/2007 02:24:59 PM</LoggedInDateTime>
  <ReportGlobalRepository>\\EVEREST\REPORTS</ReportGlobalRepository>
  <ReportServer>EVEREST</ReportServer>
  <ReportServerURL>HTTP://EVEREST/reportserver/CRYSTALLIZE.ASP</ReportServerURL>
  <sJournalGroup />
  <bExtendedMode>F</bExtendedMode>
  <LastHeartBeatTime>08/03/2007 2:38:48 PM</LastHeartBeatTime>
  <ProductType>1</ProductType>
  <ProxyUserName />
  <TimeZoneOffset>-330</TimeZoneOffset>
  <ExtendedErrorInformation />
  <ProxyDomainName />
  <ExtendedErrorInformation1>2</ExtendedErrorInformation1>
  <LockStatus />
  <EverestDate>08/03/2007</EverestDate>
  <WebClientCode />
  <ACRMServer />
  <RegisteredCompany>ASC</RegisteredCompany>
  <POSCount>0</POSCount>
  <DatabaseType>1</DatabaseType>
  <InternationalMode>0</InternationalMode>
  <TimeZoneIndex IsDayLightSavingEnabled="false">39</TimeZoneIndex>
</SessionData>

```

Figure 2.36: Session Data XML Sample (bExtendedInfo = False)



“Add-In Information Dialog Box” on page 125

“XML Samples” on page 154

“Everest Add-In Samples” on page 153

## EverestForm.GetFormData

### Purpose

Use this method to retrieve details of controls or forms.

### Syntax

```
function GetFormData(const sControlName: WideString = ""; bRecursive:  
WordBool=False): WideString;
```

### Parameters

#### *sControlName*

Use this parameter to specify the control name to return data only for the specified control.

**Example** sleDescription

This parameter is optional. If the control name is not specified, all of the control details of the form will be returned.

#### *bRecursive*

Use this parameter to indicate whether data should be retrieved for the complete child tree only or only children of the first level. If passed as true with the control name, you will get a result for that control with all its child control details.

### Return Value

This method returns the form details in XML format. For an XML Sample, see the **Item Profile Form Data XML Sample** in the **Everest Add-Ins** help files.



["Add-In Information Dialog Box" on page 125](#)

["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)



## EverestForm.GetBrowserData



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to retrieve data from a browser where it is called or invoked.

### Syntax

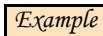
function GetBrowserData: WideString

### Parameters

None

### Return Value

This method returns the form details in XML format for **Everest** browsers.



The **Items** browser, **Customers** browser, and **Sales Orders** browser are examples of **Everest** browsers.

If you request the details for any other form type, you will receive the following error in XML format:

```
<ERROR>This method can be used only with Browsers</ERROR>
```

**Figure 2.37: EverestForm.GetBrowserData Error Message**

For an XML sample, refer to the Browser Data XML Sample (EverestForm.GetBrowserData).



[“Add-In Information Dialog Box” on page 125](#)

[“XML Samples” on page 154](#)

[“Everest Add-In Samples” on page 153](#)

## Browser Data XML Sample (EverestForm.GetBrowserData)

```
- <icqBrowser>
- <Data>
  <Row ID="1" Selected="T" ITEMNO="PTAX" DESCRIPT="+ Tax Difference" ACTIVE="T"
  COST="0" BRAND="" CATEGORY="MISC" MAX_STOCK="0" MIN_NEEDED="" MODEL=""
  QTY_STK="0" TOTAL_STOCK="0" Q_ON_CREDI="0" Q_ON_ORDER="0"
  Q_ON_CREDIT_COMM="0" Q_ON_ORDER_COMM="0" Q_ON_RESERVE_COMM="0"
  Q_ON_RMA_COMM="0" Q_ON_RESER="0" Q_ON_RMA="0" SALE_MEAS="EA"
  SELLPRIC="0" MATRIX_ITEM_TYPE="0" TYPE="2" ENTRY_TYPE="1" PURC_FRAC="0"
  AVG_COST="0" AVAILABLE="0" AVG_STOCK="0" SERIALIZE="F" SALEFRACTION="1"
  MARKUP="" PURC_MEAS="EA" MANUCODE="" FF_ACTIVE="F" FF="0" INVENTORED="F"
  AUTOSERIAL="F" FRACTION="1" MPRICE="" STOCK_MEAS="EA" LAST_COST="0"
  LASTADDONCOST="0" P_PURC_FRAC="0" S_PURC_FRAC="0"
  SALEPRICE_DEC_ACCURACY="2" PURCPRICE_DEC_ACCURACY="2"
  PURCFRACTION="1" />
</Data>
</icqBrowser>
```

**Figure 2.38: Browser Data XML Sample (EverestForm.GetBrowserData)**

## EverestForm.GetProfileData



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to retrieve data from a profile where it is called or invoked.

### Syntax

```
function GetProfileData: WideString
```

### Parameters

None

### Return Value

This method returns the form details in XML format for **Everest** profiles.



The **Item** profile and the **Customer** profile are examples of **Everest** profiles.

If you request the details for any other form type, you will receive the following error in XML format:

```
<ERROR>This method can be used only with profiles</ERROR>
```

### Figure 2.39: EverestForm.GetProfileData Error Message

For an XML sample, refer to the Profile Data XML Sample (EverestForm.GetProfileData).



[“Add-In Information Dialog Box” on page 125](#)

[“XML Samples” on page 154](#)

[“Everest Add-In Samples” on page 153](#)

## Profile Data XML Sample

```

- <ProfileData>
- <quProfile>
- <Data>
  <Row ID="1" ACTIVE="T" AUTOSERIAL="F" ITEMNO="4453" CATEGORY="France"
    DESCRIPT="1994 Pessac-Leognan Blanc" Q_ON_ORDER="0" Q_ON_RESER="1"
    Q_ON_RMA="10" Q_ON_CREDI="0" MIN_NEEDED="0" COST="70" AVG_COST="70"
    FF_ACTIVE="F" FF="0" TAXABLE="F" MDESC="" SELL_DESC="" S_EXP_DESC=""
    P_EXP_DESC="" BRAND="" MODEL="" SERIALIZE="F" INVENTORED="T" DEP_CODE=""
    PURC_MEAS="EA" SALE_MEAS="EA" PGT_1="" PGT_2="" PGT_3="" PGT_4=""
    PGT_5="" COMMISSION="T" LSI_DOC="300220" LSI_DATE="10/16/2006"
    LSI_AMT="588" LCM_DOC="600100" LCM_DATE="5/31/2006" LCM_AMT="84"
    LPR_DOC="" LPR_DATE="" LPR_AMT="" LDM_DOC="" LDM_DATE="" LDM_AMT="0" />
  </Data>
</quProfile>
- <quForm>
  <Data />
</quForm>
- <quSimilarItems>
  <Data />
</quSimilarItems>
</ProfileData>

```

Figure 2.40: Profile Data XML Sample

## EverestForm.GetData

### Purpose

Use this method to retrieve data from any form.

### Syntax

function GetData: WideString

### Parameters

None

### Return Value

This method returns the data from the datasets available on the Form in XML format. For an XML sample, refer to the GetData XML Sample.



["Add-In Information Dialog Box" on page 125](#)

["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

### EverestForm.GetData XML Sample

For the EverestForm.GetData XML Sample, see the Everest Add-Ins help files.

## EverestForm.QueryAction

### Purpose

Use this method to retrieve the status of components available in the **Everest** form.

### Syntax

```
function QueryAction(const sCommand: WideString): WordBool;
```

### Parameters

*sCommand*

If you pass a menu item or button (TButtonControl or TSpeedButton) the system returns a value indicating if it is enabled or not.

*Example*

Call QueryAction (mmiNew). If the menu or button is enabled, the result returns True.

### Return Value

This method returns True or False.



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## EverestForm.ExecuteAction

### Purpose

Use this method to execute the Click event of a menu item or button available in the **Everest** form or to open any other **Everest** form.

### Syntax

```
function ExecuteAction(const sCommand: WideString): WordBool;
```

### Parameters

#### *sCommand*

If you pass the name of a menu item or button (TButton or TSpeedButton) the system calls the respective Click event.

**Example** Call ExecuteAction (“mmiNew”). If the menu or button is enabled, the click event is called and the result returns True.

If you pass the **Everest** form details along with required parameters, the system opens the respective form and assigns the parameter values.

**Example** The following value for sCommand will open Sales Order #200234:

```
<Data>
<Form>TfrmSalesDocument</Form>
<Params>
  <Param>200234</Param>
  <Param>8</Param>
</Params>
</Data>
```

### Return Value

This method returns True if the system is able to successfully call the click event or open an **Everest** form; otherwise it returns False.



[“XML Samples” on page 154](#)

[“Everest Add-In Samples” on page 153](#)

## Everest Forms Supported with the ExecuteAction Method

Table below illustrates the list of **Everest** forms supported in 5.0 with the *ExecuteAction* method:

Everest Forms Supported with the ExecuteAction Method			
Everest Form	<Form>	<Param>	Example
Sales Document	TfrmSalesDocument	Document Number Document Type	<pre>&lt;Data&gt; &lt;Form&gt;TfrmSalesDocument&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;100001&lt;/Param&gt;   &lt;Param&gt;8&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre> <p>Leave Document Number empty to open a New document</p> <p>Document Types: SalesQuote = 7 SalesOrder = 8 SalesInvoice = 9 CreditQuote = 10 CreditOrder = 11 CreditMemo = 12</p>
Purchase Document	TfrmPurchaseDocument	Document Number Document Type	<pre>&lt;Data&gt; &lt;Form&gt;TfrmPurchaseDocument&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;100001&lt;/Param&gt;   &lt;Param&gt;2&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre> <p>Leave Document Number empty to open a New document</p> <p>Document Types: PurchaseQuote = 1 PurchaseOrder = 2 PurchaseInvoice = 3 PurchaseReceipt = 24 DebitQuote = 4 DebitOrder = 5 DebitMemo = 6</p>

Everest Forms Supported with the ExecuteAction Method			
Everest Form	<Form>	<Param>	Example
Matrix Items	TfrmMatrixItemsDlg	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmMatrixItemsDlg&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Item Aliases	TfrmItemAliasBrowser	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmItemAliasBrowser&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Related Items	TfrmItemUpGradeBrowser	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmItemUpGradeBrowser&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Add-ons	TfrmSimilarItemsProfile	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmSimilarItemsProfile&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Custom Tracking	TfrmCustomTrackingSelection	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmCustomTrackingSelection&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Multimedia Browser	TfrmMultimediaSetup	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmMultimediaSetup&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Custom Fields	TfrmCustomFieldsProfile	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmCustomFieldsProfile&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>



Everest Forms Supported with the ExecuteAction Method			
Everest Form	<Form>	<Param>	Example
E-Commerce Setup	TfrmOnlineShopSetupProfile	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmOnlineShopSetupProfile&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Replenishment Details	TfrmItemReplenishmentBrowser	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmItemReplenishmentBrowser&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Item Summary	TfrmItemSummary	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmItemSummary&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Item Sales History	TfrmItemSalesHistoryBrowser	Item Code Sales Document Type	<pre>&lt;Data&gt; &lt;Form&gt;TfrmItemSalesHistoryBrowser&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt;   &lt;Param&gt;8&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre> <p>Sales Document Types:</p> <ul style="list-style-type: none"> <li>SalesQuote = 7</li> <li>SalesOrder = 8</li> <li>SalesInvoice = 9</li> <li>CreditQuote = 10</li> <li>CreditOrder = 11</li> <li>CreditMemo = 12</li> </ul>

Everest Forms Supported with the ExecuteAction Method			
Everest Form	<Form>	<Param>	Example
Item Purchase History	TfrmItemPurchaseHistoryBrowser	Item Code Purchase Document Type	<pre>&lt;Data&gt; &lt;Form&gt;TfrmItemPurchaseHistoryBrowser&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt;   &lt;Param&gt;2&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre> <p>Purchase Document Types:            PurchaseQuote = 1            PurchaseOrder = 2            PurchaseInvoice = 3            PurchaseReceipt = 24            DebitQuote = 4            DebitOrder = 5            DebitMemo = 6</p>
Stock Chart	TfrmItemChart	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmItemChart&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Item Stock Status	TfrmItemStockStatus	Number of Items Item Code(s) Measure Type	<pre>&lt;Data&gt; &lt;Form&gt;TfrmItemStockStatus&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;2&lt;/Param&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt;   &lt;Param&gt;102WHEELS&lt;/Param&gt;   &lt;Param&gt;2&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre> <p>Measure Types:            Stock Measure = 1            Sales Measure = 2            Purchase Measure = 3</p>

Everest Forms Supported with the ExecuteAction Method			
Everest Form	<Form>	<Param>	Example
Lot Stock Status Browser	TfrmLotAreaStockStatus	Number of Items Item Code(s) Measure Type	<pre>&lt;Data&gt; &lt;Form&gt;TfrmLotAreaStockStatus&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;2&lt;/Param&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt;   &lt;Param&gt;102WHEELS&lt;/Param&gt;   &lt;Param&gt;2&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre> <p>Measure Types: Stock Measure = 1 Sales Measure = 2 Purchase Measure = 3</p>
Inventory Adjustment	TfrmInventoryAdjustment	Item Code Inventory Adjustment Type	<pre>&lt;Data&gt; &lt;Form&gt;TfrmInventoryAdjustment&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt;   &lt;Param&gt;1&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre> <p>Inventory Adjustment Types: Quantity Adjustment = 1 Write-off Quantity = 1 Cost Adjustment = 3 Price Protection = 4</p>
Inventory History	TfrmInventoryHistory	Item Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmInventoryHistory&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;101FGTIPS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Category Browser	TfrmCategoryBrowser		<pre>&lt;Data&gt; &lt;Form&gt;TfrmCategoryBrowser&lt;/Form&gt; &lt;/Data&gt;</pre>
Brand Browser	TfrmBrandBrowser		<pre>&lt;Data&gt; &lt;Form&gt;TfrmBrandBrowser&lt;/Form&gt; &lt;/Data&gt;</pre>
Customer Browser	TfrmCustomerBrowser		<pre>&lt;Data&gt; &lt;Form&gt;TfrmCustomerBrowser&lt;/Form&gt; &lt;/Data&gt;</pre>

Everest Forms Supported with the ExecuteAction Method			
Everest Form	<Form>	<Param>	Example
Item Browser	TfrmItemBrowser		<Data> <Form>TfrmItemBrowser</Form> </Data>
Vendor Browser	TfrmVendorBrowser		<Data> <Form>TfrmVendorBrowser</Form> </Data>
Serial Number Browser	TfrmSerialNoBrowser		<Data> <Form>TfrmSerialNoBrowser</Form> </Data>
Sales Opportunity Browser	TfrmSalesOpportunityBrowser		<Data> <Form>TfrmSalesOpportunityBrowser</Form> </Data>
Assembly WO Browser	TfrmAssemblyWOBrowser		<Data> <Form>TfrmAssemblyWOBrowser</Form> </Data>
Sales Quote Browser	TfrmSalesQuoteBrowser		<Data> <Form>TfrmSalesQuoteBrowser</Form> </Data>
Sales Order Browser	TfrmSalesOrderBrowser		<Data> <Form>TfrmSalesOrderBrowser</Form> </Data>
Sales Invoice Browser	TfrmSalesInvoiceBrowser		<Data> <Form>TfrmSalesInvoiceBrowser</Form> </Data>
POS Browser	TfrmPointOfSaleBrowser		<Data> <Form>TfrmPointOfSaleBrowser</Form> </Data>
Credit Quote Browser	TfrmCreditQuoteBrowser		<Data> <Form>TfrmCreditQuoteBrowser</Form> </Data>
Credit Order Browser	TfrmCreditOrderBrowser		<Data> <Form>TfrmCreditOrderBrowser</Form> </Data>
Credit Memo Browser	TfrmCreditmemoBrowser		<Data> <Form>TfrmCreditmemoBrowser</Form> </Data>
Purchase Quote Browser	TfrmPurchaseQuoteBrowser		<Data> <Form>TfrmPurchaseQuoteBrowser</Form> </Data>
Purchase Order Browser	TfrmPurchaseOrderBrowser		<Data> <Form>TfrmPurchaseOrderBrowser</Form> </Data>

Everest Forms Supported with the ExecuteAction Method			
Everest Form	<Form>	<Param>	Example
Purchase Receipt Browser	TfrmPurchaseReceiptBrowser		<Data> <Form>TfrmPurchaseReceiptBrowser</Form> </Data>
Purchase Invoice Browser	TfrmPurchaseInvoiceBrowser		<Data> <Form>TfrmPurchaseInvoiceBrowser</Form> </Data>
Debit Quote Browser	TfrmDebitQuoteBrowser		<Data> <Form>TfrmDebitQuoteBrowser</Form> </Data>
Debit Order Browser	TfrmDebitOrderBrowser		<Data> <Form>TfrmDebitOrderBrowser</Form> </Data>
Debit Memo Browser	TfrmDebitMemoBrowser		<Data> <Form>TfrmDebitMemoBrowser</Form> </Data>
Multiple Stock Transfer	TfrmMultipleStockTransfer	Item Code	<Data> <Form>TfrmMultipleStockTransfer</Form> <Params> <Param>101FGTIPS</Param> </Params> </Data>
Customer Profile	TfrmCustomerProfile	Customer Code	<Data> <Form>TfrmCustomerProfile</Form> <Params> <Param>100100</Param> </Params> </Data>
Item Profile	TfrmItemProfile	Item Code	<Data> <Form>TfrmItemProfile</Form> <Params> <Param>101FGTIPS</Param> </Params> </Data>
Vendor Profile	TfrmVendorProfile	Vendor Code	<Data> <Form>TfrmVendorProfile</Form> <Params> <Param>BIRKEN</Param> </Params> </Data>

Everest Forms Supported with the ExecuteAction Method			
Everest Form	<Form>	<Param>	Example
Category Profile	TfrmCategoryProfile	Category Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmCategoryProfile&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;BOOKS&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>
Brand Profile	TfrmBrandProfile	Brand Code	<pre>&lt;Data&gt; &lt;Form&gt;TfrmBrandProfile&lt;/Form&gt; &lt;Params&gt;   &lt;Param&gt;3M&lt;/Param&gt; &lt;/Params&gt; &lt;/Data&gt;</pre>

## EverestForm.AddCustomPage



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to add your custom forms to any **Everest** profile.

### Syntax

```
function AddCustomPage(const sCaption,  
    sObjectId: WideString): WordBool;
```

### Parameters

#### *sCaption*

This is the caption for the newly created custom page.

#### *sObjectID*

The is the object ID of your custom form.

### Return Value

This method returns True or False based on success.



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## EverestForm.SetFormData

### Purpose

Use this method to set data to a control. SetFormData is unique to every form.

### Syntax

```
function SetFormData(const sControlData: WideString): Integer;
```

### Parameters

#### *sControlData*

Refer to the sControlData schema to code this parameter. The following elements can be set using this method:



- Left
- Top
- Height
- Width
- Text
- Visible
- Focused

 ["sControlData Schema" on page 143](#)

### Return Value

This method returns an integer as follows:

Integer	Description
0	Successful
-1	General Failure
-2	Invalid Control Data
-3	Invalid Control Name or Control Name is Empty
-4	Control is Not Enabled

 ["XML Samples" on page 154](#)  
 ["Everest Add-In Samples" on page 153](#)



## EverestForm.SetCustomFieldData

### Purpose

Use this method to set custom fields in any profile where custom fields are available.

### Syntax

```
SetCustomFieldData(iIndex, iType: Integer;  
    const sValue: WideString): Integer;
```

### Parameters

#### *iIndex*

This integer is for the record reference. For example, in the case of CustChar2, the index has to be 2.

#### *iType*

This integer represents the custom field type. Available data types are as follows and are available in EverestConsts\_TLB as an enumerated data type:

- EverestConsts.enumFTCustomFields.ftCharacters
- EverestConsts.enumFTCustomFields.ftDates
- EverestConsts.enumFTCustomFields.ftLogicals
- EverestConsts.enumFTCustomFields.ftNumerics
- EverestConsts.enumFTCustomFields.ftMemo

#### *sValue*

This is the data to be set. For numeric data, pass the numeric value as text.

## Return Value

If the process is successful then the method returns 0. If there is a general error the return value is -1. The following table shows the result descriptions for the SetCustomFieldData method:

Integer	Description
0	Successful
-1	General Error
-2	Value is Empty
-3	Record Number Not Passed
-4	Field Type Not Passed
-5	Field Type Passed is Not Valid
-6	Record Number Passed is Out of Range



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## EverestForm.UpdateDocument



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to modify documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents. To create new documents, call the **ExecuteAction** method with a menu item for creating new documents and then call the **UpdateDocument** interface method for updating the values.

### Syntax

```
function UpdateDocument(const sDocumentXML: WideString): Smallint;
```

### Parameters

#### *sDocumentXML*

This is the XML data to be passed. Refer to the SalesDocument and PurchaseDocument schemas to code this information.



The SalesDocument and PurchaseDocument Schemas for **Everest Add-Ins** are the same schemas that are used for **Everest API**.



["SalesDocument Schema" on page 146](#)

["PurchaseDocument Schema" on page 144](#)

For XML Samples that show the XML format for the UpdateDocument method, refer to the following samples:

- Update Sales Document XML Sample
- Update Purchase Document XML Sample



["Update Sales Document XML Sample" on page 86](#)

["Update Purchase Document XML Sample" on page 88](#)

### Return Value

If the process is successful then the method returns 0. If there is a generic error or exception, the return value is -1. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## Update Sales Document XML Sample

```

- <SalesDocument>
  <CustomerCode>100101</CustomerCode>
  <ContactCode>0</ContactCode>
  <ShipToCode>0</ShipToCode>
  <SalesRepCode>JS</SalesRepCode>
  <PayTermsCode>N30</PayTermsCode>
  <OverrideDueDate>F</OverrideDueDate>
  <DueDate>2006-11-15</DueDate>
  <JurisdictionCode>VAOUT</JurisdictionCode>
  <ShipViaCode>FEDXG</ShipViaCode>
  <DeliveryDate>2006-10-16</DeliveryDate>
  <ExpiryDate />
  <PromotionCode />
  <TaxCalculationType>1</TaxCalculationType>
  <FOB />
  <Reference />
  <CustomerPONo>66243</CustomerPONo>
  <CustomerPODate />
  <LocationSubLocation />
  <Processed />
  <FreightPaddingCode />
  <AllowBackOrderCreation />
  <ShipmentTrackingNo>1111</ShipmentTrackingNo>
  <ShipmentTrackingNote>sample</ShipmentTrackingNote>
  <ExchangeRate />
- <CustomFields>
  <CustomCharacterField Number="1" />
  <CustomCharacterField Number="2" />
  <CustomCharacterField Number="3" />
  <CustomCharacterField Number="4" />
  <CustomDateField Number="1" />
  <CustomDateField Number="2" />
  <CustomDateField Number="3" />
  <CustomDateField Number="4" />
  <CustomLogicalField Number="1" />
  <CustomLogicalField Number="2" />
  <CustomLogicalField Number="3" />
  <CustomLogicalField Number="4" />
  <CustomNumericField Number="1" />
  <CustomNumericField Number="2" />
  <CustomNumericField Number="3" />
  <CustomNumericField Number="4" />
  <CustomMemoField Number="1" />
  <CustomMemoField Number="2" />
  <CustomMemoField Number="3" />
  <CustomMemoField Number="4" />
</CustomFields>

```

Figure 2.41: Update Sales Document XML Sample (Continued on Next Page)

## Update Sales Document XML Sample Continued

```

- <LineItems>
- <LineItem>
  <Sequence>1</Sequence>
  <ItemCode>BLSOFT66</ItemCode>
  <Description />
  <Quantity>11.000000</Quantity>
  <ItemPrice>11.600000</ItemPrice>
  <UOMCode>EA</UOMCode>
  <CommittedQuantity />
  <Cost>10.000000</Cost>
  <VendorCode>ELEC_LENSES</VendorCode>
  <DiscountValue />
  <DiscountPercent />
  <TaxCode />
  <RevenueAccountCode>41010</RevenueAccountCode>
  <DiscountAccountCode>42010</DiscountAccountCode>
  <ExpenseAccountCode>51010</ExpenseAccountCode>
  <DepartmentCode>CA</DepartmentCode>
  <JobCode />
  <Critical>F</Critical>
  <DependentOn />
  <Dropship />
  <DeliveryDate />
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1">F</LICustomLogicalField>
  <LICustomLogicalField Number="2">F</LICustomLogicalField>
  <LICustomLogicalField Number="3">F</LICustomLogicalField>
  <LICustomLogicalField Number="4">F</LICustomLogicalField>
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>
</LineItems>
</SalesDocument>

```

Figure 2.42: Update Sales Document XML Sample Continued

## Update Purchase Document XML Sample

```

- <PurchaseDocument>
  <VendorCode>BIRKEN</VendorCode>
  <ContactCode>0</ContactCode>
  <ShipToCode>0</ShipToCode>
  <PurchaseRepCode>VF</PurchaseRepCode>
  <PayTermsCode>PP</PayTermsCode>
  <OverRideDueDate>F</OverRideDueDate>
  <DueDate>2006-09-06</DueDate>
  <ShipViaCode>UPSG</ShipViaCode>
  <DeliveryDate>2006-10-17</DeliveryDate>
  <ExpiryDate />
  <FOBType>1</FOBType>
  <FOB />
  <Reference />
  <VendorInvoice />
  <VendorInvoiceDate>2006-09-06</VendorInvoiceDate>
  <LocationSubLocation />
  <ShipmentTrackingNo />
  <Processed>F</Processed>
  <TermsFromPIDate>F</TermsFromPIDate>
  <AllowBackOrderCreation>F</AllowBackOrderCreation>
  <HandOff>F</HandOff>
  <DocumentAlias />
  <Notes />
  <ExchangeRate>1.000000</ExchangeRate>
- <CustomFields>
  <CustomCharacterField Number="1" />
  <CustomCharacterField Number="2" />
  <CustomCharacterField Number="3" />
  <CustomCharacterField Number="4" />
  <CustomDateField Number="1" />
  <CustomDateField Number="2" />
  <CustomDateField Number="3" />
  <CustomDateField Number="4" />
  <CustomLogicalField Number="1">F</CustomLogicalField>
  <CustomLogicalField Number="2">F</CustomLogicalField>
  <CustomLogicalField Number="3">F</CustomLogicalField>
  <CustomLogicalField Number="4">F</CustomLogicalField>
  <CustomNumericField Number="1">.0000</CustomNumericField>
  <CustomNumericField Number="2">.0000</CustomNumericField>
  <CustomNumericField Number="3">.0000</CustomNumericField>
  <CustomNumericField Number="4">.0000</CustomNumericField>
  <CustomMemoField Number="1" />
  <CustomMemoField Number="2" />
  <CustomMemoField Number="3" />
  <CustomMemoField Number="4" />
</CustomFields>

```

Figure 2.43: Update Purchase Document XML Sample (Continued on Next Page)

## Update Purchase Document XML Sample Continued

```

- <LineItems>
- <LineItem>
  <Sequence>1</Sequence>
  <ItemCode>4453</ItemCode>
  <Description />
  <Quantity>7.000000</Quantity>
  <Cost>70.000000</Cost>
  <UOMCode>EA</UOMCode>
  <DiscountValue>.000000</DiscountValue>
  <DiscountPercent />
  <TaxCode />
  <RecoverableTax />
  <Account />
  <DepartmentCode>MAIN</DepartmentCode>
  <JobCode />
  <CostExclusiveTax />
  <DeliveryDate />
  <CustomerPO />
  <CustomerPODate />
  <VendorPartNo>200157</VendorPartNo>
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1" />
  <LICustomLogicalField Number="2" />
  <LICustomLogicalField Number="3" />
  <LICustomLogicalField Number="4" />
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>
</LineItems>
</PurchaseDocument>

```

Figure 2.44: Update Purchase Document XML Sample Continued

## EverestForm.AddLineItem



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to add line items in documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.

### Syntax

```
function AddLineItem (const sLineItemXML: WideString): Smallint;
```

### Parameters

#### *sLineItemXML*

This is the XML data to be passed. Refer to the LineItem schema to code this parameter. For an XML sample, refer to the Add Line Item XML Sample.



["LineItem Schema" on page 148](#)

### Return Value

This method returns an integer. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)



## Add Line Item XML Sample

```
- <LineItems>
- <LineItem>
  <ItemCode>BLSOFT66</ItemCode>
</LineItem>
- <LineItem>
  <ItemCode>102Wheels</ItemCode>
</LineItem>
- <LineItem>
  <ItemCode>CapeFear</ItemCode>
</LineItem>
</LineItems>
```

Figure 2.45: Add Line Item XML Sample

## EverestForm.ModifyLineItem



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to make changes to line items in documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.

### Syntax

```
function ModifyLineItem (const sLineItemXML: WideString): Smallint;
```

### Parameters

#### *sLineItemXML*

This is the XML data to be passed. Refer to the LineItem schema to code this parameter. For an XML sample, refer to the Modify Line Item XML Sample.



["Modify Line Item XML Sample" on page 93](#)

["LineItem Schema" on page 148](#)

### Return Value

This method returns an integer. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## Modify Line Item XML Sample

```

- <LineItems>
- <LineItem>
  <Sequence>1</Sequence>
  <ItemCode>BLSOFT66</ItemCode>
  <Description />
  <Quantity>20.000000</Quantity>
  <ItemPrice>25.600000</ItemPrice>
  <UOMCode>12P</UOMCode>
  <CommittedQuantity>10</CommittedQuantity>
  <Cost>20.000000</Cost>
  <VendorCode>ELEC_LENSES</VendorCode>
  <DiscountValue />
  <DiscountPercent />
  <TaxCode />
  <RevenueAccountCode>41010</RevenueAccountCode>
  <DiscountAccountCode>42010</DiscountAccountCode>
  <ExpenseAccountCode>51010</ExpenseAccountCode>
  <DepartmentCode>CA</DepartmentCode>
  <JobCode />
  <Critical>F</Critical>
  <DependentOn />
  <Dropship />
  <DeliveryDate />
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1">F</LICustomLogicalField>
  <LICustomLogicalField Number="2">F</LICustomLogicalField>
  <LICustomLogicalField Number="3">F</LICustomLogicalField>
  <LICustomLogicalField Number="4">F</LICustomLogicalField>
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>

```

Figure 2.46: Modify Line Item XML Sample (Continued on Next Page)

## Modify Line Item XML Sample Continued

```

- <LineItem>
  <Sequence>2</Sequence>
  <ItemCode>101FGTIPS</ItemCode>
  <Description />
  <Quantity>11.000000</Quantity>
  <ItemPrice>11.600000</ItemPrice>
  <UOMCode>EA</UOMCode>
  <CommittedQuantity />
  <Cost>10.000000</Cost>
  <VendorCode>ELEC_LENSSES</VendorCode>
  <DiscountValue />
  <DiscountPercent />
  <TaxCode />
  <RevenueAccountCode>41010</RevenueAccountCode>
  <DiscountAccountCode>42010</DiscountAccountCode>
  <ExpenseAccountCode>51010</ExpenseAccountCode>
  <DepartmentCode>CA</DepartmentCode>
  <JobCode />
  <Critical>F</Critical>
  <DependentOn />
  <Dropship />
  <DeliveryDate />
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1">F</LICustomLogicalField>
  <LICustomLogicalField Number="2">F</LICustomLogicalField>
  <LICustomLogicalField Number="3">F</LICustomLogicalField>
  <LICustomLogicalField Number="4">F</LICustomLogicalField>
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>
</LineItems>

```

**Figure 2.47: Modify Line Item XML Sample Continued**

## EverestForm.DeleteLineItem



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to delete line items from documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.

### Syntax

```
function DeleteLineItem (const sLineItemSequenceXML: WideString): Smallint;
```

### Parameters

#### *sLineItemSequenceXML*

This is the XML data to be passed. Refer to the DeleteLineItem schema to code this parameter. Only **SequenceNo** is required. For an XML sample, refer to the Delete Line Item XML Sample.



[“DeleteLineItem Schema” on page 150](#)

### Return Value

This method returns an integer. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form



[“XML Samples” on page 154](#)

[“Everest Add-In Samples” on page 153](#)

## Delete Line Item XML Sample

```
- <LineItems>
  <Sequence>1</Sequence>
  <Sequence>2</Sequence>
  <Sequence>3</Sequence>
  <Sequence>4</Sequence>
</LineItems>
```

Figure 2.48: Delete Line Item XML Sample

## EverestForm.SetPOSPaymentDetails



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to place data into the controls on the payment details grid of the **Payment Details** dialog box in **Everest** POS.

The following figure shows the location of the controls on the **Payment Details** dialog box.

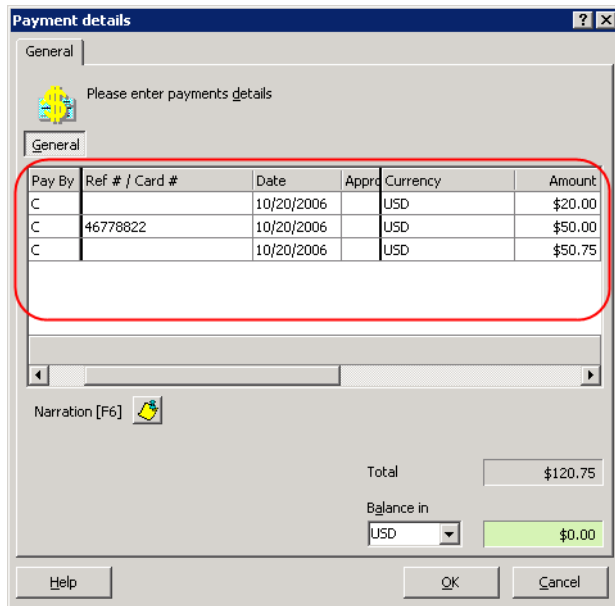


Figure 2.49: Payment Details Dialog Box with Controls Circled

### Syntax

function SetPOSPaymentDetails (const sPaymentData: WideString): Smallint;

### Parameters

#### *sPaymentData*

This is the XML data to be set. Refer to the POSPayment schema to code this parameter. For an XML sample that shows the POS payment details, see the POS Payment XML Sample.



["POS Payment XML Sample" on page 98](#)

["POSPayment Schema" on page 151](#)

## Return Value

This method returns an integer. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## EverestForm.GetPOSPaymentDetails



This method is deprecated in this **Everest** version and will be removed from a future version of **Everest**.

### Purpose

Use this method to get the details of POS payments.

### Syntax

```
function GetPOSPaymentDetails (iSequenceNumber : Integer = 0): WideString;
```

### Parameters

#### *iSequenceNo*

This is an integer representing the sequence number of the POS payment details. This is an optional parameter; when omitted it returns all payment details (all rows from the payment grid).

### Return Value

This method returns XML with POS payment details. For an XML sample that shows the POS payment details, see the POS Payment XML Sample.



["Add-In Information Dialog Box" on page 125](#)

["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## POS Payment XML Sample

```
- <Payments>
- <Payment>
  <Sequence>1</Sequence>
  <PayBy>C</PayBy>
  <ReferenceNo>124344</ReferenceNo>
  <ReferenceDate>2006-10-12</ReferenceDate>
  <Narration>my addin</Narration>
  <ApprovalCode>24234234</ApprovalCode>
  <Amount>100</Amount>
- <CurrencyDetails>
  <PayInCurrencyCode>USD</PayInCurrencyCode>
  <PayInCurrencyExchangeRate>1</PayInCurrencyExchangeRate>
  <PayInAmount>100</PayInAmount>
  </CurrencyDetails>
</Payment>
</Payments>
```

Figure 2.50: POS Payment XML Sample



## Everest Session

The EverestSession class is a client side interface that contains the following methods:

Method	Description
EverestSession.GetData	Use this method to retrieve session data.
EverestSession.SetData	Use this method to set up the session details.

 ["Everest Form" on page 62](#)

### EverestSession.GetData

#### Purpose

Use this method to retrieve session data.

#### Syntax

```
function GetData (bExtendedInfo: WordBool = False): WideString
```

#### Parameters

##### *bExtendedInfo*

Use this optional parameter to include or exclude sensitive information. When bExtendedInfo is not passed, it will assume a default value of False. When bExtendedInfo is passed as True, the return value includes sensitive information such as UID, ConnectionString, User Password, etc.

#### Return Value

This method returns XML with session information. The information returned depends upon the value of the bExtendedInfo parameter. See the following XML samples:

- Session Data XML Sample (bExtendedInfo = False)
- Session Data XML Sample (bExtendedInfo = True)

For the **Session Data XML Sample (bExtendedInfo = True)**, see the **Everest Add-Ins** help files.

## Session Data XML Sample (bExtendedInfo = False)

```

- <SessionData>
  <DBServerName>EVEREST</DBServerName>
  <AppServerName>EVEREST</AppServerName>
  <DatabaseName>EVEREST_SAMPLE</DatabaseName>
  <DBUserName>sa</DBUserName>
  <UserCode>SUP</UserCode>
  <CompanyCode>ASC</CompanyCode>
  <ComputerName>EVEREST</ComputerName>
  <LastErrorCode />
  <LastErrorTimeStamp />
  <SecChkUserCode />
  <LastSecNumber />
  <Department>MAIN</Department>
  <Jurisdiction>VA</Jurisdiction>
  <LoggedInDateTime>08/03/2007 02:24:59 PM</LoggedInDateTime>
  <ReportGlobalRepository>\\EVEREST\REPORTS</ReportGlobalRepository>
  <ReportServer>EVEREST</ReportServer>
  <ReportServerURL>HTTP://EVEREST/reportserver/CRYSTALLIZE.ASP</ReportServerURL>
  <sJournalGroup />
  <bExtendedMode>F</bExtendedMode>
  <LastHeartBeatTime>08/03/2007 2:38:48 PM</LastHeartBeatTime>
  <ProductType>1</ProductType>
  <ProxyUserName />
  <TimeZoneOffset>-330</TimeZoneOffset>
  <ExtendedErrorInformation />
  <ProxyDomainName />
  <ExtendedErrorInformation1>2</ExtendedErrorInformation1>
  <LockStatus />
  <EverestDate>08/03/2007</EverestDate>
  <WebClientCode />
  <ACRMServer />
  <RegisteredCompany>ASC</RegisteredCompany>
  <POSCount>0</POSCount>
  <DatabaseType>1</DatabaseType>
  <InternationalMode>0</InternationalMode>
  <TimeZoneIndex IsDayLightSavingEnabled="false">39</TimeZoneIndex>
</SessionData>

```

Figure 2.51: Session Data XML Sample (bExtendedInfo = False)



“Add-In Information Dialog Box” on page 125

“XML Samples” on page 154

“Everest Add-In Samples” on page 153

## EverestSession.SetData

### Purpose

Use this method to set up the session details.

### Syntax

```
function SetData(sSessionDataXML :String) : WideString
```

### Parameters

*sSessionData XML*

This is the XML data to be passed. Refer to the SessionData Schema to code this parameter.

 ["SessionData Schema" on page 142](#)

### Return Value

This method returns the session data in XML format. For an XML sample, refer to the Session XML Sample.

 ["Add-In Information Dialog Box" on page 125](#)

["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

### Session XML Sample

```
- <SessionData>  
  <Department>CA</Department>  
  <Jurisdiction>CA</Jurisdiction>  
  <TimeZoneIndex>5</TimeZoneIndex>  
  <EverestDate>2006-12-30</EverestDate>  
</SessionData>
```

Figure 2.52: Session XML Sample

## Everest Browser

The EverestBrowser class is a client side interface that contains the following method:

Method	Description
EverestBrowser.GetData	Use this method to retrieve browser data where it is called or invoked

 ["Everest Form" on page 62](#)

### EverestBrowser.GetData

#### Purpose

Use this method to retrieve data from a browser where it is called or invoked.

#### Syntax

```
function GetData(bSelected: WordBool): WideString;
```

#### Parameters

*bSelected*

If the **bSelected** parameter is **True** then the resultant XML will return only the selected records from the browser.

#### Return Value

This method returns the form details in XML format for **Everest** browsers.

**Example** The **Items** browser, **Customers** browser, and **Sales Orders** browser are examples of **Everest** browsers.

It returns Selected Cell information, Search Field, Search Text, and can also retrieve only selected records.

The <Header> section contains the Focused tab which contains the Row ID and Column Name. Use the Grid controller's FocusRecordIndex and Focused Column properties to get this information. The <Header> section also contains the Search tag which contains the Search Field and Search Text.

If you request the details for any other form type, you will receive the following error in XML format:

```
<ERROR>This method can be used only with Browsers</ERROR>
```

**Figure 2.53: EverestForm.GetBrowserData Error Message**

For XML Samples, see the following:

- EverestForm.Browser.GetData XML Sample (bSelected = False)

- EverestForm.Browser.GetData XML Sample (bSelected = True)  
For EverestForm.Browser.GetData XML Sample (bSelected = False), see the Everest Add-Ins help files.



“Add-In Information Dialog Box” on page 125

“XML Samples” on page 154

“Everest Add-In Samples” on page 153

## EverestForm.Browser.GetData XML Sample (bSelected = True)

(When executed on the Customer browser)

```
- <icqBrowser>
  <Title>Customers</Title>
- <Header>
  - <Focused>
    <Row>1</Row>
    <Column Name="ADDRESS_LINE1">547 Franklin Street</Column>
  </Focused>
- <Search>
  <Column Name="ACCOUNT_NAME">Allied Systems</Column>
</Search>
</Header>
- <Data>
  <Row ID="1" Selected="T" ACTIVE="T" CURR_CODE="USD" CUST_CODE="100100"
  SALESPERSO="MB" CONTCODE="" ADDR_CODE="12" ACCOUNT_NAME="Allied
  Systems" DEP_CODE="MAIN" FIRST_NAME="Rob" LAST_NAME="Broido"
  ETEL1="5403422507" ETEL2="" EFAX="" EMAIL="rob@allied.com"
  STREET_ADDRESS="547 Franklin Street" CITY="ROANOKE" STATE="VA"
  COUNTRY="UNITED STATES" ZIP="24007" TYPE="4" SHIPCODE=""
  ADDR_CODE_1="12" HAND_OFF="0" CURRCODE="USD"
  BASENETBALANCE="12180.93" FRXNETBALANCE="12180.93" REQUIREPO="F"
  ADDRESS_LINE1="547 Franklin Street" ADDRESS_LINE2=""
  FULL_ADDRESS="Allied Systems Rob Broido 547 Franklin Street ROANOKE,
  VA 24007" BILLCODE="12" TITLE="" CATEGORY_CODE="" MOBILE_PHONE=""
  PAGER_NO="" URL="" TEL1="5403422507" TEL2="" EXTN_TEL1="" EXTN_TEL2=""
  FAX="" EXTN_FAX="" COMPANY_NAME="Allied Systems" IS_LOCKED="F"
  APPLY_MASK_TEL="T" APPLY_MASK_MOBILE="T" APPLY_MASK_POSTAL="T"
  CUSTCHAR1="Silver" CUSTCHAR2="" CUSTCHAR3="" CUSTCHAR4=""
  CUSTDATE1="11/2/2006" CUSTDATE2="" CUSTDATE3="" CUSTDATE4=""
  CUSTLOG1="T" CUSTLOG2="F" CUSTLOG3="F" CUSTLOG4="F" CUSTMEMO1=""
  CUSTMEMO2="" CUSTMEMO3="" CUSTMEMO4="" CUSTNUM1="0" CUSTNUM2="0"
  CUSTNUM3="0" CUSTNUM4="0" />
</Data>
</icqBrowser>
```

Figure 2.54: EverestForm.Browser.GetData XML Sample (bSelected = True)

## Everest Profile

The EverestProfile class is a client side interface that contains the following methods:

Method	Description
EverestProfile.AddCustomPage	Use this method to add user-defined custom forms to an <b>Everest</b> profile.
EverestProfile.GetData	Use this method to retrieve profile data where it is called or invoked.

 ["Everest Form" on page 62](#)

### EverestProfile.AddCustomPage

#### Purpose

Use this method to add user-defined custom forms to an **Everest** profile.

#### Syntax

```
function AddCustomPage(const sCaption,
    sObjectId: WideString): WordBool;
```

#### Parameters

*sCaption*

This is the caption for the newly created custom page.

*sObjectID*

The is the object ID of your custom form.

#### Return Value

This method returns True or False based on success.

 ["XML Samples" on page 154](#)  
["Everest Add-In Samples" on page 153](#)

## EverestProfile.GetData

### Purpose

Use this method to retrieve profile data where it is called or invoked.

### Syntax

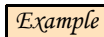
```
function GetData : String
```

### Parameters

None

### Return Value

This method returns profile data in XML format. For an XML sample, refer to the [GetData XML Sample](#).



The **Item** profile and the **Customer** profile are examples of **Everest** profiles.



["Add-In Information Dialog Box" on page 125](#)

["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

### Profile Data XML Sample (EverestForm.Profile.GetData)

For the **Profile Data XML Sample (EverestForm.Profile.GetData)**, see the **Everest Add-Ins** help files.

## Everest Document

The EverestDocument class is a client side interface that contains the following methods:

Property	Description
EverestDocument.LineItem	This property provides access to the <b>EverestDocumentLineItem</b> interface.
EverestDocument.Payment	This property provides access to the <b>EverestPayment</b> interface.

Method	Description
EverestDocument.AddCustomPage	Use this method to add your custom forms to any <b>Everest</b> profile.
EverestDocument.UpdateDocument	Use this method to modify documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents. To create new documents, call the <b>ExecuteAction</b> method with a menu item for creating new documents and then call the <b>UpdateDocument</b> interface method for updating the values.

 ["Everest Form" on page 62](#)



## EverestDocument.AddCustomPage

### Purpose

Use this method to add your custom forms to any **Everest** profile.

### Syntax

```
function AddCustomPage(const sCaption,  
    sObjectId: WideString): WordBool;
```

### Parameters

#### *sCaption*

This is the caption for the newly created custom page.

#### *sObjectID*

The is the object ID of your custom form.

### Return Value

This method returns True or False based on success.



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## EverestDocument.UpdateDocument

### Purpose

Use this method to modify documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents. To create new documents, call the **ExecuteAction** method with a menu item for creating new documents and then call the **UpdateDocument** interface method for updating the values.

### Syntax

```
function UpdateDocument(const sDocumentXML: WideString): Smallint;
```

### Parameters

#### *sDocumentXML*

This is the XML data to be passed. Refer to the SalesDocument and PurchaseDocument schemas to code this information.



The SalesDocument and PurchaseDocument Schemas for **Everest Add-Ins** are the same schemas that are used for **Everest API**.



["SalesDocument Schema" on page 146](#)

["PurchaseDocument Schema" on page 144](#)

For XML Samples that show the XML format for the UpdateDocument method, refer to the following samples:

- Update Sales Document XML Sample
- Update Purchase Document XML Sample



["Update Sales Document XML Sample" on page 109](#)

["Update Purchase Document XML Sample" on page 111](#)

### Return Value

If the process is successful then the method returns 0. If there is a generic error or exception, the return value is -1. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## Update Sales Document XML Sample

```

- <SalesDocument>
  <CustomerCode>100101</CustomerCode>
  <ContactCode>0</ContactCode>
  <ShipToCode>0</ShipToCode>
  <SalesRepCode>JS</SalesRepCode>
  <PayTermsCode>N30</PayTermsCode>
  <OverrideDueDate>F</OverrideDueDate>
  <DueDate>2006-11-15</DueDate>
  <JurisdictionCode>VAOUT</JurisdictionCode>
  <ShipViaCode>FEDXG</ShipViaCode>
  <DeliveryDate>2006-10-16</DeliveryDate>
  <ExpiryDate />
  <PromotionCode />
  <TaxCalculationType>1</TaxCalculationType>
  <FOB />
  <Reference />
  <CustomerPONo>66243</CustomerPONo>
  <CustomerPODate />
  <LocationSubLocation />
  <Processed />
  <FreightPaddingCode />
  <AllowBackOrderCreation />
  <ShipmentTrackingNo>1111</ShipmentTrackingNo>
  <ShipmentTrackingNote>sample</ShipmentTrackingNote>
  <ExchangeRate />
- <CustomFields>
  <CustomCharacterField Number="1" />
  <CustomCharacterField Number="2" />
  <CustomCharacterField Number="3" />
  <CustomCharacterField Number="4" />
  <CustomDateField Number="1" />
  <CustomDateField Number="2" />
  <CustomDateField Number="3" />
  <CustomDateField Number="4" />
  <CustomLogicalField Number="1" />
  <CustomLogicalField Number="2" />
  <CustomLogicalField Number="3" />
  <CustomLogicalField Number="4" />
  <CustomNumericField Number="1" />
  <CustomNumericField Number="2" />
  <CustomNumericField Number="3" />
  <CustomNumericField Number="4" />
  <CustomMemoField Number="1" />
  <CustomMemoField Number="2" />
  <CustomMemoField Number="3" />
  <CustomMemoField Number="4" />
</CustomFields>

```

Figure 2.55: Update Sales Document XML Sample (Continued on Next Page)

## Update Sales Document XML Sample Continued

```

- <LineItems>
  - <LineItem>
    <Sequence>1</Sequence>
    <ItemCode>BLSOFT66</ItemCode>
    <Description />
    <Quantity>11.000000</Quantity>
    <ItemPrice>11.600000</ItemPrice>
    <UOMCode>EA</UOMCode>
    <CommittedQuantity />
    <Cost>10.000000</Cost>
    <VendorCode>ELEC_LENSES</VendorCode>
    <DiscountValue />
    <DiscountPercent />
    <TaxCode />
    <RevenueAccountCode>41010</RevenueAccountCode>
    <DiscountAccountCode>42010</DiscountAccountCode>
    <ExpenseAccountCode>51010</ExpenseAccountCode>
    <DepartmentCode>CA</DepartmentCode>
    <JobCode />
    <Critical>F</Critical>
    <DependentOn />
    <Dropship />
    <DeliveryDate />
  - <LICustomFields>
    <LICustomCharacterField Number="1" />
    <LICustomCharacterField Number="2" />
    <LICustomCharacterField Number="3" />
    <LICustomCharacterField Number="4" />
    <LICustomDateField Number="1" />
    <LICustomDateField Number="2" />
    <LICustomDateField Number="3" />
    <LICustomDateField Number="4" />
    <LICustomLogicalField Number="1">F</LICustomLogicalField>
    <LICustomLogicalField Number="2">F</LICustomLogicalField>
    <LICustomLogicalField Number="3">F</LICustomLogicalField>
    <LICustomLogicalField Number="4">F</LICustomLogicalField>
    <LICustomNumericField Number="1">.0000</LICustomNumericField>
    <LICustomNumericField Number="2">.0000</LICustomNumericField>
    <LICustomNumericField Number="3">.0000</LICustomNumericField>
    <LICustomNumericField Number="4">.0000</LICustomNumericField>
    <LICustomMemoField Number="1" />
    <LICustomMemoField Number="2" />
    <LICustomMemoField Number="3" />
    <LICustomMemoField Number="4" />
  </LICustomFields>
  </LineItem>
</LineItems>
</SalesDocument>

```

Figure 2.56: Update Sales Document XML Sample Continued

## Update Purchase Document XML Sample

```

- <PurchaseDocument>
  <VendorCode>BIRKEN</VendorCode>
  <ContactCode>0</ContactCode>
  <ShipToCode>0</ShipToCode>
  <PurchaseRepCode>VF</PurchaseRepCode>
  <PayTermsCode>PP</PayTermsCode>
  <OverRideDueDate>F</OverRideDueDate>
  <DueDate>2006-09-06</DueDate>
  <ShipViaCode>UPSG</ShipViaCode>
  <DeliveryDate>2006-10-17</DeliveryDate>
  <ExpiryDate />
  <FOBType>1</FOBType>
  <FOB />
  <Reference />
  <VendorInvoice />
  <VendorInvoiceDate>2006-09-06</VendorInvoiceDate>
  <LocationSubLocation />
  <ShipmentTrackingNo />
  <Processed>F</Processed>
  <TermsFromPIDate>F</TermsFromPIDate>
  <AllowBackOrderCreation>F</AllowBackOrderCreation>
  <HandOff>F</HandOff>
  <DocumentAlias />
  <Notes />
  <ExchangeRate>1.000000</ExchangeRate>
- <CustomFields>
  <CustomCharacterField Number="1" />
  <CustomCharacterField Number="2" />
  <CustomCharacterField Number="3" />
  <CustomCharacterField Number="4" />
  <CustomDateField Number="1" />
  <CustomDateField Number="2" />
  <CustomDateField Number="3" />
  <CustomDateField Number="4" />
  <CustomLogicalField Number="1">F</CustomLogicalField>
  <CustomLogicalField Number="2">F</CustomLogicalField>
  <CustomLogicalField Number="3">F</CustomLogicalField>
  <CustomLogicalField Number="4">F</CustomLogicalField>
  <CustomNumericField Number="1">.0000</CustomNumericField>
  <CustomNumericField Number="2">.0000</CustomNumericField>
  <CustomNumericField Number="3">.0000</CustomNumericField>
  <CustomNumericField Number="4">.0000</CustomNumericField>
  <CustomMemoField Number="1" />
  <CustomMemoField Number="2" />
  <CustomMemoField Number="3" />
  <CustomMemoField Number="4" />
</CustomFields>

```

Figure 2.57: Update Purchase Document XML Sample (Continued on Next Page)

## Update Purchase Document XML Sample Continued

```

- <LineItems>
- <LineItem>
  <Sequence>1</Sequence>
  <ItemCode>4453</ItemCode>
  <Description />
  <Quantity>7.000000</Quantity>
  <Cost>70.000000</Cost>
  <UOMCode>EA</UOMCode>
  <DiscountValue>.000000</DiscountValue>
  <DiscountPercent />
  <TaxCode />
  <RecoverableTax />
  <Account />
  <DepartmentCode>MAIN</DepartmentCode>
  <JobCode />
  <CostExclusiveTax />
  <DeliveryDate />
  <CustomerPO />
  <CustomerPODate />
  <VendorPartNo>200157</VendorPartNo>
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1" />
  <LICustomLogicalField Number="2" />
  <LICustomLogicalField Number="3" />
  <LICustomLogicalField Number="4" />
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>
</LineItems>
</PurchaseDocument>

```

Figure 2.58: Update Purchase Document XML Sample Continued

## Everest Document Line Item

The EverestDocumentLineItem class is a client side interface that contains the following methods:

Method	Description
EverestDocumentLineItem.AddLineItem	Use this method to add line items in documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.
EverestDocumentLineItem.ModifyLineItem	Use this method to make changes to line items in documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.
EverestDocumentLineItem.DeleteLineItem	Use this method to delete line items from documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.



["Everest Document" on page 106](#)

## EverestDocumentLineItem.AddLineItem

### Purpose

Use this method to add line items in documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.

### Syntax

```
function AddLineItem (const sLineItemXML: WideString): Smallint;
```

### Parameters

*sLineItemXML*


This is the XML data to be passed. Refer to the LineItem schema to code this parameter. For an XML sample, refer to the Add Line Item XML Sample.

 ["LineItem Schema" on page 148](#)

### Return Value

This method returns an integer. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form

 ["XML Samples" on page 154](#)  
["Everest Add-In Samples" on page 153](#)

### Add Line Item XML Sample

```
- <LineItems>
- <LineItem>
  <ItemCode>BLSOFT66</ItemCode>
</LineItem>
- <LineItem>
  <ItemCode>102Wheels</ItemCode>
</LineItem>
- <LineItem>
  <ItemCode>CapeFear</ItemCode>
</LineItem>
</LineItems>
```

Figure 2.59: Add Line Item XML Sample



## EverestDocumentLineItem.ModifyLineItem

### Purpose

Use this method to make changes to line items in documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.

### Syntax

```
function ModifyLineItem (const sLineItemXML: WideString): Smallint;
```

### Parameters

#### *sLineItemXML*

This is the XML data to be passed. Refer to the LineItem schema to code this parameter. For an XML sample, refer to the Modify Line Item XML Sample.



[“Modify Line Item XML Sample” on page 116](#)

[“LineItem Schema” on page 148](#)

### Return Value

This method returns an integer. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form



[“XML Samples” on page 154](#)

[“Everest Add-In Samples” on page 153](#)

## Modify Line Item XML Sample

```

- <LineItems>
- <LineItem>
  <Sequence>1</Sequence>
  <ItemCode>BLSOFT66</ItemCode>
  <Description />
  <Quantity>20.000000</Quantity>
  <ItemPrice>25.600000</ItemPrice>
  <UOMCode>12P</UOMCode>
  <CommittedQuantity>10</CommittedQuantity>
  <Cost>20.000000</Cost>
  <VendorCode>ELEC_LENSES</VendorCode>
  <DiscountValue />
  <DiscountPercent />
  <TaxCode />
  <RevenueAccountCode>41010</RevenueAccountCode>
  <DiscountAccountCode>42010</DiscountAccountCode>
  <ExpenseAccountCode>51010</ExpenseAccountCode>
  <DepartmentCode>CA</DepartmentCode>
  <JobCode />
  <Critical>F</Critical>
  <DependentOn />
  <Dropship />
  <DeliveryDate />
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1">F</LICustomLogicalField>
  <LICustomLogicalField Number="2">F</LICustomLogicalField>
  <LICustomLogicalField Number="3">F</LICustomLogicalField>
  <LICustomLogicalField Number="4">F</LICustomLogicalField>
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>

```

Figure 2.60: Modify Line Item XML Sample (Continued on Next Page)

## Modify Line Item XML Sample Continued

```

- <LineItem>
  <Sequence>2</Sequence>
  <ItemCode>101FGTIPS</ItemCode>
  <Description />
  <Quantity>11.000000</Quantity>
  <ItemPrice>11.600000</ItemPrice>
  <UOMCode>EA</UOMCode>
  <CommittedQuantity />
  <Cost>10.000000</Cost>
  <VendorCode>ELEC_LENSSES</VendorCode>
  <DiscountValue />
  <DiscountPercent />
  <TaxCode />
  <RevenueAccountCode>41010</RevenueAccountCode>
  <DiscountAccountCode>42010</DiscountAccountCode>
  <ExpenseAccountCode>51010</ExpenseAccountCode>
  <DepartmentCode>CA</DepartmentCode>
  <JobCode />
  <Critical>F</Critical>
  <DependentOn />
  <Dropship />
  <DeliveryDate />
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1">F</LICustomLogicalField>
  <LICustomLogicalField Number="2">F</LICustomLogicalField>
  <LICustomLogicalField Number="3">F</LICustomLogicalField>
  <LICustomLogicalField Number="4">F</LICustomLogicalField>
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>
</LineItems>

```

**Figure 2.61: Modify Line Item XML Sample Continued**

## EverestDocumentLineItem.DeleteLineItem

### Purpose

Use this method to delete line items from documents. Documents include Sales and Purchase documents as well as Point of Sale (POS) documents.

### Syntax

function DeleteLineItem (const sLineItemSequenceXML: WideString): Smallint;

### Parameters

*sLineItemSequenceXML*


This is the XML data to be passed. Refer to the DeleteLineItem schema to code this parameter. Only **SequenceNo** is required. For an XML sample, refer to the Delete Line Item XML Sample.

 ["DeleteLineItem Schema" on page 150](#)

### Return Value

This method returns an integer. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form

 ["XML Samples" on page 154](#)  
["Everest Add-In Samples" on page 153](#)

### Delete Line Item XML Sample

```
- <LineItems>
  <Sequence>1</Sequence>
  <Sequence>2</Sequence>
  <Sequence>3</Sequence>
  <Sequence>4</Sequence>
</LineItems>
```

Figure 2.62: Delete Line Item XML Sample

## Everest Payment

The EverestPayment class is a client side interface that contains the following methods:

Method	Description
EverestPayment.SetPOSPaymentDetails	Use this method to place data into the controls on the payment details grid of the Payment Details dialog box in Everest POS.
EverestPayment.GetPOSPaymentDetails	Use this method to get the details of POS payments.



["Everest Document" on page 106](#)

## EverestPayment.SetPOSPaymentDetails

### Purpose

Use this method to place data into the controls on the payment details grid of the **Payment Details** dialog box in Everest POS.

The following figure shows the location of the controls on the **Payment Details** dialog box.

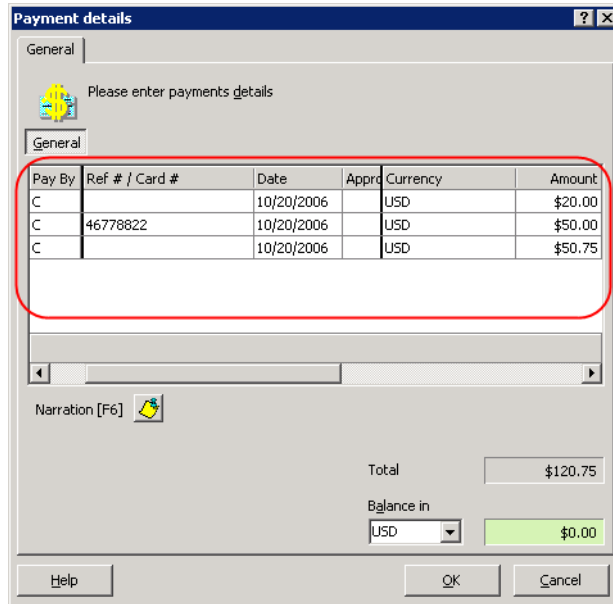


Figure 2.63: Payment Details Dialog Box with Controls Circled

### Syntax

function SetPOSPaymentDetails (const sPaymentData: WideString): Smallint;

### Parameters

#### *sPaymentData*

This is the XML data to be set. Refer to the POSPayment schema to code this parameter. For an XML sample that shows the POS payment details, see the POS Payment XML Sample.



["POS Payment XML Sample" on page 122](#)

["POSPayment Schema" on page 151](#)

## Return Value

This method returns an integer. The following table shows the result descriptions for this method:

Integer	Description
0	Successful Update
-1	Generic Error or Exception
-2	Supplied XML is Not Valid
-3	Current Active Form is Not a Document Form



["XML Samples" on page 154](#)

["Everest Add-In Samples" on page 153](#)

## EverestPayment.GetPOSPaymentDetails

### Purpose

Use this method to get the details of POS payments.

### Syntax

```
function GetPOSPaymentDetails (iSequenceNumber : Integer = 0): WideString;
```

### Parameters

#### *iSequenceNo*

This is an integer representing the sequence number of the POS payment details. This is an optional parameter; when omitted it returns all payment details (all rows from the payment grid).

### Return Value

This method returns XML with POS payment details. For an XML sample that shows the POS payment details, see the POS Payment XML Sample.



[“Add-In Information Dialog Box” on page 125](#)

[“XML Samples” on page 154](#)

[“Everest Add-In Samples” on page 153](#)

## POS Payment XML Sample

```
- <Payments>
- <Payment>
  <Sequence>1</Sequence>
  <PayBy>C</PayBy>
  <ReferenceNo>124344</ReferenceNo>
  <ReferenceDate>2006-10-12</ReferenceDate>
  <Narration>my addin</Narration>
  <ApprovalCode>24234234</ApprovalCode>
  <Amount>100</Amount>
- <CurrencyDetails>
  <PayInCurrencyCode>USD</PayInCurrencyCode>
  <PayInCurrencyExchangeRate>1</PayInCurrencyExchangeRate>
  <PayInAmount>100</PayInAmount>
</CurrencyDetails>
</Payment>
</Payments>
```

Figure 2.64: POS Payment XML Sample



## Creating an Add-In Configuration File

After you create your Add-In DLL file, you will have to create an **Add-In Configuration** file which provides additional information to **Everest** regarding your Add-In. An **Add-In** is technically a COMponent which is defined in a configuration file. The **Add-In Configuration** file is an XML configuration file used to link particular forms with specific methods at specific attach points or hooks. XML is a good way to describe hierarchies and XML definitions are generally extensible.

The **Add-In Configuration** file contains the following information:

- The menu item caption name and path (**Everest Client Menu Hooks**)
- And/or the **Everest** events that trigger actions (**Everest Client Event Hooks**)

The configuration file has an .acf extension. The following is an **Add-In Configuration Example** file:

## Add-In Configuration File Example

```

- <Addin>
  <AddinName>FreightEstimation</AddinName>
  <AddinDescription>This addin is developed using Delphi for freight
  estimation</AddinDescription>
- <Object>
  <ObjectID>FreightEstimation.FreightEstimator</ObjectID>
- <Commands>
  - <Command>
    <CommandID>1</CommandID>
  - <AttachPoints>
    - <MenuItem>
      <Caption>Freight Estimation</Caption>
      <ShortCut>Ctrl+V</ShortCut>
      <GlyphID>BMSEARCH</GlyphID>
      <Form>TfrmSalesDocument</Form>
      <MenuPath />
      <BeginGroup>True</BeginGroup>
    - <ToolBar>
      <Name>Standard</Name>
      <Caption>Standard</Caption>
      <Style>TextOnly</Style>
      <BeginGroup>True</BeginGroup>
    </ToolBar>
  </MenuItem>
  </AttachPoints>
  </Command>
</Commands>
</Object>
<Enabled>True</Enabled>
<Status>Enabled</Status>
</Addin>

```

**Figure 2.65: Add-In Configuration File Example**

Use the **Add-In Configuration File Schema** as a guide to create your Add-In Configuration file.

 [“Add-In Configuration File Schema” on page 142](#)

MenuItems and Events are hooks to the **Everest Client**. You can subscribe to them in the Add-In Configuration file. The **Everest Client** then calls Add-In methods from that hook.

An **Everest Client Event Hook** is the attach point to an **Everest** form, profile, browser, or document event. Supported event hooks that you can use in your Add-In Configuration file are listed in the **Everest Client Event Hooks** section.

 [“Everest Client Event Hooks” on page 133](#)

An **Everest Client Menu Hook** is the attach point to an **Everest** profile, document, or browser menu item. Supported menu hooks that you can use in your Add-In Configuration file are listed in the **Everest Client Menu Hooks** section.

 [“Everest Client Menu Hooks” on page 133](#)



### *Form Caption*

The **Form Caption** field displays the actual name of the browser, profile, or document as displayed in the title bar.

The following tabs are available in the **Add-In Information** dialog box:

### **Menus**

The **Menus** tab displays a list of the applicable menu captions along with the identifiers that can be used in your Add-In Configuration files.

### **Tool Bars**

The **Tool Bars** tab displays a list of the applicable toolbar captions along with the identifiers that can be used in your Add-In Configuration files.

### **Actions**

The **Actions** tab displays a list of identifiers for applicable actions that can be used in your Add-In Configuration files.

### **Events**

The **Events** tab displays a list of Event Names that can be used for the selected form in your Add-In Configuration files.

## Controls

The **Controls** tab shows the list of control names in hierarchical order. Use this tab to copy/paste control names in your code.

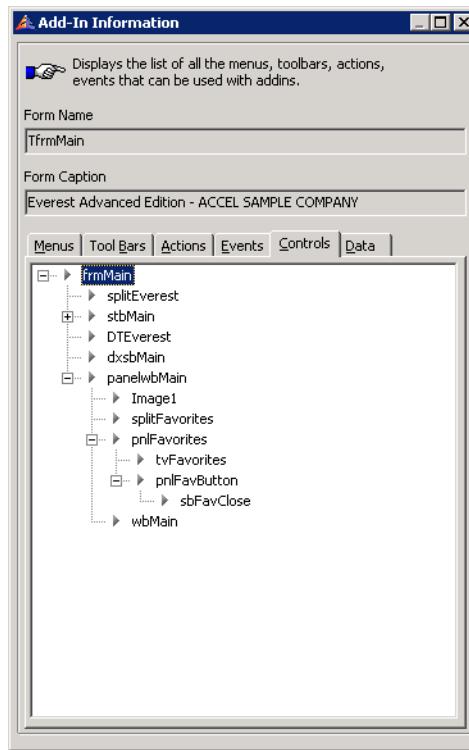


Figure 2.67: Add-In Information Dialog Box - Controls Tab

## Data

The **Data** tab allows you to view the resulting XML data for some of the important methods you can use on the selected form.

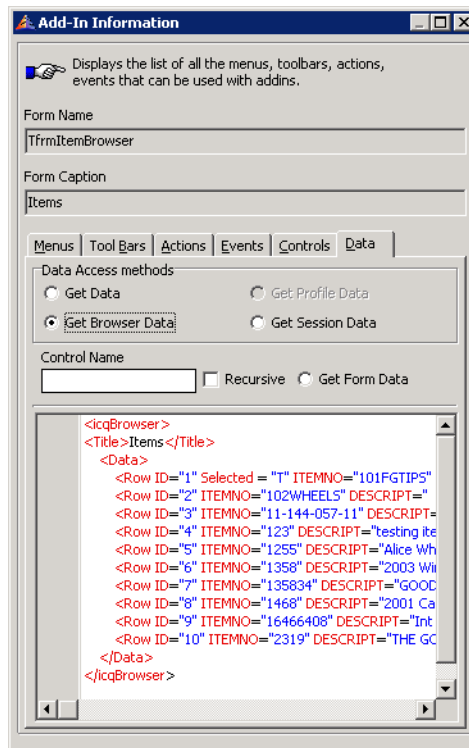


Figure 2.68: Add-In Information Dialog Box - Data Tab

### Data Access Methods

Use this section to retrieve XML data from the available methods. The following method options are based on the form type. For example, if the form is a profile then the "Get Profile Data" option is enabled. Likewise, if the form is a browser, then the "Get Browser Data" option is enabled.

#### *Get Data*

Select **Get Data** to retrieve the dataset details from any **Everest** form.

#### *Get Browser Data*

Select **Get Browser Data** to retrieve data from an **Everest** browser.

#### *Get Profile Data*

Select **Get Profile Data** to retrieve data from an **Everest** profile.

#### *Get Session Data*

Select **Get Session Data** to retrieve **Everest** session data.

### **Control Name**

Use this section to retrieve form data for controls.

#### ***Control Name***

To view data for a single control, enter the name of a control from the selected form in the **Control Name** field and then select **Get Form Data** to retrieve the form data for that control. To view all of the form controls, leave this optional field blank.

#### ***Recursive***

Select the **Recursive** checkbox before you select **Get Form Data** if you want to retrieve data on child controls.

#### ***Get Form Data***

Select this option to retrieve form data for controls. If you enter a control name in the **Control Name** field, the grid only displays data for that control. If you leave the **Control Name** field blank, the grid displays form data for the parent control on the form. If you select the **Recursive** checkbox, you will also see data for the child controls.

#### ***Example***

A programmer wants to view form data for the **pnlsummary** control along with its child controls. For the **Everest Payment Details** dialog box (POS payment details), access the **Add-In Information** dialog box and click the **Data** tab. In the **Control Name** field, enter **pnlsummary**, select **Recursive**, and then select **Get Form Data** to view the form data for that control and its child controls. The following figure shows a child control (lblTotal) circled.

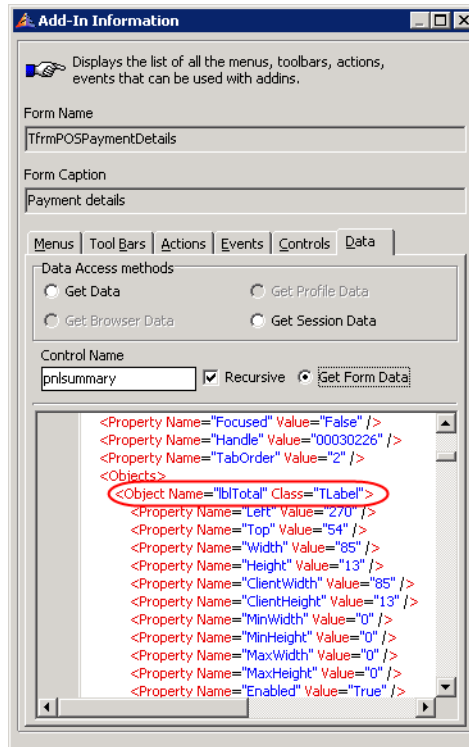


Figure 2.69: Add-In Information Dialog Box - Child Control Circled



## Installing Your Add-Ins

After you create your Add-In and Add-In Configuration file, place your Add-In Configuration file in the **Addins** directory in the following location:

```
..\Icode\Everest\Client\Addins
```

**Example** C:\Program Files\Icode\Everest\Client\Addins

If you do not see the “Addins” folder in the Client directory, you must create it.

Place your DLL file in the **Addins** directory also. If your DLL file is in another location on the system it must be registered through the COM registration utility. Restart **Everest** on the workstation to validate and register the add-ins.

### Migrating to Microsoft Windows Vista

When migrating to the Microsoft Windows® Vista operating system, you must manually move your **Everest** Add-Ins to the <ProgramData>\Everest\Addins folder.

**Example** C:\ProgramData\Everest\Addins

## Add-In Validation

**Add-In Validation** is completed upon **Everest** startup by the system to determine if an Add-In will operate correctly with the **Everest Client**. During the **Add-In Validation** process, the **Everest Client** host application performs the following checks:

- Reads the Add-In Configuration file and checks to make sure that the DLL is a COM DLL
- Instantiates the object to ensure that the add-in is responding
- Reports any errors to the user

## Add-In Registration

**Add-In Registration** is the process of registering the add-in with **Everest** on the local machine. This process runs under the currently logged in windows user upon **Everest** startup.

In the **Add-In Registration** process, the following actions are completed by the **Everest Client** host application:

- List all add-ins by polling the Add-In Configuration files in the following folders:
  - ..\Icode\Everest\Client\Addins folder
  - <ProgramData>\Everest\Addins folder
- Validate details from the Add-In Configuration file

- Register subscribed attach points
  - Registered add-ins display in the **Add-In Manager** utility.

 [“Add-In Manager Utility” on page 170](#)

Based on the configuration in the Add-In Configuration file, the currently logged in user is queried whether or not to enable the new add-in, or the add-in will be enabled automatically. The add-in can also be disabled automatically, depending upon the add-in configuration.

## Add-In Execution

In the **Add-In Execution** process, the following actions are completed by the **Everest Client** host application for each registered add-in:

- On Form Create
  - Instantiate the addin object (Calls “OnCreate”)
  - Load menu items for the add-in
- On Menu Item Click or On any subscribed Event
  - Execute the method (Calls “Execute”)
- On Form Destroy
  - Destroy the addin object (Calls “OnDestroy”)

## Everest Client Menu Hooks

An **Everest Client Menu Hook** is the attach point to an **Everest** profile, document, or browser menu item. Profiles, documents, or browsers can be identified based on their class names. Determine the profile, document, or browser to use with your Add-In and access the **Add-In Information** dialog box from within the profile, document, or browser to obtain the required information. The **Add-In Information** dialog box provides a detailed list of all of the menus, toolbars, actions, and events that can be used with add-ins for a specific form. [Path: In **Everest**, open the form (browser, document, or profile) and press the **Ctrl + Shift + F11** keys.]

 [“Add-In Information Dialog Box” on page 125](#)

## Everest Client Event Hooks

An **Everest Client Event Hook** is the attach point to an **Everest** form, profile, browser, or document event. All event calls are synchronous, that is, something has to happen before proceeding. The following table shows the possible events that are applicable to all supported forms or just profiles, browsers, or documents. Events applicable to “Documents Only” pertain to Sales and Purchase documents and include Point of Sale (POS) documents.

For a detailed list of all of the menus, toolbars, actions, and events that can be used with add-ins for a specific form, review the **Add-In Information** dialog box for that form. [Path: In **Everest**, open the form (browser, profile, or document) and press the **Ctrl + Shift + F11** keys.]

 [“Add-In Information Dialog Box” on page 125](#)

For more information, see the **Creating an Add-In Configuration File** section.

 [“Creating an Add-In Configuration File” on page 123](#)

Everest Client Event Hooks			
Event Name (Used in the Add-In Configuration file)	Applicable To	Event Data	Can Stop Further Processing ?
OnActivate	All Supported Forms	N/A	No
OnFormClick	All Supported Forms	N/A	No
OnClose	All Supported Forms	N/A	No
OnCloseQuery	All Supported Forms	N/A	Yes
OnContextPopup	All Supported Forms	N/A	No
OnCreate	All Supported Forms	N/A	No
OnFormDbClick	All Supported Forms	N/A	No
OnDeactivate	All Supported Forms	N/A	No
OnFormDragDrop	All Supported Forms	Source, x and y coordinate	No
OnFormDragOver	All Supported Forms	N/A	No
OnHide	All Supported Forms	N/A	No

Everest Client Event Hooks			
Event Name (Used in the Add-In Configuration file)	Applicable To	Event Data	Can Stop Further Processing ?
OnFormKeyDown	All Supported Forms	<Instance> <Key></Key> <ShiftStatus></ShiftStatus> </Instance>	No
OnFormKeyPress	All Supported Forms	<Instance> <Key></Key> </Instance>	No
OnFormKeyUp	All Supported Forms	<Instance> <Key></Key> <ShiftStatus></ShiftStatus> </Instance>	No
OnFormMouseDown	All Supported Forms	<Instance> <XCoordinate></XCoordinate> <YCoordinate></YCoordinate> <ShiftKeyStatus></ShiftKeyStatus> </Instance>	No
OnFormMouseMove	All Supported Forms	<Instance> <XCoordinate></XCoordinate> <YCoordinate></YCoordinate> <ShiftKeyStatus></ShiftKeyStatus> </Instance>	No
OnFormMouseUp	All Supported Forms	<Instance> <XCoordinate></XCoordinate> <YCoordinate></YCoordinate> <ShiftKeyStatus></ShiftKeyStatus> </Instance>	No
OnFormMouseWheel	All Supported Forms	N/A	No
OnFormMouseWheelDown	All Supported Forms	N/A	No

Everest Client Event Hooks			
Event Name (Used in the Add-In Configuration file)	Applicable To	Event Data	Can Stop Further Processing ?
OnFormMouseWheelUp	All Supported Forms	N/A	No
OnResize	All Supported Forms	N/A	No
OnShow	All Supported Forms	N/A	No
OnMenuClick	All Supported Forms	<Instance> <DocumentType></DocumentType> <Name></Name> <ClassName></ClassName> <Caption></Caption> <Description></Description> <Category></Category> <BarManagerName></BarManagerName> <Index></Index> <LinkCount></LinkCount> <Enabled></Enabled> <Tag></Tag> <OwnerName></OwnerName> </Instance>	Yes
OnEnter	All Supported Forms	N/A	No
OnButtonClick	All Supported Forms	<Instance> <Name></Name> <ClassName></ClassName> </Instance>	Yes
OnMinimize	All Supported Forms	N/A	No

Everest Client Event Hooks			
Event Name (Used in the Add-In Configuration file)	Applicable To	Event Data	Can Stop Further Processing ?
OnMove	All Supported Forms	N/A	No
OnLoad	Profiles Only	N/A	No
OnBeforeSave	Profiles Only	N/A	Yes
OnAfterSave	Profiles Only	N/A	No
OnPageChange	Profiles Only	N/A	No
OnDataChange	Profiles Only	N/A	No
OnRowBeforeChange	Browsers Only	N/A	No
OnRowAfterChange	Browsers Only	N/A	No
OnModifyLineItem	Documents Only*	<Instance> <DocumentType></DocumentType> <Name></Name> <SequenceNo></SequenceNo> <OldValue></OldValue> <NewValue></NewValue> </Instance>	No
OnBeforeSaveDocument	Documents Only*	<Instance> <DocumentType></DocumentType> </Instance>	Yes
OnAfterSaveDocument	Documents Only*	<Instance> <DocumentType></DocumentType> </Instance>	No
OnStatusBarClick	Documents Only*	<Instance> <DocumentType></DocumentType> <PanelName></PanelName> <PanelText></PanelText> </Instance>	Yes

Everest Client Event Hooks			
Event Name (Used in the Add-In Configuration file)	Applicable To	Event Data	Can Stop Further Processing ?
OnBeforeDeleteLineItem	Documents Only*	<Instance> <DocumentType></DocumentType> <ItemCode></ItemCode> <SequenceNo></SequenceNo> </Instance>	Yes
OnAfterDeleteLineItem	Documents Only*	<Instance> <DocumentType></DocumentType> <ItemCode></ItemCode> <SequenceNo></SequenceNo> </Instance>	No
OnScrollLineItem	Documents Only*	<Instance> <DocumentType></DocumentType> <SequenceNo></SequenceNo> <ItemCode></ItemCode> </Instance>	No
OnInsertLineItem	Documents Only*	<Instance> <DocumentType></DocumentType> </Instance>	Yes
OnDocumentDataChange	Documents Only*	<Instance> <DocumentType></DocumentType> <FieldName></FieldName> <NewValue></NewValue> <OldValue></OldValue> </Instance>	No
OnModifyFieldData	All Supported Forms	<Instance> <Name></Name> <OldValue></OldValue> <NewValue></NewValue> </Instance>	No

\* Events applicable to "Documents Only" pertain to Sales and Purchase documents and include Point of Sale (POS) documents. Event data for documents is XML formatted data supplied as "WideString;".



Shift Status = ssCtrl, ssShift, ssAlt , ssLeft, ssRight, ssMiddle, ssDouble; Key = User Entered character



## 3. Everest Add-In Schemas

---

---

Everest Add-In Schemas

Everest Add-In Samples

XML Samples



## Everest Add-In Schemas

**Everest Add-In Schemas** are examples you can use to create your Add-In Configuration Files and to guide you through coding your method parameters. The following list shows available schema examples:

- **Add-In Configuration File Schema**
- **SessionData Schema** (This schema is for the sSessionDataXML parameter of the SetData method.)
- **EverestForm.ExecuteAction Schema** (This schema is for the sCommand parameter of the ExecuteAction method.)
- **sControlData Schema** (This schema is for the sControlData parameter of the SetFormData method.)
- **PurchaseDocument Schema** (This schema is for the sDocumentXML parameter of the UpdateDocument method.)
- **SalesDocument Schema** (This schema is for the sDocumentXML parameter of the UpdateDocument method.)
- **LineItem Schema** (This schema is for the sLineItemXML parameter of the AddLineItem and ModifyLineItem methods.)
- **DeleteLineItem Schema** (This schema is for the sLineItemSequenceXML parameter of the DeleteLineItem method.)
- **POSPayment Schema** (This schema is for the sPaymentData parameter of the SetPOSPaymentDetails method.)



[“EverestSession.SetData” on page 101](#)

[“EverestForm.ExecuteAction” on page 72](#)

[“EverestForm.SetFormData” on page 82](#)

[“EverestForm.UpdateDocument” on page 85](#)

[“EverestForm.AddLineItem” on page 90](#)

[“EverestForm.ModifyLineItem” on page 92](#)

[“EverestForm.DeleteLineItem” on page 95](#)

[“EverestForm.SetPOSPaymentDetails” on page 96](#)



The SalesDocument and PurchaseDocument Schemas for **Everest Add-Ins** are the same schemas that are used for **Everest API**.



**Everest Software** provides schema examples for clarity and understanding. Because we are constantly improving the **Everest Add-Ins**, schemas may change. You should always consult the latest **Everest Service Pack Release Notes** for updates.

## Add-In Configuration File Schema

Use the **Add-In Configuration File Schema** as a guide to create your Add-In Configuration file. See the **Everest Add-Ins** help files to view the Add-In Configuration File Schema.

## SessionData Schema

Use the **SessionData Schema** as a guide to code your sSessionData XML parameter of the EverestSession.SetData method. See the **Everest Add-Ins** help files to view the SessionData Schema.

 [“EverestSession.SetData” on page 101](#)

## EverestForm.ExecuteAction Schema

Use the **EverestForm.ExecuteAction Schema** as a guide to code your sCommand parameter for the EverestForm.ExecuteAction method.

```
<?xml version="1.0" encoding="utf-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" xmlns:edt="urn:edatatypes">
  <xs:import namespace="urn:edatatypes" />
- <xs:annotation>
  <xs:documentation>XML schema for Everest Add-In actions. Copyright
  (c) 2007 Everest Inc. All rights reserved. Date : 10 May
  2007</xs:documentation>
</xs:annotation>
- <xs:element name="Data">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="Form" type="edt:T_VARCHAR_255"
  minOccurs="1" maxOccurs="1" />
- <xs:element name="Params" minOccurs="0" maxOccurs="1">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="Param"
  type="edt:T_VARCHAR_255" minOccurs="0"
  maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Figure 3.1: EverestForm.ExecuteAction Schema

 [“EverestForm.ExecuteAction” on page 72](#)

## sControlData Schema

Use the **sControlData Schema** as a guide to code your sControlData parameters for the EverestForm.SetFormData method.

```

<?xml version="1.0" encoding="utf-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
- <xs:annotation>
  <xs:documentation>Everest XML schema for Addin Manifest Copyright (c)
    2006 Everest Software, Inc. All rights reserved. Date : 25 May
    2006</xs:documentation>
</xs:annotation>
- <xs:simpleType name="T_VARCHAR_1024">
  - <xs:restriction base="xs:string">
    <xs:maxLength value="1024" />
  </xs:restriction>
</xs:simpleType>
- <xs:simpleType name="T_BOOLEAN">
  - <xs:restriction base="xs:string">
    <xs:maxLength value="5" />
    <xs:enumeration value="True" />
    <xs:enumeration value="False" />
  </xs:restriction>
</xs:simpleType>
- <xs:simpleType name="T_NN_INTEGER">
  - <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
  </xs:restriction>
</xs:simpleType>
- <xs:element name="Control">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element name="Left" type="T_NN_INTEGER" minOccurs="0"
        maxOccurs="1" />
      <xs:element name="Top" type="T_NN_INTEGER" minOccurs="0"
        maxOccurs="1" />
      <xs:element name="Height" type="T_NN_INTEGER" minOccurs="0"
        maxOccurs="1" />
      <xs:element name="Width" type="T_NN_INTEGER" minOccurs="0"
        maxOccurs="1" />
    - <xs:choice minOccurs="0" maxOccurs="1">
      <xs:element name="Text" type="T_VARCHAR_1024" minOccurs="0"
        maxOccurs="1" />
      <xs:element name="Paste" type="T_VARCHAR_1024"
        minOccurs="0" maxOccurs="1" />
    </xs:choice>
    <xs:element name="Visible" type="T_BOOLEAN" minOccurs="0"
      maxOccurs="1" />
    <xs:element name="Focused" type="T_BOOLEAN" minOccurs="0"
      maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="Name" type="T_VARCHAR_1024" />
  </xs:complexType>
</xs:element>
</xs:schema>

```

Figure 3.2: sControlData Schema



“EverestForm.SetFormData” on page 82

## PurchaseDocument Schema

Use the **PurchaseDocument** Schema as a guide to code your sDocumentXML parameter for the EverestForm.UpdateDocument method. This schema is used for Purchase documents.

 “EverestForm.UpdateDocument” on page 85

```

<?xml version="1.0" encoding="utf-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" xmlns:edt="urn:edatatypes"
  xmlns:eli="urn:elineitem">
  <xs:import namespace="urn:edatatypes" />
  <xs:import namespace="urn:elineitem" />
- <xs:annotation>
  <xs:documentation>Everest XML schema for Sales Order Copyright (c) 2005
    Everest Software, Inc. All rights reserved. Author : Johnson Date : 28
    April 2006</xs:documentation>
</xs:annotation>
- <xs:element name="PurchaseDocument">
- <xs:complexType>
  - <xs:choice minOccurs="1" maxOccurs="1">
    - <xs:sequence>
      <xs:element name="DocumentNo" type="edt:T_VARCHAR_15"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="VendorCode" type="edt:T_VARCHAR_15"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="ContactCode" type="edt:T_INTEGER"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="ShipToCode" type="edt:T_INTEGER"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="PurchaseRepCode"
        type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1"
        nillable="true" />
      <xs:element name="PayTermsCode" type="edt:T_VARCHAR_15"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="OverrideDueDate" type="edt:T_BOOLEAN"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="DueDate" type="edt:T_DATE_TIME"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="ShipViaCode" type="edt:T_VARCHAR_15"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="DeliveryDate" type="edt:T_DATE_TIME"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="ExpiryDate" type="edt:T_DATE_TIME"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="FOBType" type="edt:T_FOB_TYPE"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="FOB" type="edt:T_VARCHAR_20"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="Reference" type="edt:T_VARCHAR_20"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="VendorInvoice" type="edt:T_VARCHAR_40"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="VendorInvoiceDate" type="edt:T_DATE_TIME"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="LocationSubLocation"
        type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1"
        nillable="true" />
    
```

Figure 3.3: PurchaseDocument Schema (Continued on Next Page)

## PurchaseDocument Schema Continued

```

<xs:element name="ShipmentTrackingNo"
  type="edt:T_VARCHAR_20" minOccurs="0" maxOccurs="1"
  nillable="true" />
<xs:element name="Processed" type="edt:T_BOOLEAN"
  minOccurs="0" maxOccurs="1" nillable="true" />
<xs:element name="TermsFromPIDate" type="edt:T_BOOLEAN"
  minOccurs="0" maxOccurs="1" nillable="true" />
<xs:element name="AllowBackOrderCreation"
  type="edt:T_BOOLEAN" minOccurs="0" maxOccurs="1" />
<xs:element name="HandOff" type="edt:T_BOOLEAN"
  minOccurs="0" maxOccurs="1" nillable="true" />
<xs:element name="DocumentAlias" type="edt:T_VARCHAR_25"
  minOccurs="0" />
<xs:element name="Notes" type="edt:T_MEMO" minOccurs="0" />
<xs:element name="ExchangeRate"
  type="edt:T_P_DECIMAL_18_6" minOccurs="0" />
<xs:element name="CustomFields" type="edt:T_CUSTOM_FIELD"
  minOccurs="0" />
- <xs:element name="LineItems" minOccurs="0" maxOccurs="1">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="LineItem"
    type="eli:T_PURCHASE_LINE_ITEM" minOccurs="0"
    maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>

```

Figure 3.4: PurchaseDocument Schema (Continued)

## SalesDocument Schema

Use the **SalesDocument** Schema as a guide to code your sDocumentXML parameter for the EverestForm.UpdateDocument method. This schema is used for Sales and Point of Sale documents.

 [“EverestForm.UpdateDocument” on page 85](#)

```
<?xml version="1.0" encoding="utf-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" xmlns:edt="urn:edatatypes"
  xmlns:eli="urn:elineitem">
  <xs:import namespace="urn:edatatypes" />
  <xs:import namespace="urn:elineitem" />
  <xs:annotation>
    <xs:documentation>Everest XML schema for Sales Document Copyright
      (c) 2005 Everest Software, Inc. All rights reserved. Author : Tejas B
      Raval Date : 21 April 2006</xs:documentation>
  </xs:annotation>
  - <xs:element name="SalesDocument">
    - <xs:complexType>
      - <xs:choice maxOccurs="1">
        - <xs:sequence>
          <xs:element name="DocumentNo" type="edt:T_VARCHAR_15"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="CustomerCode"
            type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1"
            nillable="true" />
          <xs:element name="ContactCode" type="edt:T_INTEGER"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="ShipToCode" type="edt:T_INTEGER"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="SalesRepCode"
            type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1"
            nillable="true" />
          <xs:element name="PayTermsCode"
            type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1"
            nillable="true" />
          <xs:element name="OverRideDueDate"
            type="edt:T_BOOLEAN" minOccurs="0" maxOccurs="1"
            nillable="true" />
          <xs:element name="DueDate" type="edt:T_DATE_TIME"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="JurisdictionCode"
            type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1"
            nillable="true" />
          <xs:element name="ShipViaCode" type="edt:T_VARCHAR_15"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="DeliveryDate" type="edt:T_DATE_TIME"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="ExpiryDate" type="edt:T_DATE_TIME"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="PromotionCode"
            type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1"
            nillable="true" />
          <xs:element name="TaxCalculationType"
            type="edt:T_TAX_CALC_TYPE" minOccurs="0"
            maxOccurs="1" nillable="true" />
          <xs:element name="FOB" type="edt:T_VARCHAR_20"
            minOccurs="0" maxOccurs="1" nillable="true" />

```

Figure 3.5: SalesDocument Schema (Continued on Next Page)



## SalesDocument Schema Continued

```

<xs:element name="Reference" type="edt:T_VARCHAR_20"
  minOccurs="0" maxOccurs="1" nillable="true" />
<xs:element name="CustomerPONo"
  type="edt:T_VARCHAR_40" minOccurs="0" maxOccurs="1"
  nillable="true" />
<xs:element name="CustomerPODate"
  type="edt:T_DATE_TIME" minOccurs="0" maxOccurs="1"
  nillable="true" />
<xs:element name="LocationSubLocation"
  type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1"
  nillable="true" />
<xs:element name="Processed" type="edt:T_BOOLEAN"
  minOccurs="0" maxOccurs="1" nillable="true" />
<xs:element name="FreightPaddingCode"
  type="edt:T_FREIGHT_PADDING" minOccurs="0"
  maxOccurs="1" nillable="true" />
<xs:element name="AllowBackOrderCreation"
  type="edt:T_BOOLEAN" minOccurs="0" maxOccurs="1" />
<xs:element name="DocumentAlias"
  type="edt:T_VARCHAR_25" minOccurs="0" />
<xs:element name="ShipmentTrackingNo"
  type="edt:T_VARCHAR_20" minOccurs="0" />
<xs:element name="ShipmentTrackingNote"
  type="edt:T_MEMO" minOccurs="0" />
<xs:element name="ExchangeRate"
  type="edt:T_P_DECIMAL_18_6" minOccurs="0" />
<xs:element name="CustomFields"
  type="edt:T_CUSTOM_FIELD" minOccurs="0" />
- <xs:element name="LineItems" minOccurs="0" maxOccurs="1">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="LineItem"
    type="eli:T_LINE_ITEM" minOccurs="0"
    maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>

```

Figure 3.6: Sales Document Schema (Continued)

## LineItem Schema

Use the **LineItem** Schema as a guide to code your sLineItemXML parameter for the AddLineItem and ModifyLineItem methods.



“EverestForm.AddLineItem” on page 90

“EverestForm.ModifyLineItem” on page 92

```

<?xml version="1.0" encoding="utf-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:edt="urn:edatatypes" targetNamespace="urn:elineitem"
  xmlns="urn:elineitem">
<xs:import namespace="urn:edatatypes" />
- <xs:annotation>
  <xs:documentation>Everest XML schema for Line Item Copyright (c)
    2005 Everest Software, Inc. All rights reserved. Author : Manoj
    Kumar M.C Date : 23 June 2005 Desc : T_LINE_ITEM type is for Line
    item in Everest SequenceNo node defines the line item sequence no.
    LineItem node is line item information the SDK user
    sends.</xs:documentation>
</xs:annotation>
- <xs:complexType name="T_LINE_ITEM">
  - <xs:sequence>
    <xs:element name="Sequence" type="edt:T_SMALLINT"
      minOccurs="0" />
    <xs:element name="ItemCode" type="edt:T_VARCHAR_15"
      minOccurs="0" />
    <xs:element name="Description" type="edt:T_MEMO"
      minOccurs="0" />
    <xs:element name="Quantity" type="edt:T_DECIMAL_18_6"
      minOccurs="0" />
    <xs:element name="ItemPrice" type="edt:T_DECIMAL_18_6"
      minOccurs="0" />
    <xs:element name="UOMCode" type="edt:T_VARCHAR_15"
      minOccurs="0" />
    <xs:element name="CommittedQuantity"
      type="edt:T_DECIMAL_18_6" minOccurs="0" />
    <xs:element name="Cost" type="edt:T_DECIMAL_18_6"
      minOccurs="0" />
    <xs:element name="VendorCode" type="edt:T_VARCHAR_15"
      minOccurs="0" />
    <xs:element name="DiscountValue" type="edt:T_DECIMAL_18_6"
      minOccurs="0" />
    <xs:element name="DiscountPercent"
      type="edt:T_NN_DECIMAL_18_6" minOccurs="0" />
    <xs:element name="TaxCode" type="edt:T_VARCHAR_15"
      minOccurs="0" />
    <xs:element name="RevenueAccountCode"
      type="edt:T_VARCHAR_15" minOccurs="0" />
    <xs:element name="DiscountAccountCode"
      type="edt:T_VARCHAR_15" minOccurs="0" />
    <xs:element name="ExpenseAccountCode"
      type="edt:T_VARCHAR_15" minOccurs="0" />
    <xs:element name="DepartmentCode" type="edt:T_VARCHAR_15"
      minOccurs="0" />
    <xs:element name="JobCode" type="edt:T_VARCHAR_15"
      minOccurs="0" />
    <xs:element name="Critical" type="edt:T_BOOLEAN" minOccurs="0" />
    <xs:element name="DependentOn" type="edt:T_VARCHAR_15"
      minOccurs="0" />

```

Figure 3.7: LineItem Schema (Continued on Next Page)

## LineItem Schema Continued

```

<xs:element name="Dropship" type="edt:T_BOOLEAN"
  minOccurs="0" />
<xs:element name="DeliveryDate" type="edt:T_DATE_TIME"
  minOccurs="0" />
<xs:element name="LICustomFields"
  type="edt:T_LI_CUSTOM_FIELD" minOccurs="0" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name="T_PURCHASE_LINE_ITEM">
- <xs:sequence>
  <xs:element name="Sequence" type="edt:T_SMALLINT"
    minOccurs="0" />
  <xs:element name="ItemCode" type="edt:T_VARCHAR_15"
    minOccurs="0" />
  <xs:element name="Description" type="edt:T_MEMO"
    minOccurs="0" />
  <xs:element name="Quantity" type="edt:T_DECIMAL_18_6"
    minOccurs="0" />
  <xs:element name="Cost" type="edt:T_DECIMAL_18_6"
    minOccurs="0" />
  <xs:element name="UOMCode" type="edt:T_VARCHAR_15"
    minOccurs="0" />
  <xs:element name="DiscountValue" type="edt:T_NN_DECIMAL_18_6"
    minOccurs="0" />
  <xs:element name="DiscountPercent"
    type="edt:T_NN_DECIMAL_18_6" minOccurs="0" />
  <xs:element name="TaxCode" type="edt:T_VARCHAR_15"
    minOccurs="0" />
  <xs:element name="RecoverableTax" type="edt:T_DECIMAL_18_6"
    minOccurs="0" />
  <xs:element name="Account" type="edt:T_VARCHAR_15"
    minOccurs="0" />
  <xs:element name="DepartmentCode" type="edt:T_VARCHAR_15"
    minOccurs="0" />
  <xs:element name="JobCode" type="edt:T_VARCHAR_15"
    minOccurs="0" />
  <xs:element name="CostExclusiveTax" type="edt:T_DECIMAL_18_6"
    minOccurs="0" />
  <xs:element name="DeliveryDate" type="edt:T_DATE_TIME"
    minOccurs="0" />
  <xs:element name="CustomerPO" type="edt:T_VARCHAR_15"
    minOccurs="0" />
  <xs:element name="CustomerPODate" type="edt:T_DATE_TIME"
    minOccurs="0" />
  <xs:element name="VendorPartNo" type="edt:T_VARCHAR_15"
    minOccurs="0" />
  <xs:element name="LICustomFields"
    type="edt:T_LI_CUSTOM_FIELD" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:schema>

```

Figure 3.8: LineItem Schema (Continued)

## DeleteLineItem Schema

Use the **DeleteLineItem** Schema as a guide to code your `sLineItemSequenceXML` parameter of the `DeleteLineItem` method.



["EverestForm.DeleteLineItem" on page 95](#)

```

<?xml version="1.0" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
- <xs:annotation>
  <xs:documentation>XML schema used for DeleteLineItem
  method. Copyright (c) 2006 Everest, Inc. All rights
  reserved. Date : 12 September 2006</xs:documentation>
</xs:annotation>
- <xs:simpleType name="T_SEQUENCE">
- <xs:restriction base="xs:int">
  <xs:minInclusive value="1" />
</xs:restriction>
</xs:simpleType>
- <xs:element name="LineItems">
- <xs:complexType>
  - <xs:sequence>
    <xs:element name="Sequence" type="T_SEQUENCE"
    minOccurs="1" maxOccurs="unbounded"
    nillable="false" />
  </xs:sequence>
</xs:complexType>
- <xs:unique name="UniqueSequence">
  <xs:selector xpath="Sequence" />
  <xs:field xpath="." />
</xs:unique>
</xs:element>
</xs:schema>

```

Figure 3.9: DeleteLineItem Schema

## POSPayment Schema

Use the **POSPayment** Schema as a guide to code your `sPaymentData` parameter of the `SetPOSPaymentDetails` method.

 “[EverestForm.SetPOSPaymentDetails](#)” on page 96

```
<?xml version="1.0" encoding="utf-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" xmlns:edt="urn:edatatypes">
  <xs:import namespace="urn:edatatypes" />
- <xs:annotation>
  <xs:documentation xml:lang="en">Everest XML schema for Reversing a journal
  transaction Copyright (c) 2005 iCode, Inc. All rights
  reserved.</xs:documentation>
</xs:annotation>
- <xs:element name="Payments">
- <xs:complexType>
  - <xs:sequence>
    - <xs:element name="Payment" minOccurs="0" maxOccurs="unbounded">
      - <xs:complexType>
        - <xs:sequence>
          <xs:element name="Sequence" type="edt:T_NN_INTEGER"
            minOccurs="0" maxOccurs="1" />
          <xs:element name="PayBy" type="edt:T_VARCHAR_15"
            minOccurs="0" maxOccurs="1" />
          <xs:element name="ReferenceNo" type="edt:T_VARCHAR_20"
            minOccurs="0" maxOccurs="1" />
          <xs:element name="ReferenceDate" type="edt:T_DATE_TIME"
            minOccurs="0" maxOccurs="1" />
          <xs:element name="Narration" type="edt:T_MEMO"
            minOccurs="0" maxOccurs="1" />
          <xs:element name="ApprovalCode"
            type="edt:T_VARCHAR_15" minOccurs="0" maxOccurs="1" />
          <xs:element name="OverPaymentOptionType"
            type="edt:T_OVER_PAYMENT_TYPES" minOccurs="0"
            maxOccurs="1" />
          <xs:element name="BatchProcessing" type="edt:T_BOOLEAN"
            minOccurs="0" maxOccurs="1" />
          <xs:element name="Amount" type="edt:T_DECIMAL_18_6"
            minOccurs="0" maxOccurs="1" />
          <xs:element name="DiscountApplied"
            type="edt:T_DECIMAL_18_6" minOccurs="0"
            maxOccurs="1" />
        - <xs:element name="CurrencyDetails" minOccurs="0"
            maxOccurs="1">
          - <xs:complexType>
            - <xs:sequence>
              <xs:element name="PayInCurrencyCode"
                type="edt:T_VARCHAR_15" minOccurs="0"
                maxOccurs="1" />
              <xs:element name="PayInCurrencyExchangeRate"
                type="edt:T_DECIMAL_18_6" minOccurs="0"
                maxOccurs="1" />
```

Figure 3.10: POSPayment Schema (Continued on Next Page)

## POSPayment Schema Continued

```
<xs:element name="PayInAmount"  
  type="edt:T_DECIMAL_18_6" minOccurs="0"  
  maxOccurs="1" />  
</xs:sequence>  
</xs:complexType>  
</xs:element>  
</xs:sequence>  
</xs:complexType>  
- <xs:unique name="UniqueSequence">  
  <xs:selector xpath="Sequence" />  
  <xs:field xpath="." />  
</xs:unique>  
</xs:element>  
</xs:sequence>  
</xs:complexType>  
</xs:element>  
</xs:schema>
```

Figure 3.11: POSPayment Schema (Continued)

## Everest Add-In Samples

**Everest Add-In Samples** for different programming languages can be found in the following location:

..\Icode\Everest\Client\Samples\Addins

**Example** C:\Program Files\Icode\Everest\Client\Samples\Addins

A sample file with its corresponding Add-In Configuration file is available for each development tool.

## Working with Sample Add-ins

The following example shows how to effectively utilize the sample add-ins provided with **Everest**. In this example, the Payment Process source files are used.

### Payment Process Sample Add-in Instructions



Follow the instructions below to work with the Payment Process sample add-in:

- Build PaymentProcess.dll using VB.NET with the source files under <ProgramFiles>\icode\Everest\Client\Samples\Addins\PaymentProcess.
- Register the PaymentProcess.dll using RegAsm.exe. For example: "C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\Regasm.exe PaymentProcess.dll /codebase" (Replace v2.0xxxx with the version # of the latest .net framework you have installed.)
- Create a new folder named **Addins** under <ProgramFiles>\icode\Everest\Client.
- Copy PaymentProcess.acf (from <ProgramFiles>\icode\Everest\Client\Samples\Addins\PaymentProcess) to the Addins folder created above.
- Run **Everest** and make sure the Add-in registered successfully.
- Create a new SalesOrder/SalesInvoice in Everest and invoke the Payment dialog.
- Enter a CC payment method and the credit card number and click **OK**.
- This invokes the Add-in dialog for payment processing and subsequently updates the Authorization code in the Payment dialog and proceeds further.

## XML Samples

Samples of XML documents passed or returned with **Everest Add-Ins** are shown in this section. Samples that are too lengthy to be shown here can be viewed in the **Everest Add-Ins** help files.



These XML samples are valid for **Everest 5.0**. **Everest Software** provides XML examples for clarity and understanding. Because we are constantly improving the **Everest Add-Ins**, schemas may change. You should always consult the latest **Everest Service Pack Release Notes** for updates.

### Session Data XML Sample (bExtendedInfo = false)

```
- <SessionData>
  <DBServerName>EVEREST</DBServerName>
  <AppServerName>EVEREST</AppServerName>
  <DatabaseName>EVEREST_SAMPLE</DatabaseName>
  <DBUserName>sa</DBUserName>
  <UserCode>SUP</UserCode>
  <CompanyCode>ASC</CompanyCode>
  <ComputerName>EVEREST</ComputerName>
  <LastErrorCode />
  <LastErrorTimeStamp />
  <SecChkUserCode />
  <LastSecNumber />
  <Department>MAIN</Department>
  <Jurisdiction>VA</Jurisdiction>
  <LoggedInDateTime>08/03/2007 02:24:59 PM</LoggedInDateTime>
  <ReportGlobalRepository>\\EVEREST\REPORTS</ReportGlobalRepository>
  <ReportServer>EVEREST</ReportServer>
  <ReportServerURL>HTTP://EVEREST/reportserver/CRYSTALLIZE.ASP</ReportServerURL>
  <JournalGroup />
  <bExtendedMode>F</bExtendedMode>
  <LastHeartBeatTime>08/03/2007 2:38:48 PM</LastHeartBeatTime>
  <ProductType>1</ProductType>
  <ProxyUserName />
  <TimeZoneOffset>-330</TimeZoneOffset>
  <ExtendedErrorInformation />
  <ProxyDomainName />
  <ExtendedErrorInformation1>2</ExtendedErrorInformation1>
  <LockStatus />
  <EverestDate>08/03/2007</EverestDate>
  <WebClientCode />
  <ACRMServer />
  <RegisteredCompany>ASC</RegisteredCompany>
  <POSCount>0</POSCount>
  <DatabaseType>1</DatabaseType>
  <InternationalMode>0</InternationalMode>
  <TimeZoneIndex IsDayLightSavingEnabled="false">39</TimeZoneIndex>
</SessionData>
```

Figure 3.12: Session Data XML Sample (bExtendedInfo = false)

### Session Data XML Sample (bExtendedInfo = true)

For the **Session Data XML Sample (bExtendedInfo = True)**, see the **Everest Add-Ins** help files.



## Item Profile Form Data XML Sample

For an XML Sample, see the **Item Profile Form Data XML Sample** in the **Everest Add-Ins** help files.

## Browser Data XML Sample (EverestForm.GetBrowserData)

```
- <icqBrowser>
- <Data>
  <Row ID="1" Selected="T" ITEMNO="PTAX" DESCRIPT="+ Tax Difference" ACTIVE="T"
  COST="0" BRAND="" CATEGORY="MISC" MAX_STOCK="0" MIN_NEEDED="" MODEL=""
  QTY_STK="0" TOTAL_STOCK="0" Q_ON_CREDI="0" Q_ON_ORDER="0"
  Q_ON_CREDIT_COMM="0" Q_ON_ORDER_COMM="0" Q_ON_RESERVE_COMM="0"
  Q_ON_RMA_COMM="0" Q_ON_RESER="0" Q_ON_RMA="0" SALE_MEAS="EA"
  SELLPRIC="0" MATRIX_ITEM_TYPE="0" TYPE="2" ENTRY_TYPE="1" PURC_FRAC="0"
  AVG_COST="0" AVAILABLE="0" AVG_STOCK="0" SERIALIZE="F" SALEFRACTION="1"
  MARKUP="" PURC_MEAS="EA" MANUCODE="" FF_ACTIVE="F" FF="0" INVENTORED="F"
  AUTOSERIAL="F" FRACTION="1" MPRICE="" STOCK_MEAS="EA" LAST_COST="0"
  LASTADDONCOST="0" P_PURC_FRAC="0" S_PURC_FRAC="0"
  SALEPRICE_DEC_ACCURACY="2" PURCPRIE_DEC_ACCURACY="2"
  PURCFRACTION="1" />
</Data>
</icqBrowser>
```

Figure 3.13: Browser Data XML Sample (EverestForm.GetBrowserData)

## EverestForm.Browser.GetData XML Sample (bSelected = True)

(When executed on the Customer browser)

```

- <icqBrowser>
  <Title>Customers</Title>
- <Header>
  - <Focused>
    <Row>1</Row>
    <Column Name="ADDRESS_LINE1">547 Franklin Street</Column>
  </Focused>
  - <Search>
    <Column Name="ACCOUNT_NAME">Allied Systems</Column>
  </Search>
  </Header>
- <Data>
  <Row ID="1" Selected="T" ACTIVE="T" CURR_CODE="USD" CUST_CODE="100100"
  SALESPERSO="MB" CONTCODE="" ADDR_CODE="12" ACCOUNT_NAME="Allied
  Systems" DEP_CODE="MAIN" FIRST_NAME="Rob" LAST_NAME="Broido"
  ETEL1="5403422507" ETEL2="" EFAX="" EMAIL="rob@allied.com"
  STREET_ADDRESS="547 Franklin Street" CITY="ROANOKE" STATE="VA"
  COUNTRY="UNITED STATES" ZIP="24007" TYPE="4" SHIPCODE=""
  ADDR_CODE_1="12" HAND_OFF="0" CURRCODE="USD"
  BASENETBALANCE="12180.93" FRXNETBALANCE="12180.93" REQUIREPO="F"
  ADDRESS_LINE1="547 Franklin Street" ADDRESS_LINE2=""
  FULL_ADDRESS="Allied Systems Rob Broido 547 Franklin Street ROANOKE,
  VA 24007" BILLCODE="12" TITLE="" CATEGORY_CODE="" MOBILE_PHONE=""
  PAGER_NO="" URL="" TEL1="5403422507" TEL2="" EXTN_TEL1="" EXTN_TEL2=""
  FAX="" EXTN_FAX="" COMPANY_NAME="Allied Systems" IS_LOCKED="F"
  APPLY_MASK_TEL="T" APPLY_MASK_MOBILE="T" APPLY_MASK_POSTAL="T"
  CUSTCHAR1="Silver" CUSTCHAR2="" CUSTCHAR3="" CUSTCHAR4=""
  CUSTDATE1="11/2/2006" CUSTDATE2="" CUSTDATE3="" CUSTDATE4=""
  CUSTLOG1="T" CUSTLOG2="F" CUSTLOG3="F" CUSTLOG4="F" CUSTMEMO1=""
  CUSTMEMO2="" CUSTMEMO3="" CUSTMEMO4="" CUSTNUM1="0" CUSTNUM2="0"
  CUSTNUM3="0" CUSTNUM4="0" />
  </Data>
</icqBrowser>

```

Figure 3.14: EverestForm.Browser.GetData XML Sample (bSelected = True)

## EverestForm.Browsers.GetData XML Sample (bSelected = False)

For EverestForm.Browser.GetData XML Sample (bSelected = False), see the Everest Add-Ins help files.

## Profile Data XML Sample (EverestForm.GetProfileData)

```

- <ProfileData>
- <quProfile>
- <Data>
  <Row ID="1" ACTIVE="T" AUTOSERIAL="F" ITEMNO="4453" CATEGORY="France"
  DESCRIPT="1994 Pessac-Leognan Blanc" Q_ON_ORDER="0" Q_ON_RESER="1"
  Q_ON_RMA="10" Q_ON_CREDI="0" MIN_NEEDED="0" COST="70" AVG_COST="70"
  FF_ACTIVE="F" FF="0" TAXABLE="F" MDESC="" SELL_DESC="" S_EXP_DESC=""
  P_EXP_DESC="" BRAND="" MODEL="" SERIALIZE="F" INVENTORED="T" DEP_CODE=""
  PURC_MEAS="EA" SALE_MEAS="EA" PGT_1="" PGT_2="" PGT_3="" PGT_4=""
  PGT_5="" COMMISSION="T" LSI_DOC="300220" LSI_DATE="10/16/2006"
  LSI_AMT="588" LCM_DOC="600100" LCM_DATE="5/31/2006" LCM_AMT="84"
  LPR_DOC="" LPR_DATE="" LPR_AMT="" LDM_DOC="" LDM_DATE="" LDM_AMT="0" />
  </Data>
</quProfile>
- <quForm>
  <Data />
</quForm>
- <quSimilarItems>
  <Data />
</quSimilarItems>
</ProfileData>

```

Figure 3.15: Profile Data XML Sample (EverestForm.GetProfileData)

## Profile Data XML Sample (EverestForm.Profile.GetData)

For the Profile Data XML Sample (EverestForm.Profile.GetData), see the Everest Add-Ins help files.

## EverestForm.GetData XML Sample

For the EverestForm.GetData XML Sample, see the Everest Add-Ins help files.

## Update Sales Document XML Sample

```

- <SalesDocument>
  <CustomerCode>100101</CustomerCode>
  <ContactCode>0</ContactCode>
  <ShipToCode>0</ShipToCode>
  <SalesRepCode>JS</SalesRepCode>
  <PayTermsCode>N30</PayTermsCode>
  <OverrideDueDate>F</OverrideDueDate>
  <DueDate>2006-11-15</DueDate>
  <JurisdictionCode>VAOUT</JurisdictionCode>
  <ShipViaCode>FEDXG</ShipViaCode>
  <DeliveryDate>2006-10-16</DeliveryDate>
  <ExpiryDate />
  <PromotionCode />
  <TaxCalculationType>1</TaxCalculationType>
  <FOB />
  <Reference />
  <CustomerPONo>66243</CustomerPONo>
  <CustomerPODate />
  <LocationSubLocation />
  <Processed />
  <FreightPaddingCode />
  <AllowBackOrderCreation />
  <ShipmentTrackingNo>1111</ShipmentTrackingNo>
  <ShipmentTrackingNote>sample</ShipmentTrackingNote>
  <ExchangeRate />
- <CustomFields>
  <CustomCharacterField Number="1" />
  <CustomCharacterField Number="2" />
  <CustomCharacterField Number="3" />
  <CustomCharacterField Number="4" />
  <CustomDateField Number="1" />
  <CustomDateField Number="2" />
  <CustomDateField Number="3" />
  <CustomDateField Number="4" />
  <CustomLogicalField Number="1" />
  <CustomLogicalField Number="2" />
  <CustomLogicalField Number="3" />
  <CustomLogicalField Number="4" />
  <CustomNumericField Number="1" />
  <CustomNumericField Number="2" />
  <CustomNumericField Number="3" />
  <CustomNumericField Number="4" />
  <CustomMemoField Number="1" />
  <CustomMemoField Number="2" />
  <CustomMemoField Number="3" />
  <CustomMemoField Number="4" />
</CustomFields>

```

Figure 3.16: Update Sales Document XML Sample (Continued on Next Page)

## Update Sales Document XML Sample Continued

```

- <LineItems>
- <LineItem>
  <Sequence>1</Sequence>
  <ItemCode>BLSOFT66</ItemCode>
  <Description />
  <Quantity>11.000000</Quantity>
  <ItemPrice>11.600000</ItemPrice>
  <UOMCode>EA</UOMCode>
  <CommittedQuantity />
  <Cost>10.000000</Cost>
  <VendorCode>ELEC_LENSES</VendorCode>
  <DiscountValue />
  <DiscountPercent />
  <TaxCode />
  <RevenueAccountCode>41010</RevenueAccountCode>
  <DiscountAccountCode>42010</DiscountAccountCode>
  <ExpenseAccountCode>51010</ExpenseAccountCode>
  <DepartmentCode>CA</DepartmentCode>
  <JobCode />
  <Critical>F</Critical>
  <DependentOn />
  <Dropship />
  <DeliveryDate />
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1">F</LICustomLogicalField>
  <LICustomLogicalField Number="2">F</LICustomLogicalField>
  <LICustomLogicalField Number="3">F</LICustomLogicalField>
  <LICustomLogicalField Number="4">F</LICustomLogicalField>
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>
</LineItems>
</SalesDocument>

```

Figure 3.17: Update Sales Document XML Sample Continued

## Update Purchase Document XML Sample

```

- <PurchaseDocument>
  <VendorCode>BIRKEN</VendorCode>
  <ContactCode>0</ContactCode>
  <ShipToCode>0</ShipToCode>
  <PurchaseRepCode>VF</PurchaseRepCode>
  <PayTermsCode>PP</PayTermsCode>
  <OverRideDueDate>F</OverRideDueDate>
  <DueDate>2006-09-06</DueDate>
  <ShipViaCode>UPSG</ShipViaCode>
  <DeliveryDate>2006-10-17</DeliveryDate>
  <ExpiryDate />
  <FOBType>1</FOBType>
  <FOB />
  <Reference />
  <VendorInvoice />
  <VendorInvoiceDate>2006-09-06</VendorInvoiceDate>
  <LocationSubLocation />
  <ShipmentTrackingNo />
  <Processed>F</Processed>
  <TermsFromPIDate>F</TermsFromPIDate>
  <AllowBackOrderCreation>F</AllowBackOrderCreation>
  <HandOff>F</HandOff>
  <DocumentAlias />
  <Notes />
  <ExchangeRate>1.000000</ExchangeRate>
- <CustomFields>
  <CustomCharacterField Number="1" />
  <CustomCharacterField Number="2" />
  <CustomCharacterField Number="3" />
  <CustomCharacterField Number="4" />
  <CustomDateField Number="1" />
  <CustomDateField Number="2" />
  <CustomDateField Number="3" />
  <CustomDateField Number="4" />
  <CustomLogicalField Number="1">F</CustomLogicalField>
  <CustomLogicalField Number="2">F</CustomLogicalField>
  <CustomLogicalField Number="3">F</CustomLogicalField>
  <CustomLogicalField Number="4">F</CustomLogicalField>
  <CustomNumericField Number="1">.0000</CustomNumericField>
  <CustomNumericField Number="2">.0000</CustomNumericField>
  <CustomNumericField Number="3">.0000</CustomNumericField>
  <CustomNumericField Number="4">.0000</CustomNumericField>
  <CustomMemoField Number="1" />
  <CustomMemoField Number="2" />
  <CustomMemoField Number="3" />
  <CustomMemoField Number="4" />
</CustomFields>

```

Figure 3.18: Update Purchase Document XML Sample (Continued on Next Page)

## Update Purchase Document XML Sample Continued

```

- <LineItems>
- <LineItem>
  <Sequence>1</Sequence>
  <ItemCode>4453</ItemCode>
  <Description />
  <Quantity>7.000000</Quantity>
  <Cost>70.000000</Cost>
  <UOMCode>EA</UOMCode>
  <DiscountValue>.000000</DiscountValue>
  <DiscountPercent />
  <TaxCode />
  <RecoverableTax />
  <Account />
  <DepartmentCode>MAIN</DepartmentCode>
  <JobCode />
  <CostExclusiveTax />
  <DeliveryDate />
  <CustomerPO />
  <CustomerPODate />
  <VendorPartNo>200157</VendorPartNo>
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1" />
  <LICustomLogicalField Number="2" />
  <LICustomLogicalField Number="3" />
  <LICustomLogicalField Number="4" />
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>
</LineItems>
</PurchaseDocument>

```

Figure 3.19: Update Purchase Document XML Sample Continued

## Add Line Item XML Sample

```
- <LineItems>
- <LineItem>
  <ItemCode>BLSOFT66</ItemCode>
</LineItem>
- <LineItem>
  <ItemCode>102Wheels</ItemCode>
</LineItem>
- <LineItem>
  <ItemCode>CapeFear</ItemCode>
</LineItem>
</LineItems>
```

Figure 3.20: Add Line Item XML Sample



## Modify Line Item XML Sample

```

- <LineItems>
- <LineItem>
  <Sequence>1</Sequence>
  <ItemCode>BLSOFT66</ItemCode>
  <Description />
  <Quantity>20.000000</Quantity>
  <ItemPrice>25.600000</ItemPrice>
  <UOMCode>12P</UOMCode>
  <CommittedQuantity>10</CommittedQuantity>
  <Cost>20.000000</Cost>
  <VendorCode>ELEC_LENSES</VendorCode>
  <DiscountValue />
  <DiscountPercent />
  <TaxCode />
  <RevenueAccountCode>41010</RevenueAccountCode>
  <DiscountAccountCode>42010</DiscountAccountCode>
  <ExpenseAccountCode>51010</ExpenseAccountCode>
  <DepartmentCode>CA</DepartmentCode>
  <JobCode />
  <Critical>F</Critical>
  <DependentOn />
  <Dropship />
  <DeliveryDate />
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1">F</LICustomLogicalField>
  <LICustomLogicalField Number="2">F</LICustomLogicalField>
  <LICustomLogicalField Number="3">F</LICustomLogicalField>
  <LICustomLogicalField Number="4">F</LICustomLogicalField>
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>

```

Figure 3.21: Modify Line Item XML Sample (Continued on Next Page)

## Modify Line Item XML Sample Continued

```

- <LineItem>
  <Sequence>2</Sequence>
  <ItemCode>101FGTIPS</ItemCode>
  <Description />
  <Quantity>11.000000</Quantity>
  <ItemPrice>11.600000</ItemPrice>
  <UOMCode>EA</UOMCode>
  <CommittedQuantity />
  <Cost>10.000000</Cost>
  <VendorCode>ELEC_LENSSES</VendorCode>
  <DiscountValue />
  <DiscountPercent />
  <TaxCode />
  <RevenueAccountCode>41010</RevenueAccountCode>
  <DiscountAccountCode>42010</DiscountAccountCode>
  <ExpenseAccountCode>51010</ExpenseAccountCode>
  <DepartmentCode>CA</DepartmentCode>
  <JobCode />
  <Critical>F</Critical>
  <DependentOn />
  <Dropship />
  <DeliveryDate />
- <LICustomFields>
  <LICustomCharacterField Number="1" />
  <LICustomCharacterField Number="2" />
  <LICustomCharacterField Number="3" />
  <LICustomCharacterField Number="4" />
  <LICustomDateField Number="1" />
  <LICustomDateField Number="2" />
  <LICustomDateField Number="3" />
  <LICustomDateField Number="4" />
  <LICustomLogicalField Number="1">F</LICustomLogicalField>
  <LICustomLogicalField Number="2">F</LICustomLogicalField>
  <LICustomLogicalField Number="3">F</LICustomLogicalField>
  <LICustomLogicalField Number="4">F</LICustomLogicalField>
  <LICustomNumericField Number="1">.0000</LICustomNumericField>
  <LICustomNumericField Number="2">.0000</LICustomNumericField>
  <LICustomNumericField Number="3">.0000</LICustomNumericField>
  <LICustomNumericField Number="4">.0000</LICustomNumericField>
  <LICustomMemoField Number="1" />
  <LICustomMemoField Number="2" />
  <LICustomMemoField Number="3" />
  <LICustomMemoField Number="4" />
</LICustomFields>
</LineItem>
</LineItems>

```

**Figure 3.22: Modify Line Item XML Sample Continued**

## Delete Line Item XML Sample

```
- <LineItems>
  <Sequence>1</Sequence>
  <Sequence>2</Sequence>
  <Sequence>3</Sequence>
  <Sequence>4</Sequence>
</LineItems>
```

Figure 3.23: Delete Line Item XML Sample

## POS Payment XML Sample

```
- <Payments>
- <Payment>
  <Sequence>1</Sequence>
  <PayBy>C</PayBy>
  <ReferenceNo>124344</ReferenceNo>
  <ReferenceDate>2006-10-12</ReferenceDate>
  <Narration>my addin</Narration>
  <ApprovalCode>24234234</ApprovalCode>
  <Amount>100</Amount>
- <CurrencyDetails>
  <PayInCurrencyCode>USD</PayInCurrencyCode>
  <PayInCurrencyExchangeRate>1</PayInCurrencyExchangeRate>
  <PayInAmount>100</PayInAmount>
</CurrencyDetails>
</Payment>
</Payments>
```

Figure 3.24: POS Payment XML Sample



## 4. Managing Everest Add-Ins

---

---

Add-Ins

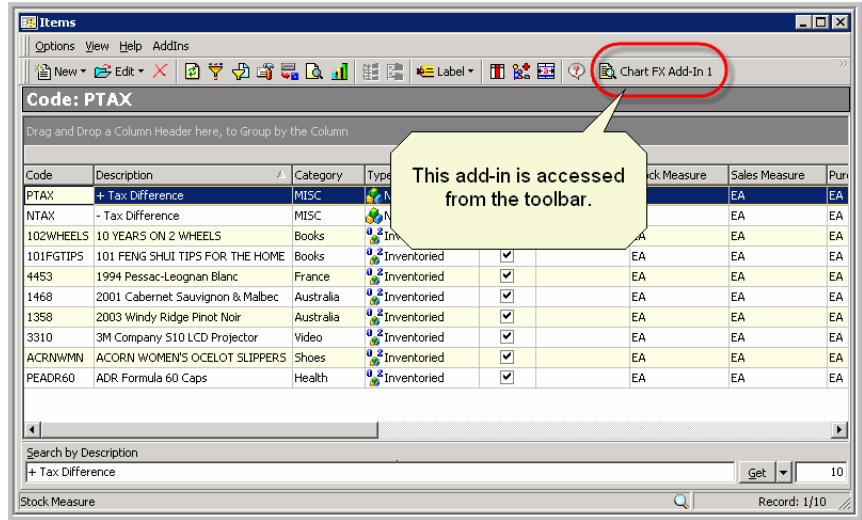
Add-In Manager Utility



## Add-Ins

An **Add-In** is a supplemental program that extends the functionality of the main **Everest** application. Usually they are created by third-party programmers to better satisfy the requirements of your business environment. **Everest** browsers, documents and profiles can use add-ins.

**Example** A Chart add-in is added to the Items browser which displays the top ten items when selected from the menu bar. The following figure displays the location of the add-in on the toolbar.



**Figure 4.1: Items Browser with Chart FX Add-In**

In this example, when the user opens the **Items** browser and clicks the **Chart FX Add-In** icon on the toolbar, the add-in executes and a chart displays the top ten items.

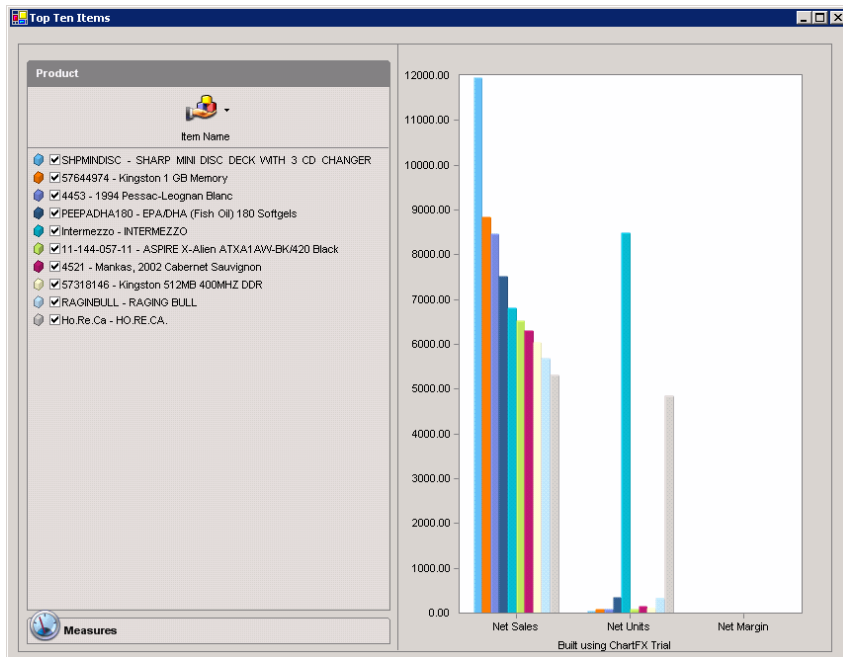


Figure 4.2: Chart FX Add-In Showing the Top Ten Items

## Add-In Manager Utility

Use the **Add-In Manager** utility (path: **Utility > Add-In Manager**) to manage **Everest Add-Ins**. The **Add-In Manager** utility allows you to view all registered add-ins, enable or disable registered add-ins, and remove add-ins. Your add-in provider may also allow you to make additional add-in configurations from this utility. You must have an add-in security right to launch the **Add-In Manager** utility.

**Example** A programmer creates a Debit Card Processing add-in for **Everest** Point of Sale and installs it into **Everest**. The head cashier uses the **Add-In Manager** utility to manage this add-in and enables it on the **Everest** Point of Sale workstations. After the add-in is enabled and a cashier selects the Debit Card payment type, the Debit Card Processing add-in launches and the cashier processes the debit card through the **Everest Add-In**. When the debit card processing is complete, the add-in closes and the cashier continues with the **Everest** Point of Sale application.

## Register Add-Ins

Programmers create add-in configuration files that are added to **Everest**. These configuration files give **Everest** necessary information about the add-in such as where the add-in attaches to **Everest**, where the add-in is located, and whether or not the add-in is enabled by default. When you start **Everest**, these add-in



configuration files are validated by the system. **Everest** then registers the add-in on the workstation. Add-ins must be installed and registered on each workstation where they will be used.

## Manage Registered Add-Ins

You can manage the list of registered **Everest** add-ins on a particular workstation. The add-in list shows the current status of each add-in; descriptions of selected add-ins display in the **Details** section below.

**1** **2** **3** → To access the Add-In Manager utility for a workstation, do the following:

- From the main menu bar of the appropriate workstation, select **Utility > Add-In Manager**. You will see the **Add-In Manager** dialog box.

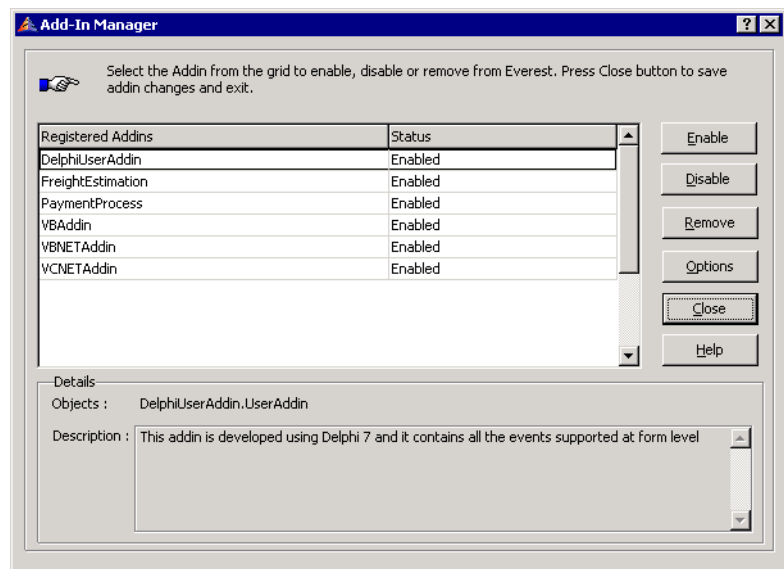


Figure 4.3: Add-In Manager Dialog Box

## The Add-In ManagerDialog Box

Use the **Add-In Manager** dialog box to manage the **Everest Add-Ins** installed and registered on each workstation. You can view, enable, disable, configure, and remove selected add-ins.

The **Add-In Manager** dialog box contains a list of registered add-ins. Each row contains information on one add-in.

### Registered Add-Ins

This column displays a list of add-ins registered on the workstation. The add-in names come from the add-in configuration file.

### Status

This column displays the current status of the add-in. Valid values include the following:

- **Enabled** - The add-in is activated on this machine and is ready for use.
- **Disabled** - The add-in is not in operation on this machine and cannot be used.
- **Not Responding** - There is a problem with this add-in. You can try restarting **Everest** to resolve this issue. If this status does not change after restarting **Everest**, contact your add-in provider for additional assistance.

### Details

#### *Objects*

The top portion of the **Details** section displays the object names from the add-in configuration file of the selected add-in.

#### *Description*

The lower portion of the **Details** section displays the description of the add-in from the add-in configuration file of the selected add-in.

The following buttons are available in the **Add-In Manager** dialog box:

#### **Enable**

Click the **Enable** button to activate the selected add-in on this workstation. An add-in must be enabled before it can be used on the workstation.

#### **Disable**

Click the **Disable** button to turn off the add-in on this workstation. Disable an add-in if you do not want it to be used on this workstation. You can activate it again, if necessary.

#### **Remove**

Click the **Remove** button to remove the add-in from the Add-In Manager list so that it cannot be enabled. The configuration file extension is renamed from ACF to BAK. Do not select this option if you only want to remove the add-in temporarily since the add-in file configuration file extension is renamed. If you want to remove the add-in temporarily, use the **Disable** option instead.

#### **Options**

Click the **Options** button to view any configurations specific to the selected add-in. Your add-in provider may make additional configuration selections available using this option. The **Options** button is disabled when the add-in is disabled.

#### **Close**

Click the **Close** button to exit the **Add-In Manager** utility.

## View Registered Add-ins

Installed and validated add-ins register upon **Everest** startup. You can view the list of registered add-ins on a workstation.



To view all registered add-ins on a workstation, do the following:

- From the main menu bar of the appropriate workstation, select **Utility > Add-In Manager**. You will see the **Add-In Manager** dialog box.
- Review the list of add-ins and add-in statuses.

- Click the **Close** button to exit the **Add-In Manager** utility without making any changes.

## Enable Registered Add-Ins

Some add-ins are automatically enabled. You can also enable registered add-ins at any time from the **Add-In Manager** utility.



To enable a registered add-in on a workstation, do the following:

- From the main menu bar of the appropriate workstation, select **Utility > Add-In Manager**. You will see the **Add-In Manager** dialog box.
- Select the add-in to be enabled from the list.
- Click the **Enable** button. The add-in status of the selected add-in changes to “Enabled”. The add-in is now available for use on this workstation.
- Click the **Close** button to exit the **Add-In Manager** utility.

## Disable Registered Add-Ins

If you want to prevent users on a workstation from using an add-on, you can disable the add-in. The add-in will not be available for use until it is enabled again.



To disable a registered add-in on a workstation, do the following:

- From the main menu bar of the appropriate workstation, select **Utility > Add-In Manager**. You will see the **Add-In Manager** dialog box.
- Select the add-in to be disabled from the list.
- Click the **Disable** button. The add-in status of the selected add-in changes to “Disabled”. The add-in is no longer available for use on this workstation.
- Click the **Close** button to exit the **Add-In Manager** utility.

## Configure Add-Ins

Some add-in providers allow you to make configurations from the **Add-In Manager** utility. If the **Options** button is enabled for a selected add-in, you can configure it.



To configure a registered add-in on a workstation, do the following:

- From the main menu bar of the appropriate workstation, select **Utility > Add-In Manager**. You will see the **Add-In Manager** dialog box.
- Select the add-in to be configured from the list.
- Click the **Options** button. Make any necessary add-in configurations according to the instructions of the add-in. Contact your add-in provider if you need assistance with these configurations.
- Click the **Close** button to exit the **Add-In Manager** utility.

## Remove Add-Ins

You can completely remove an add-in from a workstation, if necessary. This removes the add-in from the Add-In Manager list and it cannot be enabled. If you need to use this add-in again in the future, it must be reinstalled.



To remove an add-in from the Add-In Manager list, do the following:

- From the main menu bar of the appropriate workstation, select **Utility > Add-In Manager**. You will see the **Add-In Manager** dialog box.
- Select the add-in to be removed from the list.
- Click the **Remove** button. The selected add-in is removed from the list and the add-in configuration file extension is renamed from ACF to BAK.
- Click the **Close** button to exit the **Add-In Manager** utility.

## Installing Add-Ins

Your add-in provider should give you the necessary setup files to install **Everest** add-ins. If you do not receive any setup files, place the Add-in Configuration files (.acf extension) and DLL files in the **Addins** directory in the following location:

..\Icode\Everest\Client\Addins

*Example*

C:\Program Files\Icode\Everest\Client\Addins

If you do not see the “Addins” folder in the Client directory, you must create it.

Restart **Everest** on the workstation to validate and register the add-ins.



For more information, refer to the **Everest\_Add-Ins Help** file located in the ..\Icode\Everest\Client\Help directory. For example, if you installed **Everest** on the C: drive, it is located in the C:\Program Files\Icode\Everest\Client\Help directory.

## Migrating to Microsoft Windows Vista

When migrating to the Microsoft Windows® Vista operating system, you must manually move your **Everest** Add-Ins to the <ProgramData>\Everest\Addins folder.

*Example*

C:\ProgramData\Everest\Addins

# Index

## A

- about Everest Add-Ins 11
- add line item xml sample 91, 114, 162
- add-in
  - creating 19
- add-in configuration file
  - create 123
- add-in configuration file example 124
- add-in configuration file schema 142
- add-in example 169
- add-in execution 132
- add-in information dialog box 125
- add-in interface 58
- add-in manager dialog box 171
- add-in manager utility 170
  - add-in manager dialog box 171
  - configure add-ins 173
  - disable registered add-ins 173
  - enable registered add-ins 173
  - install add-ins 174
  - manage registered add-ins 171
  - register add-ins 170
  - remove add-ins 174
  - view registered add-ins 172
- add-in registration 131
- add-in validation 131
- add-in workflow example 17
- add-ins 169
  - configure 173
  - disable 173
  - enable 173
  - Everest 16
  - install 174
  - manage 171
  - register 170
  - remove 174
  - view 172
  - workflow example 17
- assumptions 5

## B

- browser data xml sample 67, 155

## C

### C#

- create a project and add the Everest reference libraries 28
- create the add-in configuration file 39

- creating Everest Add-Ins 27
  - implement the IEverestAddin interface methods 36
  - install, register, and Enable your Add-In 39
  - make additional calls to the Everest form as required 38
  - Register the Everest reference libraries 27
- C# sample code 24
- create
  - add-in configuration file 123
- create a new activex library
  - delphi 7 47
- create a project and add the Everest Reference Libraries
  - C# 28
- create a project and add the Everest reference libraries
  - visual basic 6.0 41
- create an automation object
  - delphi 7 48
- create the add-in configuration file
  - C# 39
  - delphi 7 57
  - visual basic 6.0 45
- creating an add-in for Everest 19
- creating Everest Add-Ins
  - Delphi 7 46
- creating Everest Add-Ins using visual basic 6.0 40
- creating Everest-Add-ins Using C# 27

## D

- delete line item xml sample 95, 118, 165
- DeleteLineItem schema 150
- Delphi 7
  - create a new activex library 47
  - create an automation object 48
  - creating Everest Add-Ins 46
  - implement the IEverestAddin interface methods 53
  - register the Everest Reference Libraries 46
- delphi 7
  - install, register, and enable your add-in 57
  - make additional calls to the Everest form as required 57
- delphi sample code 20
- dialog box
  - add-in information 125
  - add-in manager 171

- document conventions 6
- document structure 5
- E**
- Everest Add-In samples 153
- Everest Add-In schemas 141
- Everest Add-Ins 16
  - about 11
  - assumptions 5
  - document conventions 6
  - document structure 5
  - features 3
  - getting started 11
  - introduction 3
  - overview 16
  - quick start 12
  - step by step procedures for developing 26
- Everest Browser 102
- Everest Client event hooks 133
- Everest Client menu hooks 133
- Everest Document 106
- Everest Document Line Item 113
- Everest Form 62
- Everest Payment 119
- Everest Profile 104
- Everest public interface 61
- Everest Session 99
- EverestBrowser.GetData 102
- EverestDcoument.AddCustomPage 107
- EverestDocument.UpdateDocument 108
- EverestDocumentLineItem.AddLineItem 114
- EverestDocumentLineItem.DeleteLineItem 118
- EverestDocumentLineItem.ModifyLineItem 115
- EverestForm.AddCustomPage 81
- EverestForm.AddLineItem 90
- EverestForm.Browser.GetData xml sample 103, 156
- EverestForm.DeleteLineItem 95
- EverestForm.ExecuteAction 72
- EverestForm.ExecuteAction Schema 142
- EverestForm.GetBrowserData 67
- EverestForm.GetData 70
- EverestForm.GetFormData 66
- EverestForm.GetPOSPaymentDetails 98
- EverestForm.GetProfileData 68
- EverestForm.GetSessionData 64
- EverestForm.ModifyLineItem 92
- EverestForm.Profile.GetData xml sample 105, 157
- EverestForm.QueryAction 71
- EverestForm.SetCustomFieldData 83
- EverestForm.SetFormData 82
- EverestForm.SetPOSPaymentDetails 96
- EverestForm.UpdateDocument 85
- EverestPayment.GetPOSPaymentDetails 122
- EverestPayment.SetPOSPaymentDetails 120
- EverestProfile.AddCustomPage 104
- EverestProfile.GetData 105
- EverestSession.GetData 99
- EverestSession.SetData 101
- EverstForm.GetData xml sample 70, 157
- example
  - add-in configuration file 124
- execution
  - add-in 132
- F**
- features 3
- G**
- getting started 11
- H**
- hooks
  - Everest Client event hooks 133
  - Everest Client menu hooks 133
- I**
- IEverestAddin 58
- IEverestAddin.Execute 58
- IEverestAddin.GetOptions 60
- IEverestAddin.OnCreate 59
- IEverestAddin.OnDestroy 60
- implement the IEverestAddin interface
  - methods
    - C# 36
    - delphi 7 53
    - visual basic 6.0 43
- install, register, and enable your add-in
  - delphi 7 57
  - visual basic 6.0 45
- installation
  - installing your add-ins 131
- installing your add-ins 131
- interface
  - add-in 58
  - Everest public 61
- introduction 3
- item profile form data xml sample 155

**L**

LineItem schema 148

**M**

make additional calls to the Everest form as required  
 C# 38  
 delphi 7 57  
 visual basic 6.0 45  
 microsoft windows vista  
 migrating to 174  
 migrating to microsoft windows vista 174  
 modify line item xml sample 93, 116, 163

**O**

overview 16  
 why xml? 18

**P**

POS payment xml sample 98, 122, 165  
 POSPayment schema 151  
 profile data xml sample 69, 157  
 public interface  
 Everest 61  
 PurchaseDocument schema 144

**Q**

quick start 12

**R**

register the Everest reference libraries  
 C# 27  
 delphi 7 46  
 visual basic 6.0 40  
 registration  
 add-in 131

**S**

SalesDocument schema 146  
 sample code  
 C# 24  
 delphi 20  
 visual basic 20  
 visual basic .NET 21  
 visual C++ 23  
 samples  
 Everest Add-In 153  
 working with sample add-ins 153  
 xml 154

**schema**

add-in configuration file 142  
 DeleteLineItem 150  
 EverestForm.ExecuteAction 142  
 LineItem 148  
 POSPayment 151  
 PurchaseDocument 144  
 SalesDocument 146  
 sControlData 143  
 SessionData 142  
 schemas 141  
 sControlData schema 143  
 session data xml sample 65, 100, 154  
 session xml sample 101  
 SessionData schema 142  
 step-by-step procedures for developing Everest Add-Ins 26

**U**

update purchase document xml sample 88, 111, 160  
 update sales document xml sample 86, 109, 158  
 utilities  
 add-ins 169  
 utility  
 add-in manager 170

**V**

validation  
 add-in 131  
 visual basic .NET sample code 21  
 Visual Basic 6.0  
 creating Everest Add-Ins 40  
 visual basic 6.0  
 create a project and add the Everest reference libraries 41  
 create the add-in configuration file 45  
 implement the IEverestAddin interface  
 methods 43  
 install, register, and enable your add-in 45  
 make additional calls to the Everest form as required 45  
 register the Everest reference libraries 40  
 visual basic sample code 20  
 visual C++ sample code 23

**W**

working with sample add-ins 153

**X**

- xml sample
  - EverestForm.Browser.GetData 103, 156
- xml samples 154
  - add line item 91, 114, 162
  - browser data 67, 155
  - delete line item 95, 118, 165
  - EverestForm.GetData 70, 157
  - EverestForm.Profile.GetData 105, 157
  - item profile form data 155
  - modify line item 93, 116, 163
  - POS payment 98, 122, 165
  - profile data 69, 157
  - session data 65, 100, 154
  - session xml sample 101
  - update purchase document 88, 111, 160
  - update sales document 86, 109
  - update sales document xml 158