Baan E-Sales 2.1

E-Sales 2.1 and E-Configuration 2.2 Integration Guide

A publication of:

Baan Development B.V.
P.O.Box 143
3770 AC Barneveld
The Netherlands

Printed in the Netherlands

Document Information

Code:	U7521A US
Group:	User Documentation
Edition:	A
Date:	October, 2000

Table of contents

About this document

This document describes how to set up the integration between Baan E-Sales 2.1 and Baan E-Configuration 2.2. This guide is intended for administrators who are responsible for the implementation of the Baan E-Enterprise suite.

1 E-Configurator Component

COM component that provides a simple wrapper around the Baan Internet Configurator 2.1 component (bic.dll) and the Baan E-Configuration 2.2 component (bec_haik.dll).

Definitions

BIC
Baan E-Configurator 99.1 (E-Configurator 2.1)

BEC
Baan E-Configurator 2.2

Installation checklist

EConfiguration.dll : Installed by E-Sales 2.1 Service Pack 1 installation

Bic.dll : Installed during E-Sales installation

Bec_haik.dll : Requires the following setup procedure:

1. Unregister the Bic.dll by using regsvr32.exe
2. Copy the Bec_haik.dll into the web server's E-Sales component directory
3. Register Bec_haik.dll by using regsvr32.exe
4. Add the following entries to the Windows Registry:
   - [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BEC\Parameters]
   - "ODBC report source"="ConfiguratorReport"
   - "ODBC config source"="ConfiguratorConfig"
   - "ODBC resource source"="ConfiguratorResource"
   - "ODBC connect password"=""
   - "ODBC connect username"="sa"
   - "ODBC admin source"="ConfiguratorAdmin"

5 Make sure that on the E-Sales web server the following ODBC DSN are created and configured for use by the Baan E-Configurator 2.2 database:
   – ConfiguratorAdmin
   – ConfiguratorConfig
   – ConfiguratorReport
   – ConfiguratorResource

   For example, if the Baan E-Configurator 2.2 database is located on machine cnl02090, the DSN entries all use the following:
   – SQL server authentication
   – Login ID: sa
   – Password: <empty>
   – Default database: becdb

6 Set the E-Sales BIC Path parameter to the directory where the Baan E-Configurator 2.2 Applet is located (<webserver>/Configurator/HAIK)

7 Reload “global.asp” (restart Internet Information Server)

Interface definition, usage and modification

Definition

The main class Econfigurator.Configurator implements the interface class IConfigurator1. The properties and methods exposed by this interface are:

- Property Get Completed(ConfigId As String) As Boolean
- Property Get HomeCurrency(ConfigId As String) As String
- Property Get Price(ConfigId As String) As Currency
- Property Get EDIString(ConfigId As String, WebOrderId, WebOrderLine) As String
- Property Get UsingBic() As Boolean
- Sub AddConfig(ConfigId As String, ProductId As String, LanguageCode As String, CountryCode As String)
- Sub SetFactor(ConfigId As String, CurrencyCode As String)

Usage

Visual Basic example

```
// using the IConfigurator1 interface
Dim obj as Econfigurator.IConfigurator1

// implemented by Configurator class
Set obj = new  Econfigurator.Configurator

// create a new configuration 'context' for product "hp" with id "1234"
obj.AddConfig "1234", "hp", "en", "US"

// retrieve the price in homecurrency units
price = obj.Price("1234")

// retrieve status of configuration, true if configuration is completed
if obj.Completed("1234") then

        // Determine version of Baan Configurator component
        // true if BIC2.1 (bic.dll) is used, false if BEC2.2 (bec-haik.dll) is used
        if obj.UsingBic then

                '// retrieve product variant structure in Baan EDI format
                edi = obj.EDIString("1234","1","1")
        else

                // method EDIString only supported for BEC2.2
                edi =""
        endif
endif
```

ASP Script example

```
Dim obj

// call special method on Configurator to cast obj to type IConfigurator1
Set obj = Server.CreateObject( _
                "Econfigurator.Configurator").GetInterface_IConfigurator1

// create a new configuration 'context' for product "hp" with id "1234"
obj.AddConfig "1234", "hp", "en", "US"

// tell configuration "1234" to use currencycode "USD"
// prices are now displayed in USD
obj.Setfactor "1234", "USD"

//start configurator applet
…
…
```

Modification

Although extending and creating interfaces requires basic component development knowledge, the following section will briefly describe the most important aspects of developing interfaces, and will outline a method for using the E-Configurator interfaces with the type unsafe ASP script language.

1 To add methods or properties, simply add the definition to the interface and implement the code in the Configurator class.

```
IConfigurator1:  Private Property Get MyProperty() as String
                 End Property

Configurator:    Private Property Get IConfigurator1_MyProperty() as String
                        IConfigurator1_MyProperty = "hello world"
                 End Property
```

2 To Add an interface without losing the old code (inheritance)

Before change:

IConfigurator1:
```
Private Property Get MyProperty() as String
End Property
```

Configurator:
```
Private Property Get IConfigurator1_MyProperty() as String
        IConfigurator1_MyProperty = “hello world”
End Property
```

After change

IConfigurator2:
```
Private Property Get NewProperty() as String
End Property
```

Configurator:
```
Implements IConfigurator2

Private Property Get IConfigurator2_MyProperty() as String
        IConfigurator2_MyProperty = IConfigurator1_MyProperty
End Property

Private Property Get IConfigurator2_NewProperty() as String
        IConfigurator2_NewProperty = “hello again”
End Property
```

3 Keep Configurator class “ASP script compatible”. Because in ASP script it is not possible to dimension types a special method of the Configurator class will allow one to “cast” the variant script variable to the desired interface.

Configurator:

```
Public Function GetInterface_IConfigurator1() As EConfigurator.IConfigurator1
   Set GetInterface_IConfigurator1 = Me
End Function
```

ASP Script:

```
Set obj = Server.CreateObject( _
                “Econfiguration.Configurator). GetInterface_IConfigurator1
```

E-Sales 2.1 Service Pack 1 Configurator related changes

The following changes took place:

Components

- EEPipeline.PriceDiscount
  - Private Function ConfigurableWebPricing(…)
- ESales.WebOrder
  - Private Function ProcessOneOrder(…)

ASP Script

- ../E-Catalog/EndUser/Product_List.asp
  - Function Handle_Fields(…)
- ../E-Catalog/EndUser/Product_View.asp
  - sub fillConfigInfo(…)
  - function buildConfiguratorLink(…)
- ../E-Sales/CurrentOrder/Item_List.asp
  - function GenerateConfigLink

Step-by-step example

The following example describes a typical setup and configuration scenario. The example will highlight all the steps required to create a configurable product in E-Sales.

Installation

1. Install the Baan E-Configurator 2.2 + Service Pack 1.
2. Install the Baan E-Configurator Modeler.
3. Install E-Sales 2.1 + Service Pack 1.

Export Baan item

4. Start Baan Windows.
5. Start the Maintain BC-Export Interface Parameters (tipcf7200m000) session.
6. Verify settings (that is, memorize the path name of the **Path to Output files** field).
7. Start the Generate Baan Configuration Model from PCF Model (tipcf7209m000) session.

8 Select the generic item that you want to export (for example, select **LAPTOP**) and click **Continue**.

9 In the directory pointed to in **Path to Output files** field (see step 6); there are two files (assuming there is no price matrix) : LAPTOP.bcp and LAPTOP.res.

10 Copy (or ftp) these two files to a directory on the computer where the Baan E-Configurator Modeler is installed ("..\BaanConfiguration\Modeler\Projects" folder).

Create Configurator model

11 Start the BEC Modeler.

12 Create a new project.

13 Remove the (default) model (do not save the project).

14 Import the "LAPTOP.bcp" file.

15 Save the project in the "..\projects" directory.

16 Make sure that the resource that represents the price of the product is identified as "Price" (E-sales will look for this resource). Check this in the model-laptop.bpd screen, select properties of \#Toplevel\Price.

Common Properties

Name	Price
Identifier	Price

17 Compile the model.

18 Run the add-in E-Configuration Modeler Extension.

19 Save and exit.

Publish Model

20 Copy the contents of "..\Project\" folder to the "..\BaanConfiguration\Internet\demos\laptop" folder on the BEC2.2 web server.

21 Use your browser to connect to the BEC2.2 web server's admin site (that is, "http:\\<webserver>\ConfiguratorAdmin\").

22 Select **Server** and click **Add Server**.

23 Set **Hostname** to hostname server
Set Web, APP, and Active to Yes.

Click **Finish** and **OK** by the proxy server warning

24 Close browser and reconnect.

Models

25 Select **Models** and click **Insert**

26 Set **Model File** to the “laptop.b02” file in the “..\Demos\laptop\” folder

27 Click **Next**.

28 Select the appropriate category UID (for prices this will be Currency, see also step 40, Categories).

29 Click **Next**.

Languages

30 Select Models and select Languages (top-right panel)

31 Select **Insert**

32 Set **GUI File** to the “laptop.zip” file in the “..\Demos\laptop\” folder.

33 Set **Language** and **Country** to “*” .

34 Point **Model UID** to MODEL_LAPTOP

35 Click **Next**.

36 Repeat steps 27 to 32 for each language that is supported by the model (for example use **Language** is ”en” and **Country** is “US”).

Products

37 Select **Products** and click **Insert**.

38 Set **Product Code** to “laptop.”

39 Set **Name** to “laptop.”

40 Let **Default Product** be “0.”

41 In the **Model UID** list, select **model_laptop** (that is, this name is set in the Modeler as identifier of the laptop model (see Modeler -> laptop ->Common Propertires -> Identifier).

42 Click **Next** until finished (that is, until you are in the default Run-Time Options view)

Categories

43 Select **Categories**.

44 Check that the category “Currency” exists and that the **Home Unit of Measure** is set to **USD**.

Resource Source Options

45 Select **Resource Sources Options** and click **Insert**.

46 Set **Resource Source UID** to **laptop_lineprices**.

47 Set **Resource Unit** to **USD** (note that this currency code must also exist in E-Sales)

NOTE: If you use demo resource created during E-Configuration installation, make sure that the **Source Reference** field is set to **ConfiguratorResource**.

48 Set **Resource Format** to **unused**.

49 Set **Table Name** to **res_laptop** (your are expected to manually create a table in the configurator database with this name. The table must have two columns, one for the key (“ext_id”) and one for the value (“line_prices”).

Table:

Line 1	ext_id	varchar	64	0
Line 2	Price	float	8	53

This tables is used to store the product variant options with their associated values (in our laptop example, this are the prices for the different options. See section at the end of this document)

50 Set **Key Column Name** to **ext_id**

51 Set **Value Column Name** to **Line_Prices**, and click **Next** until finished.

Resource Groups

52 Select **Resource Groups**.

53 Click **Insert**

54 Set **Group ID** to **laptop_resourcegroup**.

55 Set **Group Type** to **1** and click **Next**.

56 Set **Resource Source ID** to **laptop_lineprices**.

57 Set **Resource UID** to **Line_Prices** and click **Next**.

Resource Bindings

58 Select **Resource Bindings**.

59 Click **Insert.**

60 Set **Resource Binding UID** to **laptop** and click **Next**.

61 Set **Resource Group UID** to **laptop_resourcegroup**.

Import Resources

62 Before you can import the resources, you must create the table in the Configurator database.

63 Select **Import Resources**.

64 Browse to “laptop.res” file in the “..\Demos\laptop\” folder.

65 Set **Table Name** to **res_laptop**.

66 Set **Key Column Name** to **ext_id**.

67 Set **Value Column Name** to **line_prices**.

68 Click **Next**.

Configure E-Sales

69 Start E-Sales.

70 Add a configurable product with **Part Number** as **laptop**.

71 Check currency codes (that is, make sure the home unit of measure of your model matches an currency code in E-Sales).

NOTE: The original settings of demo products are available in E-Sales 2.1. If you use the demo products in E-Configuration 2.2 to work with E-Sales 2.1, make sure that you have carried out steps 16, 36 and 48.

Error checklist

In case the integration with the E-Configurator does not work, first check the steps as mentioned earlier. After that, make sure that the following topics are present

1 **Setup Sales parameters**
In the E-Sales startup page, after logging on, go to "E-Manager"-"E-Sales"-"Sales Parameters". Select "BIC Virtual Directory". Fill in the path to the server directory where the Baan E-Configurator is located. Use one of the two options below

BIC:

For example: http://cnl05352/configuratorESIK/

BEC:

For example: http://cnl02090/configuratorHAIK/

2 **Setup ODBC DSN**
Set the ODBC System DSN connections in the Windows NT Control Panel.

BIC:

ConfiguratorResource

ConfiguratorReport

ConfiguratorConfig

(user: bicsrv, passwd: bicsrv, default database: bicdb)

BEC:

ConfiguratorAdmin

ConfiguratorConfig

ConfiguratorReport

ConfiguratorResource

(user: becsrv, passwd: becsrv, default database: becdb)

3 **Presence if the Registry Key**
Make sure that the registry key is filled.

BIC:

None

BEC:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BEC\Parameters]

"ODBC report source"="ConfiguratorReport"

"ODBC config source"="ConfiguratorConfig"

"ODBC resource source"="ConfiguratorResource"

"ODBC connect password"="becsrv"

"ODBC connect username"="becsrv"

"ODBC admin source"="ConfiguratorAdmin"

4 **DLLs Registration**

Use regsrv32 to check whether the E-Configurator DLLs are registered correctly. The following registrations have to be in place on the E-Sales server:

BIC:

EConfigurator.dll

- ProgID: Configurator.Configurator

BIC.dll

- ProgID: BIC.ConfigData

NOTE: BEC_Haik.dll must be unregistered if the BIC configurator is used.

BEC:

EConfigurator.dll

- ProgID: Configurator.Configurator

BEC_HAIK.dll:

- ProgID 1: Bec_Haik.HAIK_Maint

- ProgID 2: Bec_Haik.HAIK_Usage

NOTE: Registration of BIC.dll is optional if the BEC configurator is used (See BIC).

After modifying the topics as described earlier, the E-Sales site must be restarted.