**IWS BI Dashboard Template
Engineer Guide**

Introduction

This document describes the procedures to import the Dashboard Template into an existing application, convert the project resolution, how to license the application and a basic description of the application tags, functions, scripts and screens.

Comments

General:

The software shall provide the necessary tools to allow a user to create a Dashboard based on the queries created.

Electronic Signatures (Security System)

The system administrator must be able to access the user account settings to create new accounts, lockout users, and de-authorize them.

Nobody (not even the System Administrator) can have access to the password of any user.

The Business Intelligence Dashboard Template application can be executed as a Stand-alone application, or merged into an User Project

Executing as Stand-Alone application

This section contains instructions on how to execute the BI Dashboard Template as a Stand-alone application.

The user can execute the application as a stand-alone, without importing it into a project. To do this, follow the steps below:

1- Open the development environment of IWS.

2- Open the BI Dashboard Template application and execute it.

Import Dashboard Template

This section contains instructions on how to import the Dashboard Template into another project. These instructions have to be followed for the project to run. Convert the Dashboard Template resolution before importing to the project.

1- Open the User Project where the Dashboard Template is going to be imported to.

2- Create a backup of the project scripts, open the Main Procedures, Right click with the mouse and click on Select All, copy it to a notepad and save. Repeat the process with the Graphics Script, and Startup Script. The backup is necessary so you don't lose the scripts after importing the application.
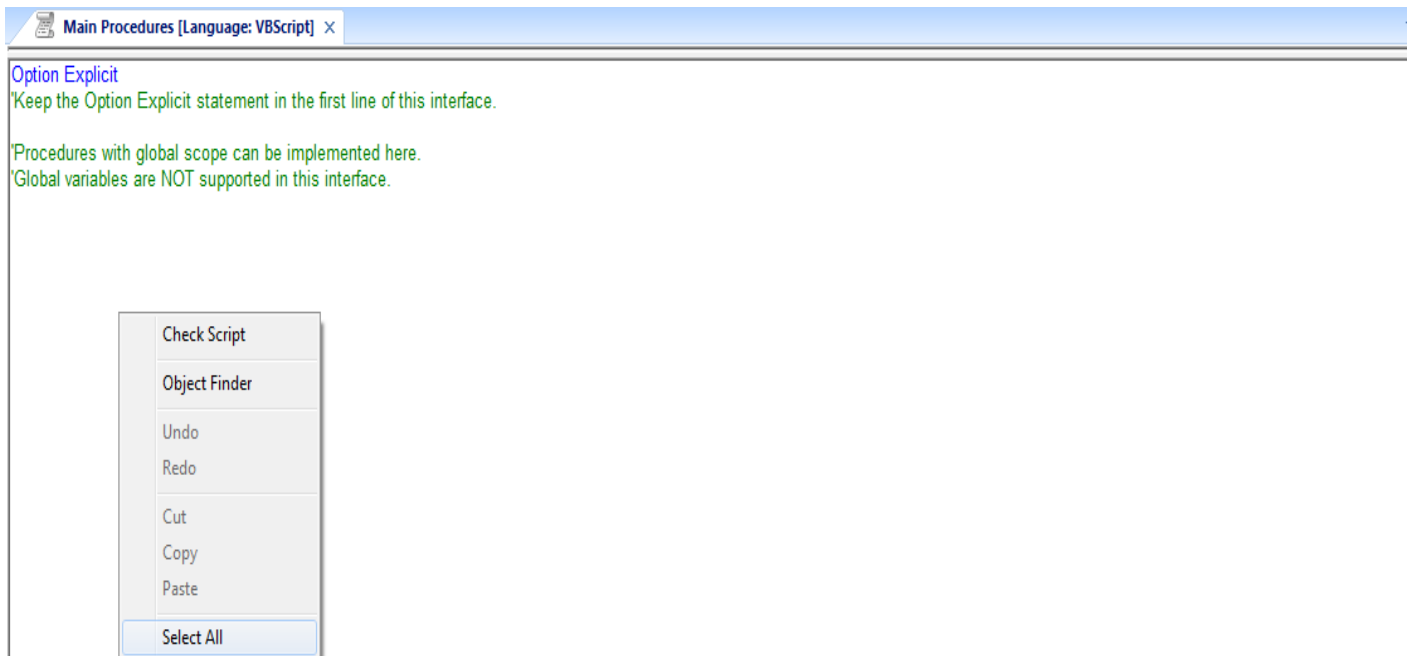
3- After creating the backup of the Scripts (make sure you already converted the resolution of the Dashboard Template application), with the User Project opened, go to the Home ribbon of the IWS and click on Import Wizard.

4- The following screen opens, select InduSoft Web Studio Database, and click on next
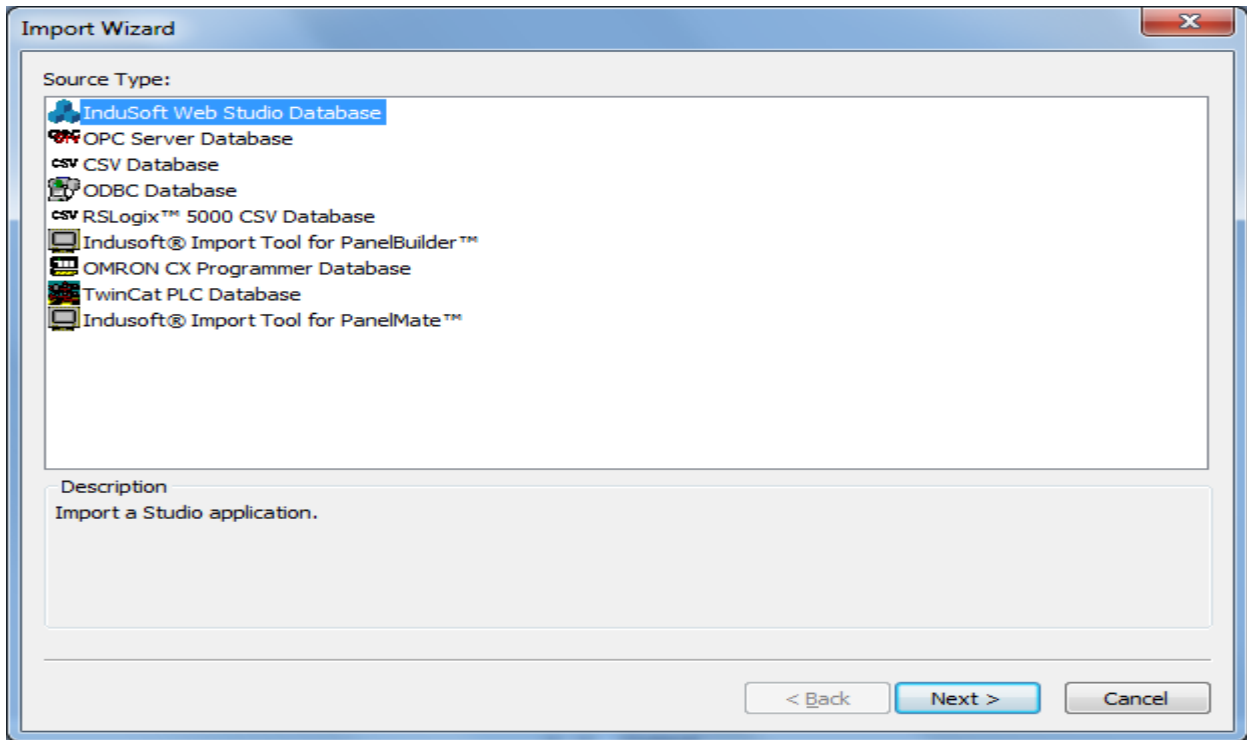
5- On the next screen, select "Import the whole project" and click on browse to find the Dashboard Template application. Select the Dashboard Template and click on next.
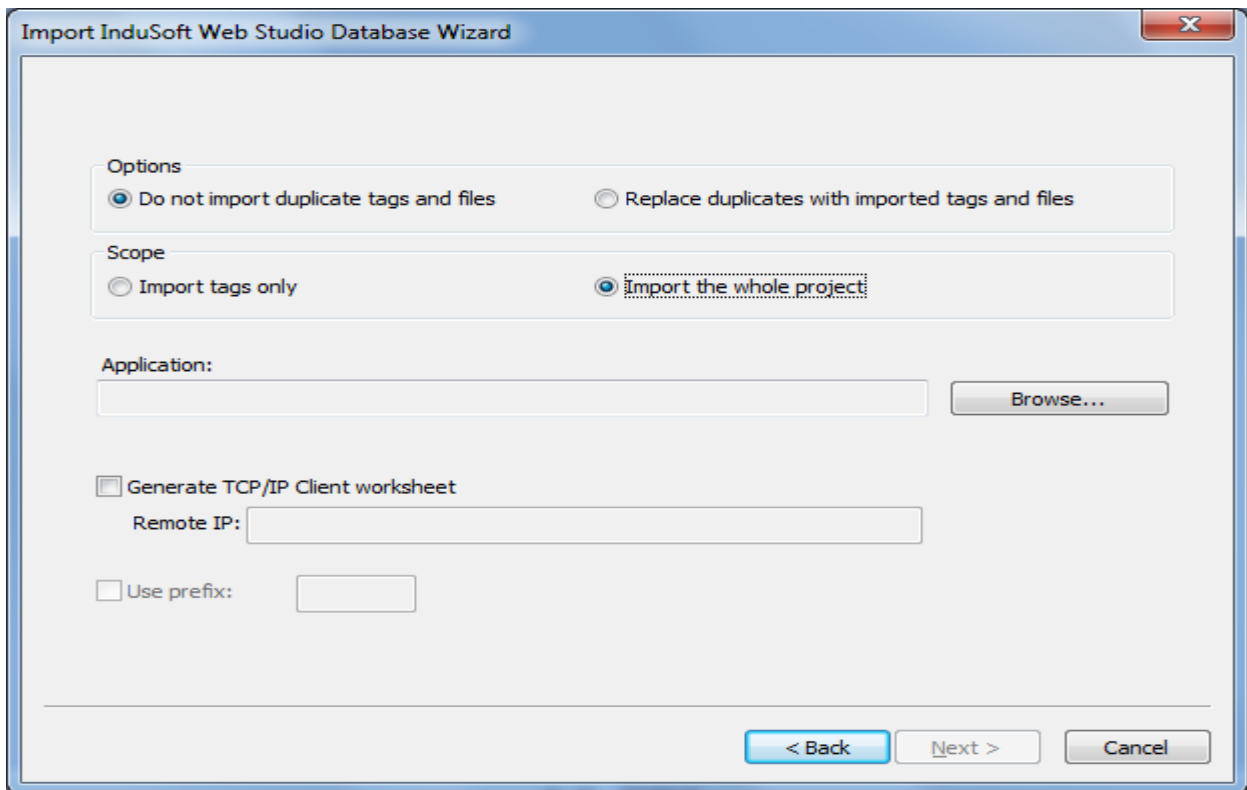
6- The screen below contains a list with the tags that will be imported to the project, and info about the import status for those tags. Check the list for any errors before importing. If there's any tag with an error, it'll be marked with red color. Check if there's any tag on the project with the same name as the imported tags, rename it and you should not have any errors.
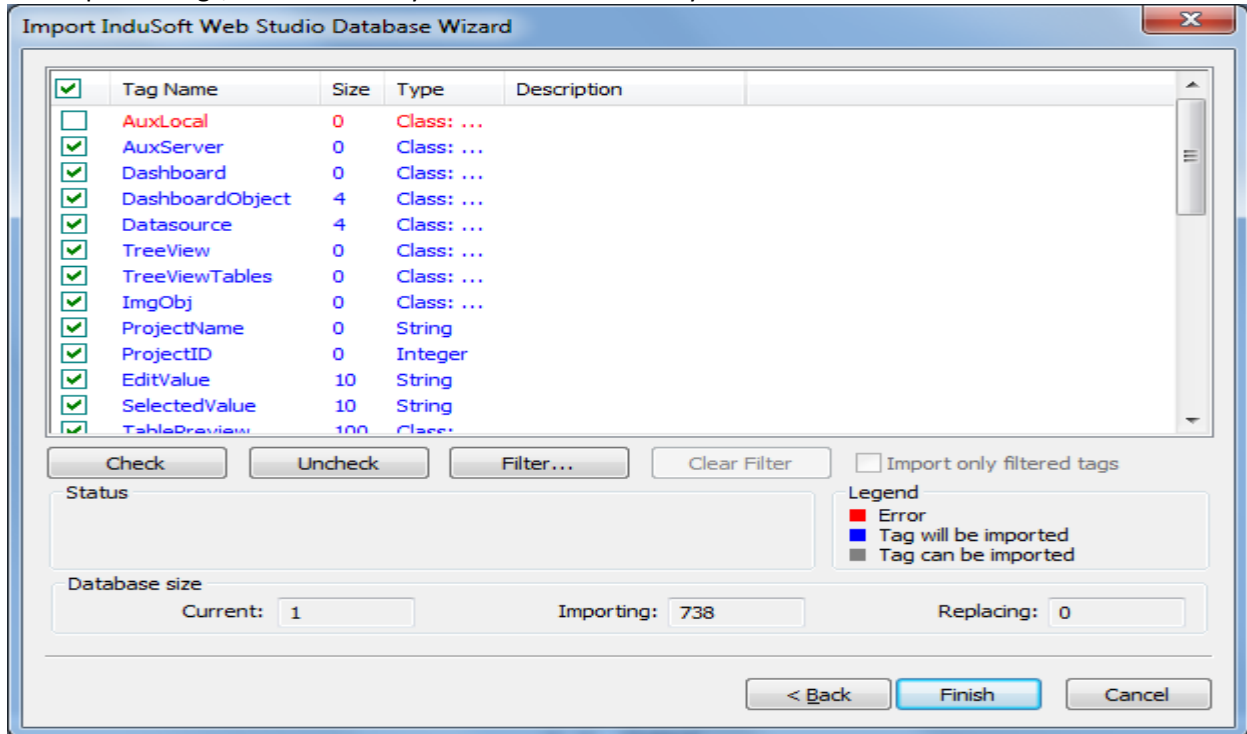
7- If there's no error, and all tags are listed with blue color, click on finish to import the Dashboard Template into the project.
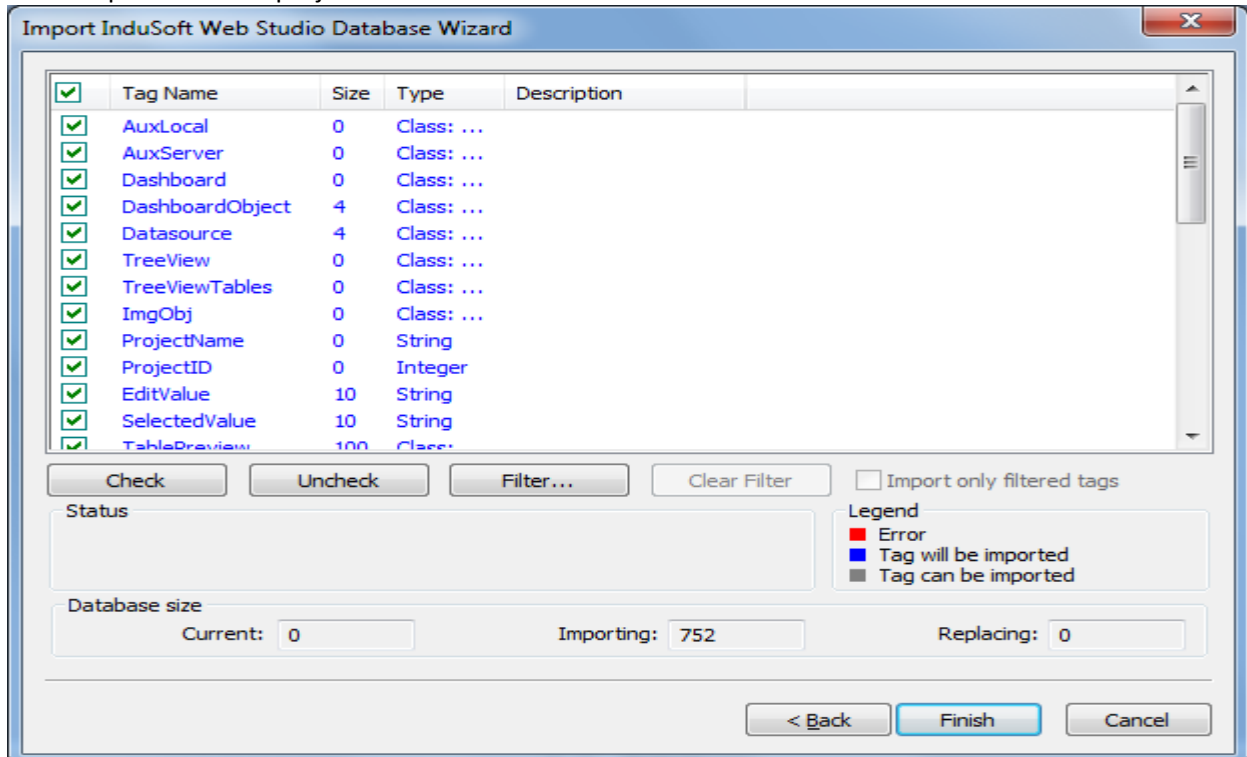
8- Wait until the IWS finish importing the application and you receive the message
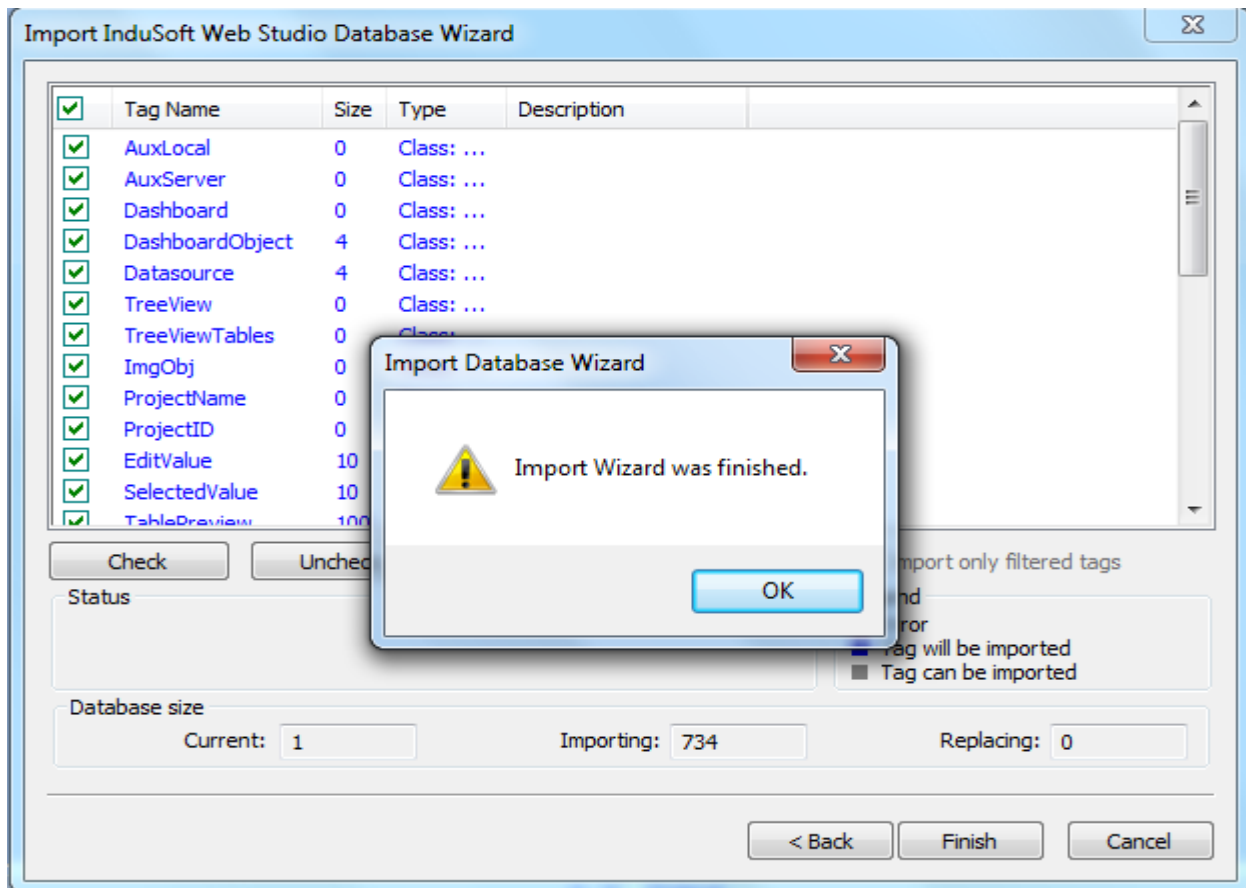
9- After importing the project, there're some configurations that should be done manually. The following configurations should be done:

- Include DB/ERP connections;
- Include screen group;
- Include the BI Dashboard Template Procedure;
- Copy the mdb settings;
- Copy the AddOn and Images Folder, and the ReportLogo image;
- Copy the Symbols folder
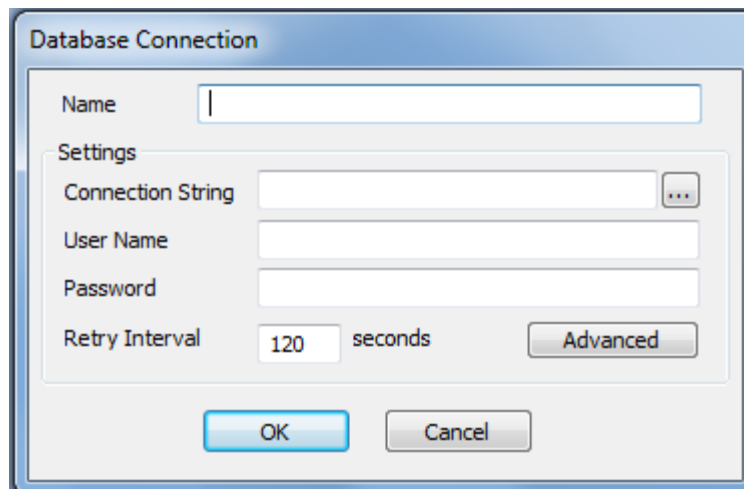- Copy the Graphics Script;
- Copy the Startup Script;

a. Include DB/ERP Connections: On the Project Explorer go to Tasks -> Database/ERP, Right click on Connections, and Insert.
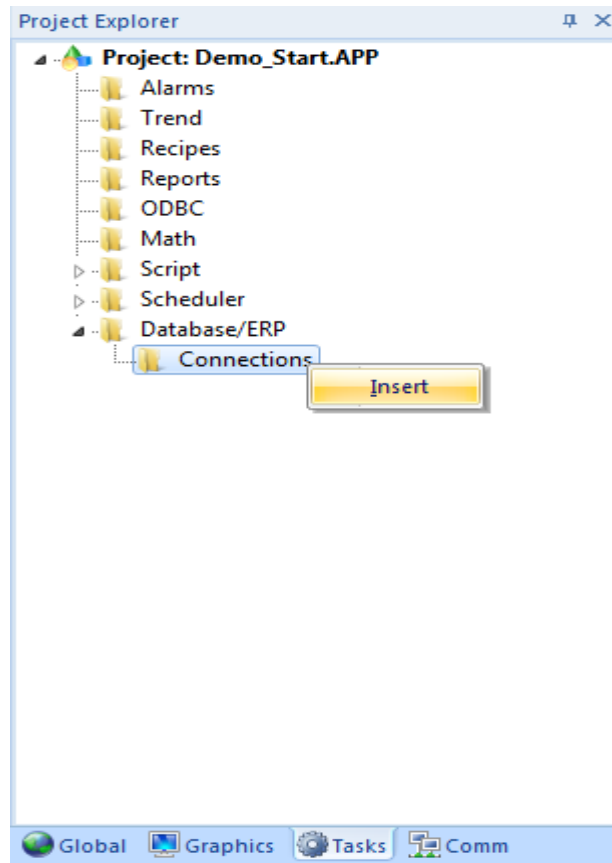
The Database Connection popup opens, configure the fields Name and Connection String with:

Name: CNF
Connection String: {"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + GetAppPath() + "AppSettings.mdb"}

Repeat the process for these settings:

Name: DS1
Connection String: {Datasource[1].ConnectionString}

Name: DS2
Connection String: {Datasource[2].ConnectionString}
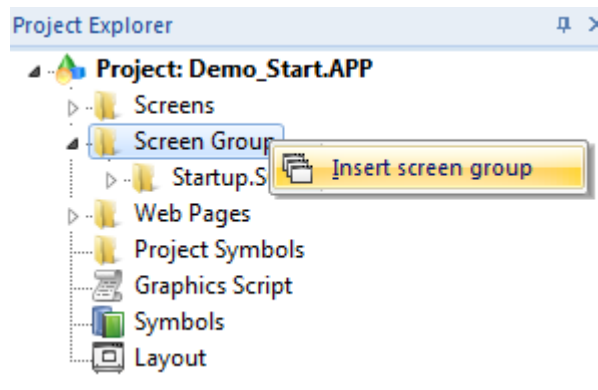
Name: DS3
Connection String: {Datasource[3].ConnectionString}

Name: DS4
Connection String: {Datasource[4].ConnectionString}

The Database/ERP connections for the Dashboard Template are configured now.

> b. Include Screen Group: On the Graphics tab of the project explorer, Right click on Screen Group and Insert Screen Group

A list with the screens show up, give it a name like "DBTemp" and select the screens HEADER, HOME and STATUSBAR from the list. These should be the screens from the Dashboard Application. After selecting them click on OK.
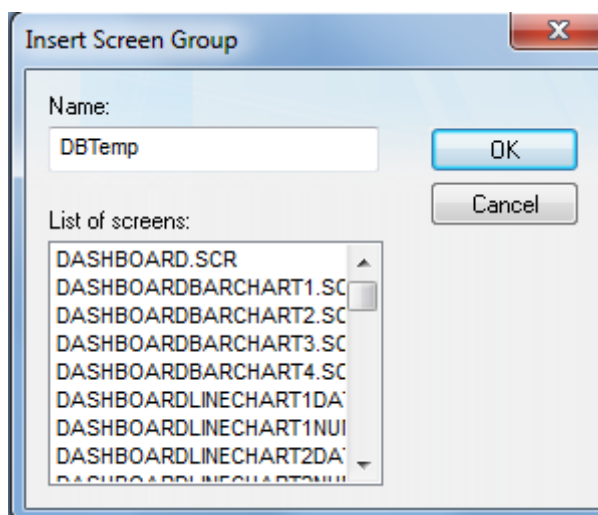
To open the Dashboard Application from your project, you have to create a button on one of the screens to open the screen group created. Create the button with the command to open the "DBTemp.sg", and the user can open the Dashboard Application clicking on this button.

    c.   Include BI Dashboard Template Procedures, LicenseDashboard and TreeView Procedures: Open the Config folder of the Dashboard Template application. Copy the BI Dashboard Template Procedures, LicenseDashboard and TreeView Procedures files to the Config folder of the User Project application.

ALARMHISTORY.XML	08/23/2012 8:20 AM	XML File	8 KB
BI Dashboard Template Procedures.gic	08/23/2012 11:28 ...	GIC File	698 KB
BI Dashboard Template Procedures.gis	08/23/2012 11:28 ...	GIS File	698 KB
CNF.xdc	03/19/2012 11:22 ...	XDC File	1 KB
DATABASEDEFAULTS.XML	08/23/2012 8:20 AM	XML File	8 KB
DS1.xdc	03/19/2012 11:22 ...	XDC File	1 KB
DS2.xdc	03/19/2012 11:22 ...	XDC File	1 KB
DS3.xdc	03/19/2012 11:22 ...	XDC File	1 KB
DS4.xdc	03/19/2012 11:22 ...	XDC File	1 KB
EVENTHISTORY.XML	08/23/2012 8:20 AM	XML File	8 KB
LicenseDashboard.gic	08/23/2012 11:28 ...	GIC File	79 KB
LicenseDashboard.gis	08/23/2012 11:28 ...	GIS File	79 KB
SCHED001.SCH	08/23/2012 11:28 ...	SQL Server Replica...	2 KB
SCRIPT0001.IS	08/23/2012 11:28 ...	IS File	5 KB
script0001.isc	08/23/2012 12:09 ...	ISC File	1 KB
SCRIPT0002.IS	08/23/2012 11:28 ...	IS File	9 KB
script0002.isc	08/23/2012 12:09 ...	ISC File	2 KB
scriptrt.isc	08/23/2012 12:09 ...	ISC File	178 KB
scriptrt.rtgis	08/23/2012 11:28 ...	RTGIS File	234 KB
scriptstartup.gic	08/23/2012 11:28 ...	GIC File	6 KB
scriptstartup.gis	08/23/2012 11:28 ...	GIS File	6 KB
scriptstartup.isc	08/23/2012 12:09 ...	ISC File	1 KB
TreeView Procedures.gic	08/23/2012 11:28 ...	GIC File	161 KB
TreeView Procedures.gis	08/23/2012 11:28 ...	GIS File	161 KB

d. Copy the mdb settings: Open the Dashboard Template folder, copy the AppSettings.mdb file and paste it on the User Project Folder.

Name	Type	Size
Alarm	File folder	
Config	File folder	
Database	File folder	
Hst	File folder	
Screen	File folder	
Symbol	File folder	
Web	File folder	
AppSettings.mdb	Microsoft Access ...	580 KB
DashBoardTemplate.APP	InduSoft Web Stu...	16 KB
DemoDB.mdb	Microsoft Access ...	464 KB

e. Copy the AddOn and Images Folder, and the ReportLogo: Open the Web folder of the Dashboard Template application, copy the AddOn and Images folders and the ReportLogo.jpg file and paste them on the Web folder of your project.

Note: It's possible to change the Logo on the reports by replacing the ReportLogo.jpg file on the Web folder. The new logo should be named ReportLogo.jpg for that to work.

AddOn	File folder	
Images	File folder	
Rep.css	Cascading Style S...	2 KB
ReportLogo.jpg	JPEG image	6 KB
Translation.trn	TRN File	495 KB

f. Copy the Symbols Folder: Copy the Symbols folder from the Dashboard Template application to your project folder. The Symbol folder of your project should now have the following files/folders:

General	File folder	
TextIO	File folder	
TRD.sym	SYM File	23 KB

g. Copy the Graphics Script: On the graphics tab of the project explorer open the Graphics Script. Right Click-> Select All and delete. Copy the Graphics Script from your Project there, and then copy the script from the Dashboard Template application.
Copy the following commands from the Dashboard Template script on the Graphics script.

```
Sub Graphics_OnStart()
    $AuxLocal.Language = "English"
    If $AuxLocal.IsServer=0 Then
        $ProjectName = $AuxServer.ProjectName
        $ProjectID = $AuxServer.ProjectID
        $AuxLocal.SkinDefault = $AuxServer.SkinDefault
    End If
End Sub


Sub Graphics_WhileRunning()
    If $AuxLocal.IsServer=1 Then
        $AuxServer.ProjectName = $ProjectName
        $AuxServer.ProjectID = $ProjectID
        $AuxServer.SkinDefault=$AuxLocal.SkinDefault
    End If
End Sub
```

h. Copy the Startup Script: On the tasks tab of the Project explorer, open the Startup Script, and copy both the backup texts from your Project and the Dashboard Application. The Startup Script should have both the instructions from the Dashboard Template and your project Project in it:

```
$AuxLocal.IsServer=1
$AuxLocal.Language = "English"
Call SetProjectLanguage("English", 1033)
$AuxServer.StatusMessage = $Ext("Checking Application License")
Call StartupLicense(1)
$StartBracket = "["
$EndBracket = "]"
Call UpdateFile()
```

10- Verify Project: On the Home ribbon of the IWS, click on the Verify button.

The following popup opens, click on Verify.

Page 10/22
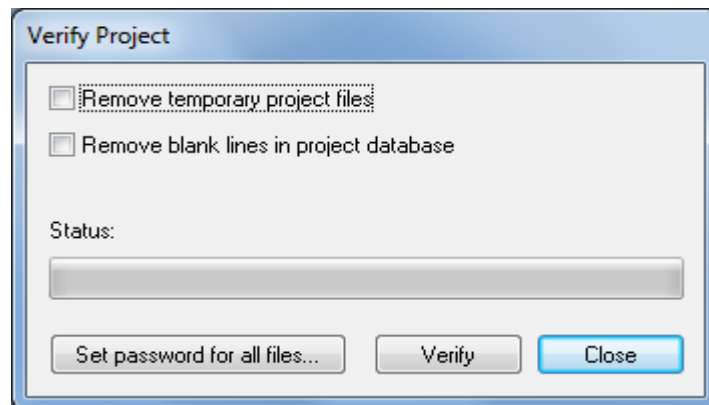
Wait for the IWS to verify the Project and the message with the result appears.
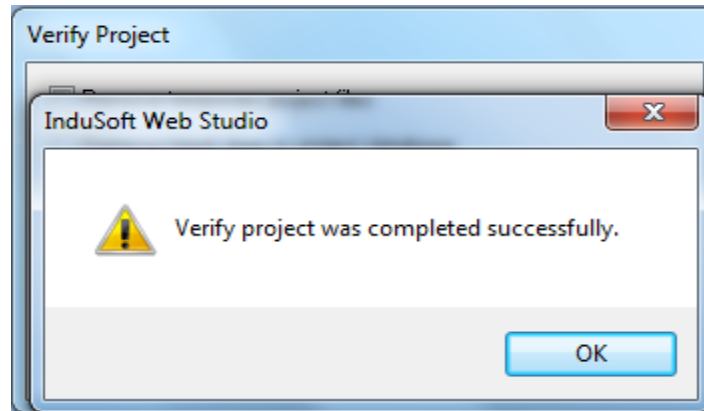
The Dashboard Template can now be accessed from your Project.

Important Notes:

- Make sure there're no screens, tags, scripts, or functions with the same name as the ones used on the Dashboard Application. A list with them is at the Application Basic Description section of this manual, if the project uses any of those names, please rename them in the project before importing the application.
- Convert the Dashboard Application resolution before importing it to your Project.
- Verify the project after importing the Dashboard Template, to make sure it's running properly.
- You may get an error message while importing the project if you don't have the Studio Controls installed. You can install the Studio Controls by executing the file Setup.exe from the "\BIDashboardTemplate\Web\AddOn" directory. Further information about how to install the Studio Controls is available in the BI Dashboard Template User Manual.

Convert Resolution

In this section the engineer can learn the procedures to convert the application resolution, before importing it into a different project. These procedures have to be done before importing the application.

1- The first step before converting the application is to make a backup of the screens. You can find the screens at the Screens Folder of the application. The screens that create the dashboard objects should not be converted, so create a backup of the following screens:

- dashboardBarChart

- dashboardLineChartDateTime

- dashboardLineChartNumeric

- dashboardObjBlank

- dashboardPieChart

- dashboardTable

You'll have to use the backup of those screens to substitute the converted ones later.

2- Open the Dashboard Template application, go to the Home ribbon at the header of the InduSoft and click on Convert Resolution.

3- The following popup opens, here you can select the resolutions for the project. The resolution on the "From" area is the resolution of the project. Edit the resolution on the "To" area to the resolution of the project where the application will be imported to. For example, an application with Width 1440 and Height 900, you insert the resolution at the popup and click on convert.
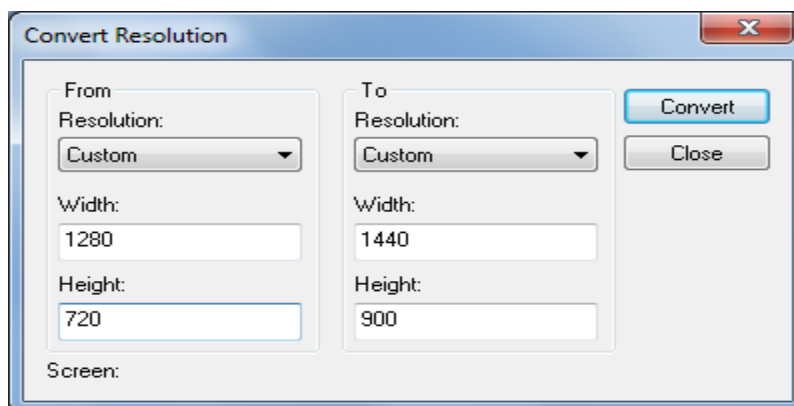
4- Wait until the InduSoft convert all the screens and you'll receive the following message:
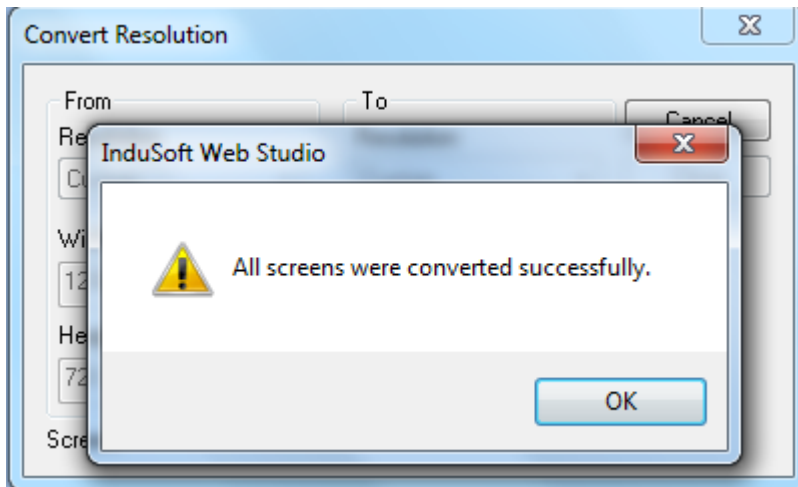
5- After all the screens were converted copy the screens from the backup back in the Screens Folder of the converted application. After copying the backup screens to the application, it can be used with the new resolution.

Important Notes:

It's not possible to convert the resolution of the popupLicense and popupLicenseChange screens. If you need to convert the resolution of these screens, please contact InduSoft.

Application Basic Description

Project Tags

This section contains a list with the project tags and a simple description for them.

Tag Name	Description
AuxLocal	Auxiliar Tag for Local Settings
AuxServer	Auxiliar Tag for Server Settings
Dashboard	Auxiliar Tag for the Dashboard Settings
DashboardObject	Auxiliar Tag for the Dashboard objects settings
Datasource	Auxiliar Tag for the Datasource Settings
TreeView	Auxiliar Tag for the TreeView Settings
TreeViewTables	Auxiliar Tag for the TreeViewTables Settings
ImgObj	Auxiliar Tag for the Dashboard Objects Image (number of objects in a dashboard)
ProjectName	Tag used to get the Name for the new project
ProjectID	Tag used to identify the current project, modified when the user open a project
EditValue	Tag used to edit the values used on the Grids
SelectedValue	Tag used to get the selected line on the Grid
TablePreview	Auxiliar Tag to create the table preview
LocalArrayIdx	Auxiliar Tag to send Array Tags value for other Array Tag
EditGridValue	Tag used to edit the values on the Grids on a different level
SelectedGridValue	Tag used to get the selected line on the Grid on a different level
ReloadGrid	Tag used to reload the grid

Tag Name	Description
TableObj1	Tag used to create the table object on the position 1
TableObj2	Tag used to create the table object on the position 2
TableObj3	Tag used to create the table object on the position 3
TableObj4	Tag used to create the table object on the position 4
CheckDBStatus	Auxiliar Tag used to verify the Database Status
DBStatus	Tag used to identify the Database Status
Rep	Auxiliar Tag to create the report
CNFSecSys	Auxiliar Tag used to manage the security system
CNFSecSysUsers	Auxiliar Tag used to manage the security system users
CNFSelUser	Auxiliar Tag to get the Selected user
RepStyle	Auxiliar Tag to select the Report orientation (Landscape, Portrait)
StartBracket	Delimiter inserted before the name of a column when creating a query
EndBracket	Delimiter inserted after the name of a column when creating a query

Project Global Procedures

This section contains a list with the Project Global Procedures and a description for them.

BI Dashboard Template Procedures

Procedure Name	Description
BuildGroupQuery	Build the GROUP Statement from the Wizard
BuildSelectQuery	Build the SELECT Statement from the Wizard
BuildSortQuery	Build the SORT Statement from the Wizard
BuildSQLQuery	Build the query from the Wizard
BuildTreeViewTableList	Get the List of tables from DB
BuildWhereQuery	Build the WHERE Statement from the Wizard
GetCNFDBSettings	Get settings from the GeneralSettings Table on the Configuration Database
GetCNFDBSettingsValue	Get value from the GeneralSettings Table on the Configuration Database
GetCols2GridObj	Get the column name from a specified Query
GetColumnDataType	Get the column type from a specified column
GetColumnDelimiter	Get the delimiter from a column type
GetComboIDFromColumnName	Get Combo Id from the column name
GetDatasourceIdx	Get the datasource Idx from a specified datasource name
GetDatasources	Configure the application Datasources
GetDBProvider	Open Datalink to select the connection string
GetIDFromNewData	Get ID from the inserted data
GetNextObjName	Get the default name for the next object
GetTableListFromDB	Get the List of tables from DB
GetTableQueryName	Get the QueryName from the specified ID

Procedure Name	Description
ReplaceSQLSpecialCharacter	Replace the special characters in the SQL string
SetCNFDBSettings	Set settings to the GeneralSettings Table on the configuration Database
SetCNFDBSettingsValue	Set value to the GeneralSettings Table on the configuration Database
CloseADOServer	Close StADOSvr.exe
CnfDBCompact	Compact the configuration database
EvtOnTimerChange	Event Timer
GetTableListFromDBAsync	Run the function to read the table list Async
HandleError	Generates diagnostic data when an error is detected in a function, indicating the Function Name, the Error Code and the Error Description
RunProcedureAsyncOnServer	Run the script Async
SetProjectLanguage	Set Project Language
GetSpecialFolder	Get the Special Folder for an object, like Temporary_Internet_Files, Local_Application_Data, Application_Data and User_Profile
RepAppendPicture	Appends a picture into the report
RepAppendTable	Appends a table (from a database) into the report
RepAppendText	Appends a text into the report
RepBreakPage	Breaks the current page and create a header for the next page, if applicable
RepBuildHeader	Writes the Header of the report
RepCheckPageBreak	Checks if the current page is filled, so a page break must be created
RepCreateCSS	Creates the style sheet (CSS file) for the report in the \Web sub-folder of the application. This function must be executed on the Server
RepCreateFile	Creates the report file
RepEnd	Finishes the report file. This function must be called once after appending all data into the report

Procedure Name	Description
RepStart	Starts a new report. This function must be called once when generating a new report
RepWriteToFile	Writes data to the file. This function must be executed on the Server to create the file on the Server computer
CloseDashBoardObject	Close the specified dashboard
DrawBarChart	Draw the Bar Chart object
DrawObject	Draw the Template object
DrawPieChart	Draw the Pie Chart object
GetBarColor	Get the color for the Bar
GetDashBoardObjectValues	Get values from the selected object
OpenDashboardObject	Open the specified object on the dashboard
OpenScreen	Open a Screen
OpenTemplate	Open the selected Template
ChangeVariableOnQuery	Replace the variable names in a query for its value before opening a dashboard
CreateNewVariable	Create a new variable in the project, that can be used on queries
DeleteProjectVariable	Delete the selected variable from the project
GetVariableAvailable	Verify if the variable name is available, or if it's already used in the project
ValidateNewVariable	Verify if the value of the variable is valid for its type and if the name does not contain special characters, before creating/editing a variable

TreeView Procedures

Procedure Name	Description
BuildTreeView	Build the TreeView on the MenuPanel
GetSelectedNode	Get the selected node from treeview object
TreeViewDeleteCmd	Delete the selected node

Procedure Name	Description
TreeViewEditCmd	Edit the selected node
TreeViewEditSelectedNode	Edit the selected node
TreeViewNewCmd	Create a new TreeView node

Project Screens

This section contains a list with the Project Screens and its description.

Screen Name	Description
dashboard	Screen where the user can see the dashboards.
dashboardBarChart1	Screen that creates the Bar Chart object on the position 1
dashboardBarChart2	Screen that creates the Bar Chart object on the position 2
dashboardBarChart3	Screen that creates the Bar Chart object on the position 3
dashboardBarChart4	Screen that creates the Bar Chart object on the position 4
dashboardLineChart1DateTime	Screen that creates the Line Chart – Date Time object on the position 1
dashboardLineChart1Numeric	Screen that creates the Line Chart – Numeric object on the position 1
dashboardLineChart2DateTime	Screen that creates the Line Chart – Date Time object on the position 2
dashboardLineChart2Numeric	Screen that creates the Line Chart – Numeric object on the position 2
dashboardLineChart3DateTime	Screen that creates the Line Chart – Date Time object on the position 3
dashboardLineChart3Numeric	Screen that creates the Line Chart – Numeric object on the position 3
dashboardLineChart4DateTime	Screen that creates the Line Chart – Date Time object on the position 4
dashboardLineChart4Numeric	Screen that creates the Line Chart – Numeric object on the position 4
dashboardObjBlank	Screen to show when the object is blank on any position
dashboardPieChart1	Screen that creates the Pie Chart object on the position 1
dashboardPieChart2	Screen that creates the Pie Chart object on the position 2
dashboardPieChart3	Screen that creates the Pie Chart object on the position 3
dashboardPieChart4	Screen that creates the Pie Chart object on the position 4
dashboardTable	Screen to create the Dashboard Table objects.
header	Screen with the header of the application.
home	Main Screen where the user can install the Chart Components.
menuPanel	Screen where the application creates the TreeView.

Screen Name	Description
popupDashboardSettings	Screen where the user can change the default settings for Dashboards.
popupDBStatus	Screen where the user can see the status for the DB connections.
popupEditDashboard	Screen where the user create/edit one Dashboard.
popupEditDatasource	Screen where the user create/edit one Datasource connection.
popupEditObject	Screen where the user edit the selected object for the Dashboard.
popupEditProject	Screen where the user create a new project.
popupEditUsers	Screen where the user configure the users for the application.
popupEditVariable	Screen where the user configure a variable used in the queries.
popupLanguages	Screen where the user select the language for the application.
popupOpenDatasource	Screen where the user select the Datasource to use for a query.
popupOpenProject	Screen where the user select one project to open.
popupOpenTableQuery	Screen where the user select the Table Query to use.
popupQueryDesignFilter	Screen where the user can see the Filters for the Design Query.
popupQueryDesignGroup	Screen where the user can see the Groups for the Design Query.
popupQueryDesignPreview	Screen where the user can see a preview of the Table Query.
popupQueryDesignSort	Screen where the user can see the Sorts for the Design Query.
popupQueryDesignView	Screen where the user can create a query on Design Mode.
popupQueryEditColumn	Screen where the user can add/edit a column for the Design Query.
popupQueryEditFilter	Screen where the user create/edit the Filters for the Design Query.
popupQueryEditGroup	Screen where the user create/edit the Groups for the Design Query.
popupqueryEditSort	Screen where the user create/edit the Sorts for the Design Query.
popupQuerySQLView	Screen where the user can create a query on SQL Mode.
popupQueryType	Screen where the user selects if the query is Design/SQL Mode.
popupReport	Screen where the user can create reports for the selected Dashboard.

Screen Name	Description
popupSelectVariableFilter	Screen where the user can select a variable to be used as a filter on the Design Mode query.
popupVariableList	Screen where the user can see a list with the project variables and Add/Edit/Delete variables.
statusBar	Screen where the user can see the status of the application.

Project Scripts/Tasks

This section contains a list with the Project Scripts/Tasks and its description.

Script Name	Description
Startup Script	- Sets the Default language as English - Gets the application resolution - Gets the Configuration Database settings - Gets the Datasource connections
Async Functions	Verify if there's any Async function running and update the application status
Datasource Status	- Verify the Datasource connection status: - No Connection - Connection Ok - Connection Failed
Scheduler -> General	Trigger the tag to start the script to Verify the Datasource connection status
CNF Connection	Connection to the Database with the application settings
DS1 Connection	Connection with the first Datasource configured
DS2 Connection	Connection with the second Datasource configured
DS3 Connection	Connection with the third Datasource configured
DS4 Connection	Connection with the fourth Datasource configured