



# Overview of Solver's BI360: A Powerful, Excel-based BI Solution

---

**Reporting - Planning - Dashboards - Data Warehouse**

A Solver White Paper

July 2012  
solverusa.com  
Copyright © 2012



## Table of Contents

|   |    |
|---|----|
| Table of Contents .....                                 | 1  |
| Introduction .....                                      | 3  |
| Architecture Overview .....                             | 3  |
| User Interface Overview .....                           | 5  |
| Reporting Components .....                              | 6  |
| Excel Ribbon .....                                      | 6  |
| Excel Task Pane .....                                   | 7  |
| Composer for Ad-Hoc Reporting .....                     | 9  |
| Practical Usage Examples .....                          | 10 |
| Typical Implementation Process .....                    | 13 |
| BI360 – Planning Module .....                           | 14 |
| User Interface Overview .....                           | 15 |
| Planning Components .....                               | 15 |
| Data Settings.....                                      | 16 |
| Interface Settings.....                                 | 17 |
| Data Entry Window.....                                  | 17 |
| Practical Usage Examples.....                           | 18 |
| Typical Implementation Process .....                    | 20 |
| BI360 – Dashboard Module .....                          | 22 |
| User Interface Overview .....                           | 23 |
| System Requirements.....                                | 24 |
| BI360 – Data Warehouse.....                             | 27 |
| Data Warehouse Interface (Admin Console) Overview ..... | 27 |
| Data Warehouse Manager Components.....                  | 27 |
| Modules .....   | 27 |
| Dimensions.....   | 28 |
| Currency .....  | 29 |
| Data Explorer .....                                     | 30 |

|                                      |    |
|--------------------------------------|----|
| Processes .....                      | 31 |
| Practical Usage Examples .....       | 32 |
| Typical Implementation Process ..... | 33 |
| Module – Dimension Mapping .....     | 34 |

## Introduction

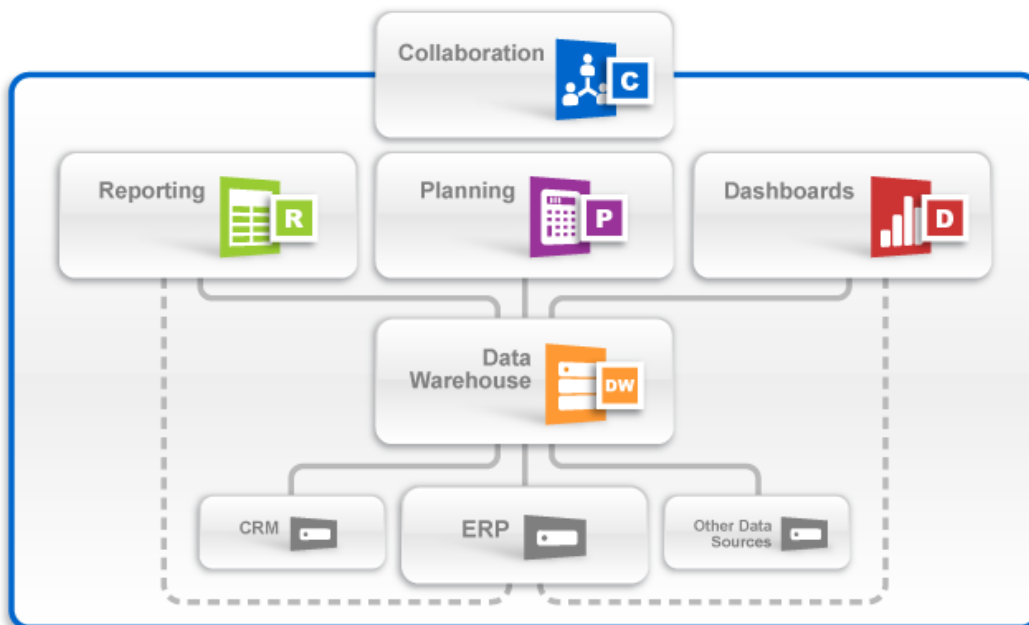
BI360 is the first full Business Intelligence (BI) suite powered by Microsoft Excel (Budgeting and Reporting), Microsoft Silverlight (Dashboards) and Microsoft SQL Server (Data Warehouse). Built for quick, low risk implementations and to be completely managed by business users, BI360 is designed to completely satisfy the business performance management needs of divisions as well as corporate headquarters.

The features that differentiate BI360 the most from other BI applications are:

1. Third generation Excel add-ins for simple or complex reporting and budgeting processes.
2. Pre-Configured and extensible “smart” data warehouse.

## Architecture Overview

Below you can see a simplified architecture of all BI360 modules. The *Reporting* module can both report live on any of the Microsoft Dynamics ERP systems as well as off the BI360 *Data Warehouse Manager*. The *Planning* module reads data to and from the *Data Warehouse Manager*, while the *Dashboard* module reads the KPIs from the *Data Warehouse Manager*. The *Data Warehouse Manager* allows BI360 to work on top of an organization’s data source(s) by loading data in and exposing it to BI360’s three end-user tools.



# Reporting



As an Excel-based report writer that can connect directly to your Dynamics ERP or CRM system, and to the BI360 *Data Warehouse Manager*, BI360 *Reporting* (OneStop Reporting 'OSR') can be used for financial reporting as well as various types of operational reporting.

With all, or a majority of the reports built with BI360 *Reporting*, organizations typically find that they now get 80% of the "BI" and reporting done with 20% of the effort it previously required. In other words, some of the key reasons for implementing OSR are:

1. Excel interface that most business users are comfortable with.
2. Easy to build reports with drag and drop wizards and pre-built formulas.
3. Reporting on GL as well as sub-ledgers and Microsoft CRM.
4. Reporting on *Data Warehouse Manager* (if implemented).
5. Single report writer for many purposes.
6. Consolidations across companies (requires BI360 *Data Warehouse Manager* if multiple different ERP systems are in use).
7. Currency conversion (requires BI360 *Data Warehouse Manager* for advanced currency conversion).
8. Special needs such as allocations and reconciliations.
9. Eliminate any manual Excel exports/reports as well as proprietary report writers.
10. Ad-hoc reporting (OSR Composer module) without the need for formulas, OLAP cubes, and pivot tables.

As seen in the architecture diagram earlier in this document, BI360 *Reporting* is Excel-based and it is one of four BI360 modules. The other three modules are:

1. BI360 *Data Warehouse Manager* – Optional, this is for support of reporting across multiple ERP systems, etc.).
2. BI360 *Planning* module - Optional.

3. BI360 *Dashboard* module – Optional, this provides a dashboard interface to data stored to the *Data Warehouse Manager*.

In the architecture diagram earlier in this document, you can see how BI360 *Reporting* interacts with the other BI360 modules. In essence, it serves as a report writer and an ad-hoc query tool both for live reporting on Microsoft's ERP and CRM systems, as well as for the BI360 *Data Warehouse Manager*, in which case data can come from any source system.

Here are some scenarios for when to use BI360 *Reporting* with the BI360 *Data Warehouse Manager*:

- Reporting across multiple source systems (like multiple ERP systems).
- Advanced currency conversion.
- Performance (want faster reporting without slowing down the source system).
- Budget reports (either for BI360's *Planning* module or for other budget system).
- Designing budget templates and other business input forms (with BI360's *Planning* module).

## User Interface Overview

BI360 *Reporting* is a modern Excel add-in that has been designed with the latest Microsoft technologies and follows Microsoft best practices for interface design and usability. The Interface was designed to maintain a similar look and feel to the Microsoft Office products, including the use of a ribbon and an Outlook-style menu bar as the main navigation components.

BI360 *Reporting* also has a back-end tool called the Meta Data Designer. It is the tool used by power users to create integrations between the Excel front-end and the data source (such as the ERP system). The meta data produced by the Meta Data Designer provides BI360 *Reporting* with major advantages over standard query tools and report writers that connects directly to a data source without any form of meta data. To begin with, it shields report designer users from any type of SQL queries and it translates often difficult SQL field and table names into plain English business terms. Maybe even more important, the Meta Data Designer allows developers and power users to pre-define important logic such as

time/period logic, which then can eliminate the use of technical period formulas when users design reports.

*Note: For Microsoft Dynamics systems, the BI360 Data Warehouse Manager and certain other systems, the meta data is already developed and out-of-the box, while for other systems it takes configuration to be set up. The Meta Data Designer can also be made available to partners/customers so that they, on their own, can expand the use of BI360 Reporting to also provide live reporting on other in-house databases.*

## Reporting Components

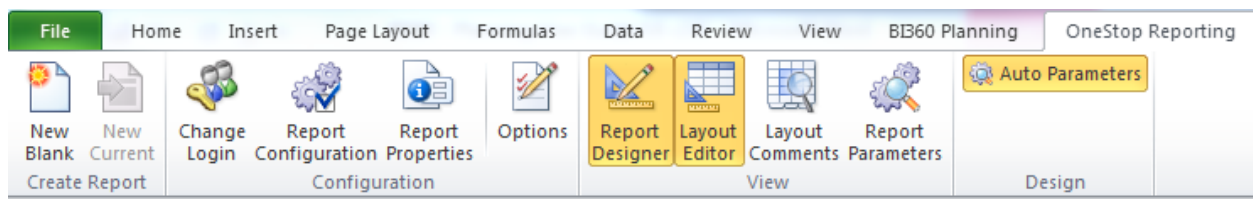
BI360 Reporting is an Excel add-in just like BI360's Planning module. As you see in the image below, both BI360 Reporting (OneStop Reporting) and BI360 Planning are located right next to each other on the Excel ribbon. In other words, Reporting and Planning are available side-by-side without the need to leave Excel.

BI360 Reporting consists of two menus: The Excel ribbon and the Excel task pane. The Excel ribbon is usually only used by administrators for setup and maintenance of BI360 Reporting reports.

## Excel Ribbon

The Excel ribbon has buttons that provide access to the following functions:

- Start the design of new reports (either from scratch or based on an existing Excel report).
- Manage your login (server connection).
- Manage configuration of the current report (as it relates to the data source).
- Manage report properties and setting options.
- Open/close Task pane and Design windows.
- Manage report parameters (prompts).



## Excel Task Pane

BI360's *Reporting* task pane features the popular Microsoft Outlook-style menu bar. It comes in two main flavors:

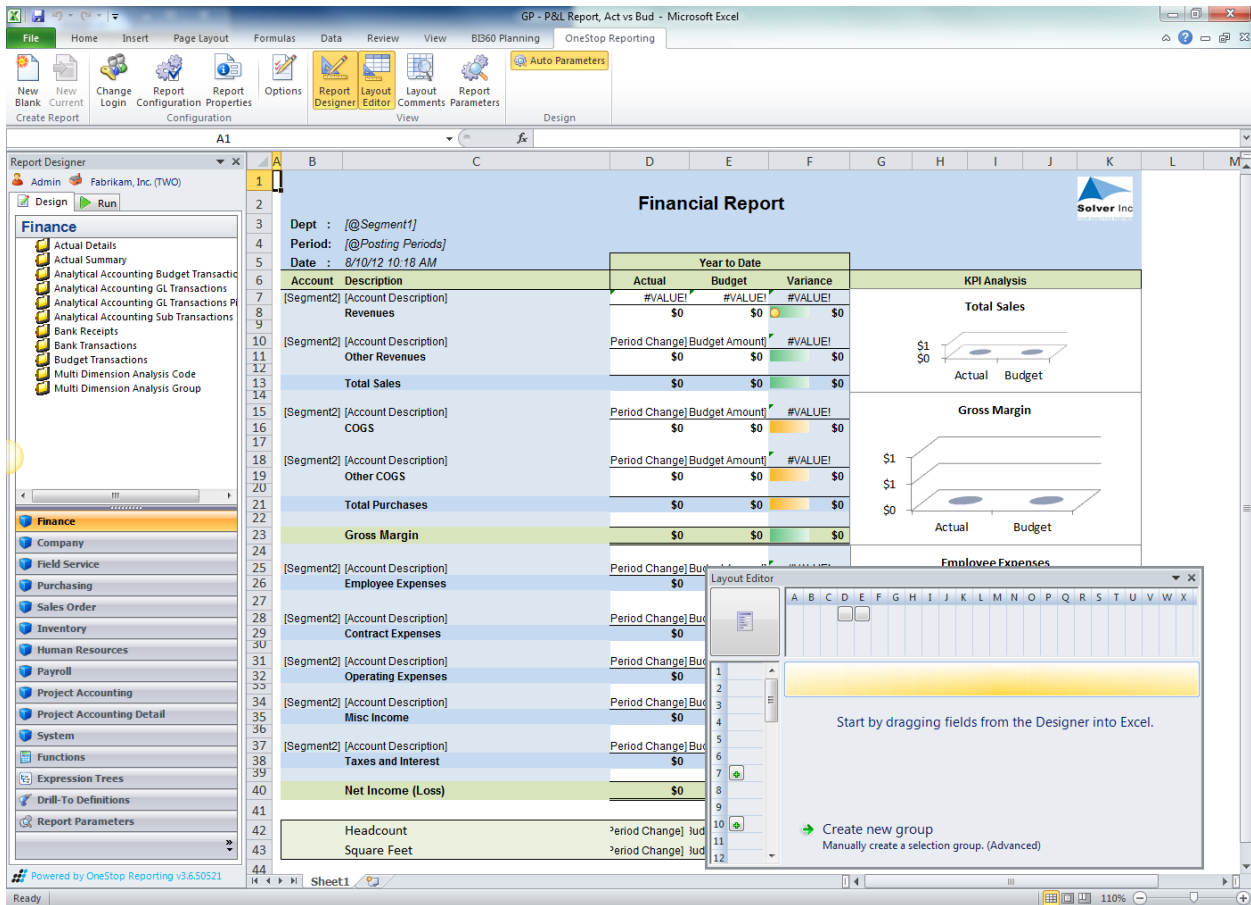
1. OneStop Reporting - Designer: Gives access to both design and run reports (typically for power users).
2. OneStop Reporting - Player: Gives access to run reports (typically for end users).

The OSR Designer menu is shown on the left side of the image below and it offers the following main features:

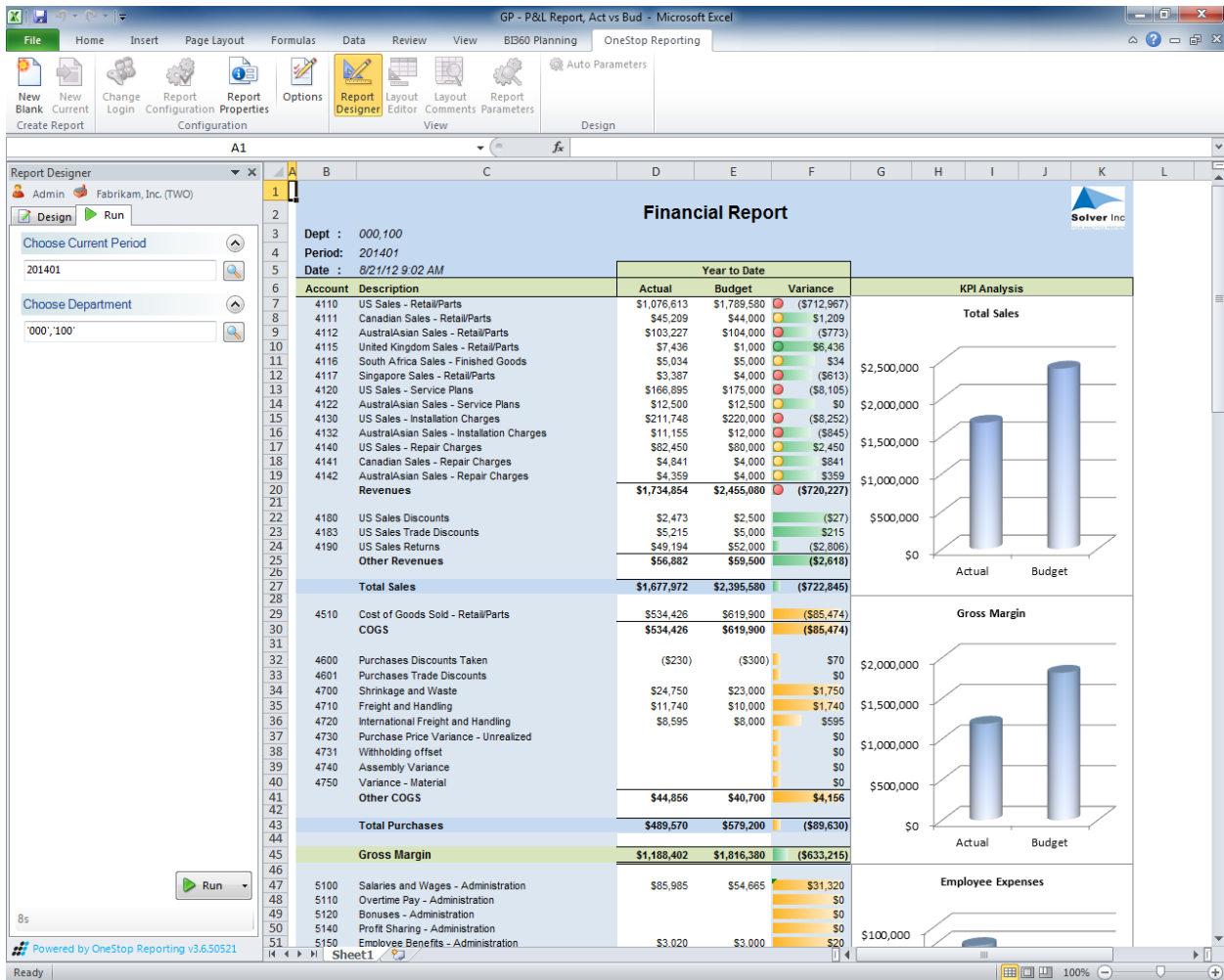
- Access to fields in different modules in the source database (like GL, Sales, Accounts Payable, etc.).
- KPIs (KPIs can be pre-defined and dragged into any report).
- Functions (pre-defined period functions such as: Current Month, YTD, Rolling 12 Month and many more).
- Expression Trees (pre-defined trees that can be dragged into any report. E.g. Profit & Loss account trees).
- Report Parameters (access to the parameters that will display for the users as filters when they run a report).

When designing a report, the user selects the fields, trees, periods, etc. from the OSR Designer menu and drags and drops these items into the appropriate location in the Excel sheet. Most reports can be designed without entering a single formula, with the exception of for example a standard Excel formula that deducts one Excel cell from another to calculate the variance between a column with actual figures and a column with budget figures.





The OSR Player menu is available as a second tab on the OSR task pane for users with the OSR Designer license and it is also available as a stand-alone for true end-users that only need it to run reports. Either way, it looks and works exactly the same. Essentially, it displays the report parameters as prompts on the left side of the screen (see image below) and the user can click on the lookup buttons to choose from dimension list that will appear (e.g. to choose a period or one or more departments). The report parameters (prompts) are created by the person designing the report, and they make a report truly dynamic so it can be run for any month, any department, any vendor, any project, and so on. The resulting data will be filtered before it is pulled from the source database, thereby making reports faster and eliminating the need to push all data to Excel to then filter it.



Once a report has been run (see image above), the user can right-click in any cell to drill down and analyze the underlying detailed transactions. The drill-down function is always available without the need to pre-define any drill down functions when the report is designed.

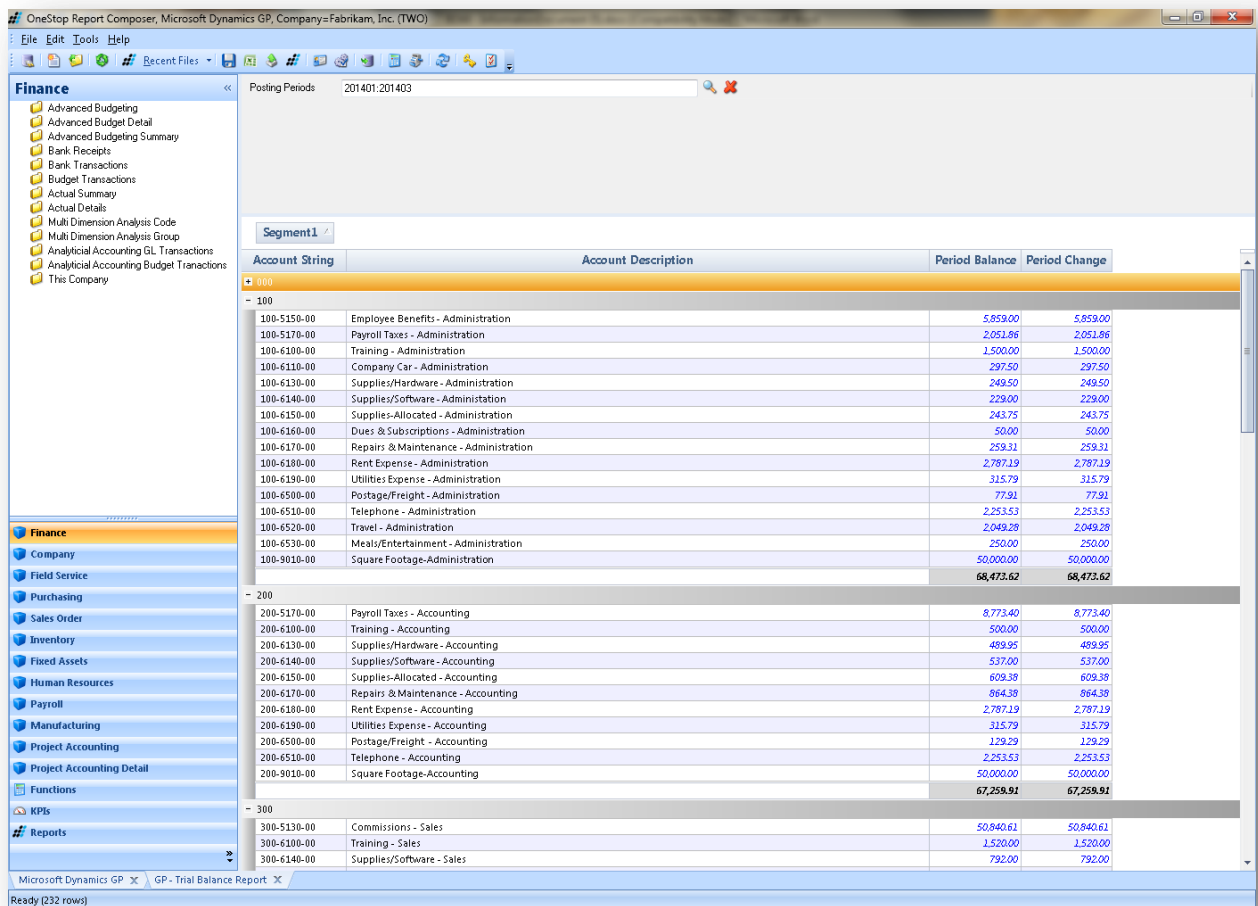
### Composer for Ad-Hoc Reporting

BI360 Reporting also offers an optional module called Composer (see image below). The Composer is designed to solve ad-hoc (on the fly, instant) reporting needs as compared to the BI360 Reporting Excel add-in which is focused on creating and running formatted reports. Typical ad-hoc users are accountants needing to see e.g. how much has been paid to a certain vendor so far this year, or a sales manager that wants to see what they have actually invoiced to a customer and if they have been paid yet.

The OSR Composer has a very simple interface where everything is accomplished with drag and drop from the familiar Outlook-style menu on

the left. This menu is the exact same menu that BI360 *Reporting* utilizes in Excel (see prior paragraphs about the OSR Designer).

The composer is also connected to the OSR Designer and a query designed in the Composer can, with a few clicks, be turned into a fully formatted Excel report template in the OSR Designer.



## Practical Usage Examples

In most cases, BI360 *Reporting* will be used as a reporting and ad-hoc query tool, either stand alone or as part of the BI360 suite. Either way, there are two main deployment options:

1. Give the reporting tools to a few central power users.
2. Provide the tools to end users so that on their own they can get information and drill-down whenever needed.

Below is a table listing various deployment options:

| <b>Deployment</b>  | <b>Advantage</b>  | <b>Disadvantage</b>  |
|--|---|--|
| Power users run reports centrally and distribute/e-mail to users                             | <ul style="list-style-type: none"> <li>• No license required for end users</li> </ul>   | <ul style="list-style-type: none"> <li>• End users cannot get reports exactly when they need them</li> <li>• End users may want to see reports for other filter criteria</li> <li>• End users cannot drill down</li> </ul> |
| Power users run reports and post to a shared network drive or SharePoint                     | <ul style="list-style-type: none"> <li>• No license required for end users</li> </ul>   | <ul style="list-style-type: none"> <li>• End users cannot get reports exactly when they need them</li> <li>• End users may want to see reports for other filter criteria</li> <li>• End users cannot drill down</li> </ul> |
| Power users create ad-hoc queries and export result to Excel and e-mail/print/save to server | <ul style="list-style-type: none"> <li>• No license required for end users</li> </ul>   | <ul style="list-style-type: none"> <li>• End users cannot answer instant questions when they need it and without requesting information from power user</li> </ul>   |
| End-users run their own reports  | <ul style="list-style-type: none"> <li>• End users get reports exactly when they need them</li> <li>• End users can run reports with different filter criteria</li> <li>• End users can drill down</li> </ul> | <ul style="list-style-type: none"> <li>• License required for end-user</li> </ul>  |
| End-users create ad-hoc queries  | <ul style="list-style-type: none"> <li>• End users can answer instant questions when they need it and without requesting information from power users</li> </ul>  | <ul style="list-style-type: none"> <li>• License required for end-user</li> </ul>  |

Here are examples of the numerous potential uses of BI360's Reporting module:

| <b>Purpose</b>        | <b>Usage Examples</b>  |
|-----------------------|--|
| Financial reporting   | <ul style="list-style-type: none"> <li>• Financial statements in various formats, like Profit &amp; Loss, Balance Sheet and Cash Flow reports.</li> </ul>  |
| Operational reporting | <ul style="list-style-type: none"> <li>• Sales reports</li> <li>• Vendor reports</li> <li>• Project reports</li> <li>• HR/Payroll reports</li> </ul>   |
| Dashboard reports     | <ul style="list-style-type: none"> <li>• Create Excel-based dashboards that combine metrics/KPIs with charts and indicators (Excel 2007 and later supports traffic lights, arrows, etc.).</li> <li>• If data is coming from multiple sources, deploy BI360's <i>Data Warehouse Manager</i> module, import data there and deploy <i>Dashboard</i> reports on top of the <i>Data Warehouse Manager</i>.</li> </ul>                               |
| Scorecard reports     | <ul style="list-style-type: none"> <li>• Create Excel-based scorecards with KPIs (Excel 2007 and later supports traffic lights, arrows, etc.).</li> <li>• Management's scorecard comments can be stored with BI360's Planning module.</li> <li>• If data is coming from multiple sources, deploy BI360's <i>Data Warehouse Manager</i>, import data there and deploy dashboard reports on top of the <i>Data Warehouse Manager</i>.</li> </ul> |
| KPI reports           | <ul style="list-style-type: none"> <li>• Create Excel-based reports that generate and display KPIs. (Excel 2007/2010 supports traffic lights, arrows, etc.).</li> </ul>  |
| Allocation reports    | <ul style="list-style-type: none"> <li>• Create Excel-based reports that calculate allocations.</li> <li>• If you want to store the allocations and bring them back to the ERP system, use <i>BI360's Planning</i> module. You can also use <i>BI360 Planning</i> to collect and store allocation drivers.</li> </ul>  |
| Commentary input      | <ul style="list-style-type: none"> <li>• Create Excel-based reports (e.g. a P&amp;L) and set up BI360's budget module to collect and save text comments.</li> <li>• Create Excel-based reports to show all comments made to certain accounts, for</li> </ul>   |

|                        |  |
|------------------------|--|
|                        | certain variances, etc.  |
| Budget template design | <ul style="list-style-type: none"> <li>• Create Excel-based input templates for use with BI360's budget module.</li> </ul>                 |
| Budget reports         | <ul style="list-style-type: none"> <li>• Create Excel-based reports to consolidate/view budget.</li> </ul>                                 |
| Workflow reports       | <ul style="list-style-type: none"> <li>• Create Excel-based reports for instant views of all Approved/Pending/Rejected budgets.</li> </ul> |

## Typical Implementation Process

A typical implementation process of BI360 *Reporting* could look like this (in this example assume that BI360 *Reporting* is being used for typical reporting processes):

1. Train power users.
2. Identify reporting needs.
3. Preparation of the BI360 *Data Warehouse Manager* – Skip this step if OSR will only be used for live reporting
  - a. Populate the *Data Warehouse Manager* with data and dimensions from your source systems. Examples of dimensions are: Account, Department, Project, Product, Employee, Asset, etc.
  - b. Enter any other dimensions or data directly in the *Data Warehouse Manager* if they do not already reside in another database from which they can be imported.
4. Design Excel report templates with BI360 *Reporting*.
5. Set up security (if different users require different access rights).
6. Train end users (will only take minutes as they just need to learn how to run reports and drill down).

# Planning



## BI360 – Planning Module

As an Excel-based data entry tool with a flexible SQL Server database and a powerful companion report writer (*BI360 Reporting*), *Planning* can be used to automate budgeting and forecasting as well as other business processes that require data entry and a database. In other words, think of *BI360 Planning* as an automated way to save any data from Excel into a structured SQL database. For most people, budgets and forecasts are the first processes that come to mind, but in reality, most organizations have numerous manual data collection processes that could be automated with *Planning/BI360*.

As seen in the architecture diagram earlier in this document, *Planning* is Excel-based and it is one of four BI360 modules. The other three modules are:

1. *BI360 Data Warehouse Manager* – Required, this is where *Planning* stores all data from Excel.
2. *BI360 Reporting* module – Optional.
  - a. Used if you want to create dynamics budget templates.
  - b. Used as a report writer to report on all the data that *Planning* store to the *Data Warehouse Manager* database.
3. *BI360 Dashboard* module – Optional, this provides a dashboard interface with the data stored to the *Data Warehouse Manager*.

In the architecture diagram shown earlier in this document, you see how *BI360 Planning* interacts with the other BI360 modules. As you will learn later in this document, administrators first configure *Planning* and the *Data Warehouse Manager* database for the Excel model where the data entry and write-back will take place. The Excel template(s) is then provided to the end-users for data entry. After the users have entered and saved their data, managers can start reporting on the entered data. Finally, if desirable, the data can be transferred from the *Data Warehouse Manager*

database and back to other source systems such as a budget table in the general ledger.

## User Interface Overview

BI360 *Planning* is a modern Excel add-in that has been designed with the latest Microsoft technologies and follows Microsoft best practices for interface design and usability. The interface was designed to maintain similar look and feel to the Office products, including the use of a ribbon and Outlook-style menu bar as the main navigation components.

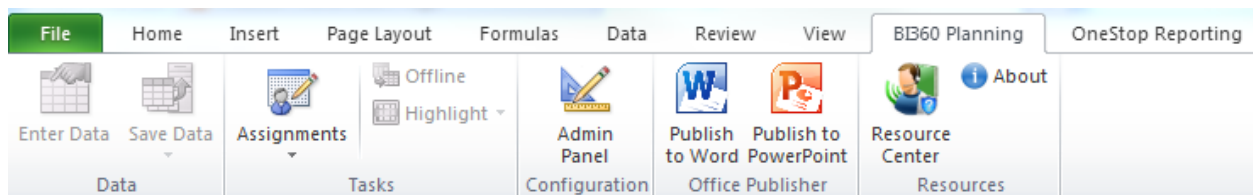
## Planning Components

BI360 *Planning* is an Excel add-in just like BI360 *Reporting*. As you see in the image below, both *Planning* and *Reporting* are located right next to each other on the Excel ribbon. In other words, reporting and budgeting are available side-by-side without the need to leave Excel.

BI360 *Planning* consists of two menus: the Excel ribbon and the Excel task pane. The latter is only used by administrators for setup and maintenance of *Planning*.

### Excel Ribbon

- View, Check-in & Check-out Assignments.
- Save data.
- Update model settings from the server (the BI360 *Data Warehouse Manager*).
- Connection information (to the BI360 *Data Warehouse Manager* database).
- Access to administrative functions (task pane).
- Direct link to resources (Including BI360 documentation, videos and samples).



### Excel Task Pane

BI360 *Planning* task pane features the popular Microsoft Outlook-style menu bar. It has three main menu items:



## Data Settings

This is where the administrator performs the initial configuration by providing *Planning* with vital information about where data and dimensions are located in the underlying spreadsheet. Depending on how well organized and structured your Excel model is, *BI360 Planning* can be quickly configured to handle unlimited data input sections in one or multiple worksheets in a workbook. *BI360 Planning* can also automatically adapt to spreadsheets that “grow” (e.g. a new row or column that gets inserted in the Excel model) or “shrink” without the administrator having to update the settings. If there is a significant structural change to the company’s Excel model, then the administrator can make the necessary changes to the *Planning* Data Settings that the end-users will see next time they open the Excel template (if the same Excel template is accessed centrally by all users) or if the Excel model has been distributed to end-users, they simply click the “Update Settings” button on the *Planning* Excel ribbon and it will retrieve the latest settings from the *BI360 Data Warehouse Manager* database. Part of the data setting setup is to choose where in the *BI360 Data Warehouse Manager* database the data from Excel should be stored based on the type of data. For example, data (like budgets) related to Accounts is stored to the Finance (General Ledger) Module, detailed revenue data is stored in the Revenue Module, detailed employee/position data is stored to the Payroll Module, and so on.

The screenshot shows the BI360 Data Settings dialog box. The dialog box is titled "Data Settings" and contains the following information:

- Worksheet:** Forecast
- Copy Settings from:** (empty)
- Grid Title:** ForecastGL
- Grid Rules:** Ignore "I" columns, Ignore "I" r
- Module:** GL
- Data Grid Range:** Forecast!\$F\$13:\$L\$14
- Add Optional Dimensions:** (empty)
- Dimension, Dimension Location, Cell Reference table:**

| Dimension | Dimension Location | Cell Reference           |
|-----------|--------------------|--------------------------|
| Account   | Row                | Forecast!\$B\$13         |
| Currency  | Global             | Forecast!\$C\$7          |
| Entity    | Global             | Forecast!\$C\$4          |
| Scenario  | Global             | Forecast!\$C\$6          |
| Period    | Column             | Forecast!\$F\$10:\$L\$10 |
- Add Storage Field:** UDF1, UDF2, Amount
- Field, Column Reference table:**

| Field  | Column Reference         |
|--------|--------------------------|
| UDF1   | Forecast!\$I\$13         |
| UDF2   | Forecast!\$J\$13         |
| Amount | Forecast!\$F\$13:\$G\$13 |

## Interface Settings

The *Planning* Interface Settings (see screenshot below) are directly related to the *Planning* Data Settings. While the Data Settings are focused on what sections of Excel should be saved where in the *Data Warehouse* and with which dimensions, the Interface Settings are focused on what you want the end-users to see if they use the *Planning* data entry screen (shown in the next paragraph).

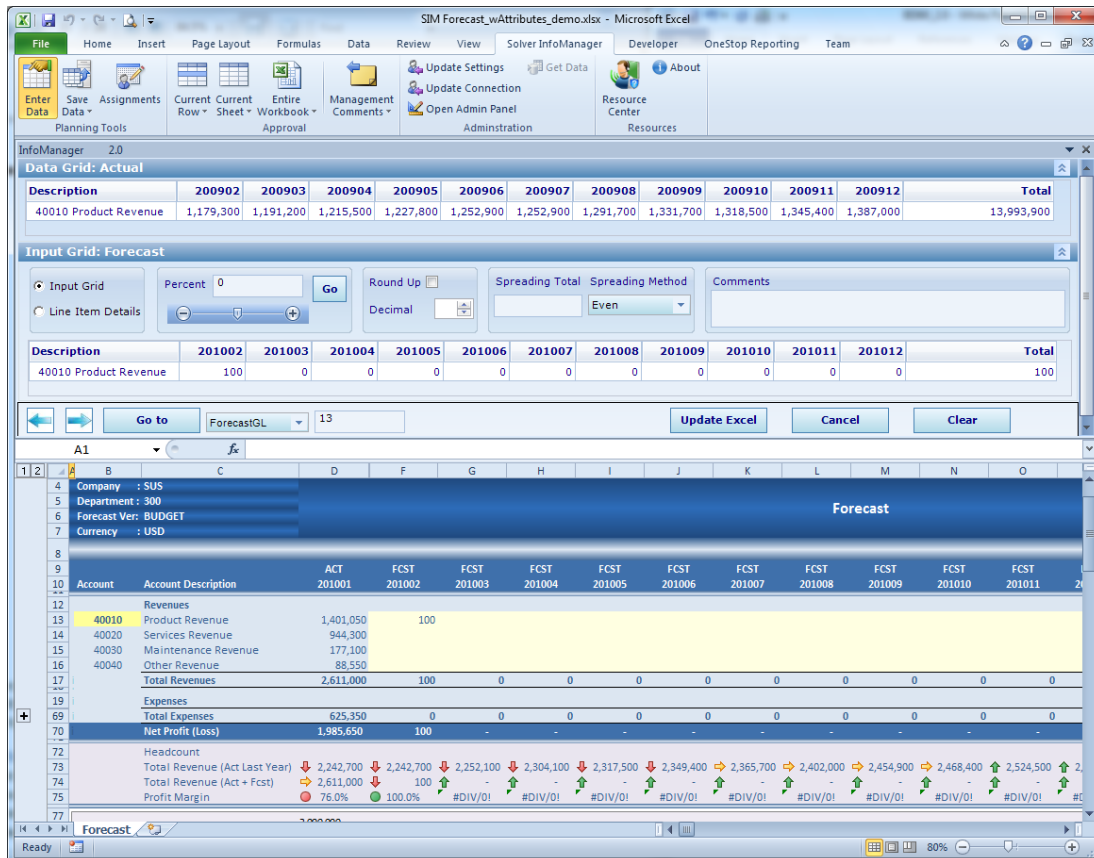


## Data Entry Window

The *Planning* data entry window (see image below) is an optional way of entering data in Excel. In other words, users can type information directly into Excel, or they can open the BI360 *Planning* data entry screen and enter it there. When the data has been entered into Excel, it is automatically saved into the underlying row in Excel and then can be saved to the *Data Warehouse* database. The *Planning* data entry screen is a powerful tool for:

- Automatic spreading of numbers across many cells in Excel. Many spreading rules are available, including rules that can use prior year actuals or seasonal trends as a base.
- Input of text comments at the row or line item level.

- Input of many rows of line item detail below a single row in Excel (e.g. the user can list many business trips that automatically will roll up to a single travel expense row in Excel).
- Automatic charting of the current row in Excel and comparison to e.g. another row with historical data.



In short, *BI360 Planning* can store data (numbers and text) from any Excel spreadsheet. The data will always be stored to the *Data Warehouse* database. From there, the stored data can be sent back to the ERP system, reported on or even processed into OLAP cubes for advanced analysis.

## Practical Usage Examples

In most cases, *Planning* will be used as a budgeting and forecasting tool. Below is a list of the most typical variations of Excel budget models that *BI360 Planning* can handle.

*Note that the Excel "input forms" can either be your own "static" Excel files or dynamic Excel templates that you design with BI360 Reporting or with similar Excel-based tools:*

| <b>Excel Configuration</b>  | <b>Examples</b>   | <b>Typical Distribution</b>  |
|---|---|--|
| Multiple Excel workbooks with a single budget sheet                                     | Departmental Profit & Loss<br>Departmental Expenses   | Input done by central staff<br>Spreadsheet\template distributed to end-users |
| Single Excel workbook with multiple budget sheets with the <u>same</u> layout           | Different departments on different sheets<br>Different products on different sheets                               | Input done by central staff  |
| Multiple similar Excel workbooks with multiple different budget sheets in each workbook | Workbook with:<br>Summary Profit & Loss or Expense sheet<br>Revenue Detail<br>Employee Detail<br>Capital Expenses | Spreadsheet\template distributed to end-users                                |

Here are examples of other uses of Planning as a data collection tool.

*Note that the Excel "input forms" can either be your own "static" Excel files or dynamic Excel templates that you design with the BI360 report writer or with similar Excel-based tools:*

| <b>Excel Configuration and Input Type</b> | <b>Usage Examples</b>  |
|---|--|
| Modeling                                  | Create a product or service price model in Excel. Use <i>BI360 Planning</i> to store each scenario to the <i>BI360 Data Warehouse</i> and then use <i>BI360 Reporting</i> to output reports/charts.                |
| Break-back analysis                       | Use one of Solver's special break-back templates (or design your own), to simulate items such as Net Profit and have BI360 auto-generate the related revenues and expenses based on for example historical trends. |

|   |   |
|---|---|
| Allocations   | Create the allocations model in Excel (can use the BI360 report module to create automated, parameter-driven allocation calculations), use BI360 <i>Planning</i> to store the allocated numbers to the BI360 <i>Data Warehouse</i> . From there, use the allocations in BI360 reports or transfer the allocated numbers back to the ERP system. |
| Electronic data entry of: <ul style="list-style-type: none"> <li>○ Statistics</li> <li>○ Key Performance Indicators (KPIs)</li> <li>○ Other organizational information</li> </ul> | Create input forms with Excel for any text or numerical information and use BI360 <i>Planning</i> to store it to the BI360 <i>Data Warehouse</i> . From there create reports with the BI360 <i>Reporting</i> , create OLAP cubes for analysis, or transfer the collected data to other systems.   |
| Commentary input for: <ul style="list-style-type: none"> <li>○ Monthly actual/budget variance reports</li> <li>○ Scorecard KPIs</li> <li>○ Quarterly/Annual reports</li> </ul>    | Add comment columns to your Excel-based financial statements or scorecards (e.g. generated dynamically with BI360 <i>Reporting</i> ), and use BI360 <i>Planning</i> to save the text to the BI360 <i>Data Warehouse</i> . Use BI360 <i>Reporting</i> to create reports that pulls the text comments back up from the database.                  |

## Typical Implementation Process

A typical implementation process of BI360 *Planning* would look like this (in this example assume that BI360 *Planning* is being used for budgeting):

1. Train power users.
2. Identify budgeting needs.
3. Preparation of the BI360 *Data Warehouse*.
4. Populate the *Data Warehouse* with historical figures and dimensions from your General Ledger (and if needed also from systems like Payroll and Fixed Assets). Examples of dimensions are: Account, Department, Project, Product, Employee, Asset, etc.
5. Enter any other dimensions or historical data directly in the *Data Warehouse* if they do not already reside in another database from which they can be imported.
6. Preparation of Excel budget templates.
7. Design your budget templates (or use BI360's pre-defined templates) with BI360 *Reporting*. In most cases, you can also use other Excel add-ins to produce your Excel templates. They will now be dynamic

and parameter driven based on the dimensions and data you populated to the *Data Warehouse* in step 1 above.

8. Manually created Excel budget workbooks will also work. As a best practice, make them as structured as possible and make sure all dimensions are listed somewhere.
9. *Planning Setup* (as explained earlier in this section).
10. Train end users.

# Dashboard



## BI360 – Dashboard Module

Thanks to its capability to calculate and store metrics and KPIs, BI360 can support any dashboard product that can read from a Microsoft SQL Server database (or OLAP cubes if those are created on top of the BI360 *Data Warehouse*). The BI360 suite offers its own dashboard module as well. It is based on Microsoft Silverlight and can run as a stand-alone in your web browser or it can be deployed as part of a Microsoft SharePoint portal.

The main purpose of BI360's *Dashboard* module is to support an organization's performance management initiatives by offering a relatively low cost and easy to implement, professional dashboard environment.

Some of the key reasons for implementing BI360's *Dashboard* module are:

1. Get managers and users focused on key metrics that drive organizational performance and processes.
2. Very user-friendly Microsoft-Silverlight-based interface that usually does not require training for end-users.
3. Relatively low cost and easy to implement.
4. Business users can quickly learn how to build custom dashboards and scorecards using the Dashboard Designer

As seen in the architecture diagram earlier in this document, the *Dashboard* module is one of four BI360 modules. The other three modules are:

1. BI360 *Data Warehouse* – Optional, this is for support of reporting across multiple ERP systems, etc.)
2. BI360 *Planning* module - Optional
3. BI360 *Reporting* module – Optional

In the architecture diagram earlier in this document, you saw how the BI360 *Dashboard* module interacts with the other BI360 modules. In

essence, it serves as a dashboard on top of the BI360 *Data Warehouse*, and the data can come from any source system. It can also connect to OLAP cubes or directly to transactional systems.

## User Interface Overview

Below is a sample screenshot of a BI360 *Dashboard*. Using the Dashboard Designer, the power user can create virtually any kind of dashboard layout. The interface was designed to maintain a similar look and feel to the Office products, including the use of an Outlook-style menu bar as the main navigation component.







## System Requirements

Depending on the components installed, the BI360 *Data Warehouse Manager* can be implemented on the same SQL server as the ERP. However, if warehouse reporting is intended to offload reporting directly from the ERP, it is recommended that a dedicated SQL server be used for all scenarios.

Performance metrics should be established once a baseline system is configured. There are many factors that contribute to the performance of the BI360 solution, including number of concurrent users, number of transactions, size and design of reports and input forms. For more details, please contact a certified BI360 partner or Solver.

| <b>Data Warehouse(Budget Database) SQL Server</b> |  |                         |
|---|--|-------------------------|
|   | <b>1-50 Concurrent Users</b>   | <b>50 or More Users</b> |
| <b>Operating System</b>                           | Windows Server 2008 [Any edition], (32-bit or 64-bit <sup>1</sup> )<br>Windows Server 2003 [Any edition], (32-bit or 64- | Same                    |

|                           |  |                                    |
|---------------------------|--|------------------------------------|
|                           | bit <sup>1)</sup> )  |                                    |
| <b>Database</b>           | SQL Server 2012 [Any edition], (32-bit or 64-bit <sup>1)</sup><br>SQL Server 2008 [Any edition], (32-bit or 64-bit <sup>1)</sup><br>SQL Server 2005 [Any edition], (32-bit or 64-bit <sup>1)</sup> | Same                               |
| <b>Processor</b>          | Two dual core /expandable  | Single/dual quad-core - Expandable |
| <b>Ram</b>                | 4 GB or greater  | 6 GB or greater                    |
| <b>Hard Disk</b>          | Based on data integrated into warehouse and/or transactional input within Budget model (contact Partner)   | Same                               |
| <b>Disk Configuration</b> | RAID 10 is recommended, Raid 1 minimum recommended.  | Same                               |

Notes:

1. 64-bit recommended

| <b>BI360 – Client Components</b>   |  |                    |              |  |
|------------------------------------|--|--------------------|--------------|--|
|                                    | <b>Reporting<sup>1</sup></b>   | <b>Planning</b>    | <b>DWM</b>   | <b>Dashboard</b>   |
| <b>Operating System</b>            | <ul style="list-style-type: none"> <li>• Win XP SP 3</li> <li>• Win Vista</li> <li>• Win 7 (32-bit or 64-bit)</li> <li>• Win Server 2003 (32-bit or 64-bit)</li> <li>• Win Server 2008 (32-bit or 64-bit)</li> </ul> |                    |              |  |
| <b>Ram</b>                         | • 2 GB (4GB Recommended) <sup>5</sup>  |                    |              |  |
| <b>Microsoft Office</b>            | <ul style="list-style-type: none"> <li>• Excel 2003<sup>2</sup></li> <li>• Excel 2007 (32/64 bit) <sup>4</sup></li> <li>• Excel 2010 (32/64 bit) <sup>4</sup></li> </ul>   | <same <sup>3</sup> | N/A          | N/A  |
| <b>.NET</b>                        | • 3,5 SP1 or newer   |                    |              |  |
| <b>Other software requirements</b> | Visual Studio Tools for the Office system 3.0 Runtime Service Pack 1   | <same              | Not Required | <ul style="list-style-type: none"> <li>• Silverlight 3.0 or higher</li> <li>• Explorer 7 or higher, Firefox 2 or higher, Safari 3 or higher</li> </ul> |

Notes:

1. Reporting will install a small repository database on the same server as the ERP and/or Warehouse.
2. Reporting Designer requires Office Excel 2007 or later.
3. BI360 Planning Administrative features require Office Excel 2007 or later. The end user input will work on Excel 2003. The installed Office version will be detected during the BI360 Planning install.
4. Drill Anywhere runs on 32bit Win OS/Excel only. For more details on Drill Anywhere or Excel Dicer, please contact Solver.
5. Terminal Server deployments: Please contact your BI360 partner or Solver for assistance.

| <b>BI360 Dashboard Web Server</b> |  |
|-----------------------------------|--|
| <b>Operating System</b>           | <ul style="list-style-type: none"> <li>• MS Windows Server 2008 [Any edition], (32-bit or 64-bit).</li> <li>• MS Windows Server 2003 [Any edition], (32-bit or 64-bit).</li> </ul> |
| <b>IIS</b>                        | <ul style="list-style-type: none"> <li>• IIS 6 or 7</li> </ul>   |
| <b>Processor</b>                  | <ul style="list-style-type: none"> <li>• 1.5 GHz or faster, 2.0 GHz dual-core recommended</li> </ul>   |
| <b>Ram</b>                        | <ul style="list-style-type: none"> <li>• 2 GB or more, 4 GB recommended</li> </ul>   |
| <b>Hard Disk</b>                  | <ul style="list-style-type: none"> <li>• Installation directory: 30 MB</li> <li>• Web application: 5 MB</li> </ul>   |

# Data Warehouse



## BI360 – Data Warehouse

BI360 *Data Warehouse Manager* (DWM) is BI360's configurable data warehouse solution. The DWM database is fully integrated with BI360's Excel-based *Reporting, Planning & Dashboard* modules. This enables immediate access to advanced financial and operational reporting capabilities as well as fully customizable budgeting, forecasting and data entry features. The *Data Warehouse Manager* is delivered with a simple to use desktop interface where administrators can configure and manage the *Data Warehouse Manager* without the need for technical skills.

## Data Warehouse Interface (Admin Console) Overview

The *Data Warehouse Manager* interface has been designed with the latest Microsoft technologies and it follows Microsoft best practices for interface design and usability. The interface was designed to maintain a similar look and feel to the Office products, including the use of a ribbon and Outlook-style menu bar as the main navigation components.

## Data Warehouse Manager Components

### Modules

The BI360 *Data Warehouse Manager* database has been pre-designed with a number of popular financial modules as well as five generic (user-defined) modules for any data type. However, the design is not limited to financial modeling; non-financial modeling can be handled just as well.

Pre-configured modules include:

- General Ledger
- Accounts Payable
- Accounts Receivable
- Capital (Asset Information)

- Payroll
- Projects
- Revenues
- Other X (Generic Modules 1 – 5)

To dramatically reduce implementation time, each module contains pre-defined fields specific to that module/area. The modules can be renamed using the *Data Warehouse Manager's* interface. For each module and without any need for customization, an unlimited number of User Defined Fields can be added using the Admin Console to further extend the capacity of the transaction tables.

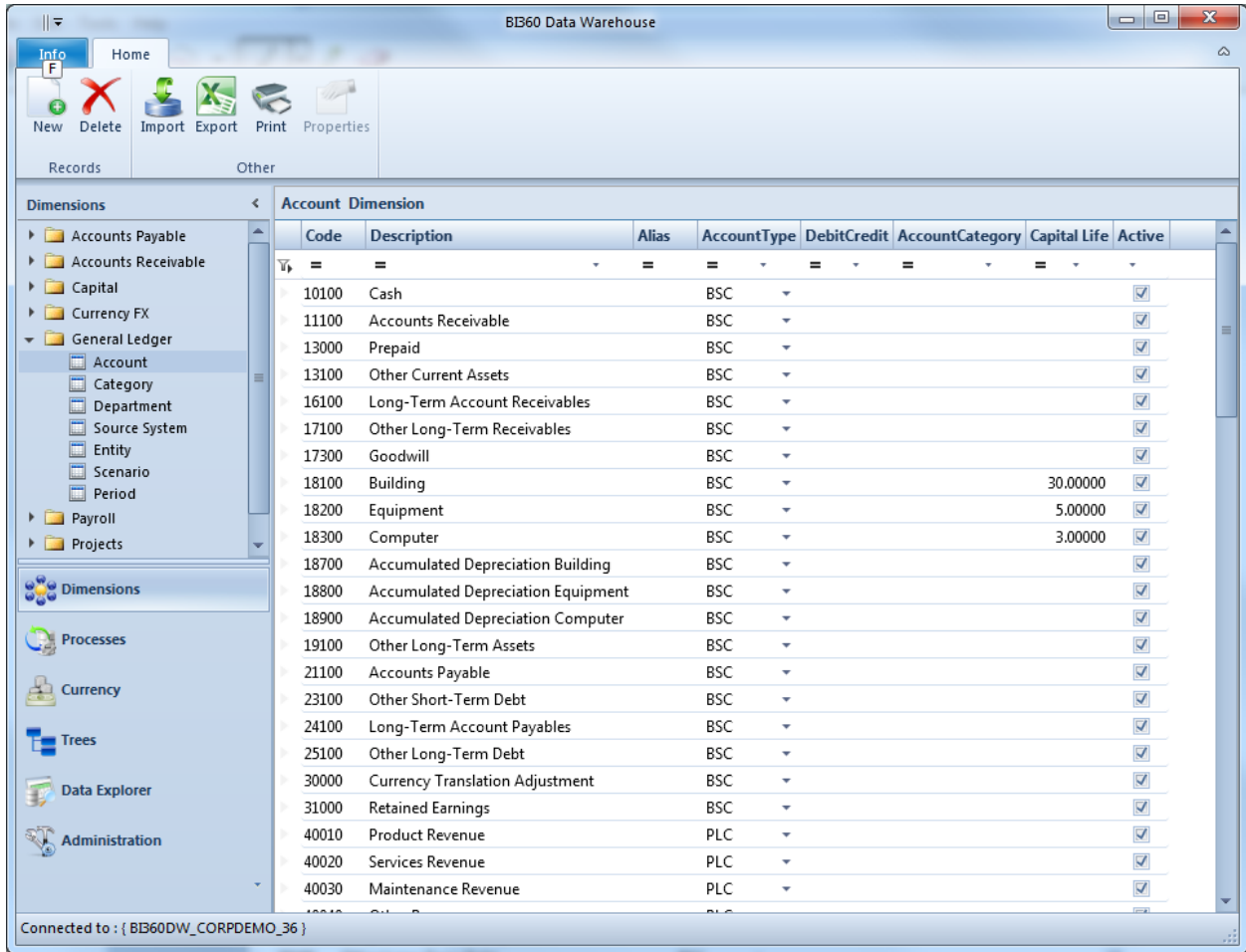
### Dimensions

The BI360 *Data Warehouse Manager* database contains dimensions specific to each module but also contains ten "Free" dimensions which can be shared across modules. Using the *Data Warehouse Manager* interface, the dimensions can be mapped to each module and labeled as needed. Each dimension contains many attributes specific to the dimension. Each of the ten free dimensions also contains several User Defined Fields (UDFs) to customize the dimensions to match a customer's specific data model. Up to 40 User Defined Fields can be added to each dimension using the Admin Console.

### Dimension Management

Dimensions can be managed within the *Data Warehouse Manager* interface. There are many ways to load the dimensions into the *Data Warehouse Manager*:

1. Direct import via the *Data Warehouse Manager* Interface import feature from a CSV file.
2. Automated SQL Server Integration Services (SSIS) integration import.
3. Manual entry: Manual entry is specifically useful for budgeting and forecasting. Since *Data Warehouse Manager* is a warehouse, Accounts and other dimensions that do not exist in the source system(s) can be created and maintained directly within the *Data Warehouse Manager* environment.

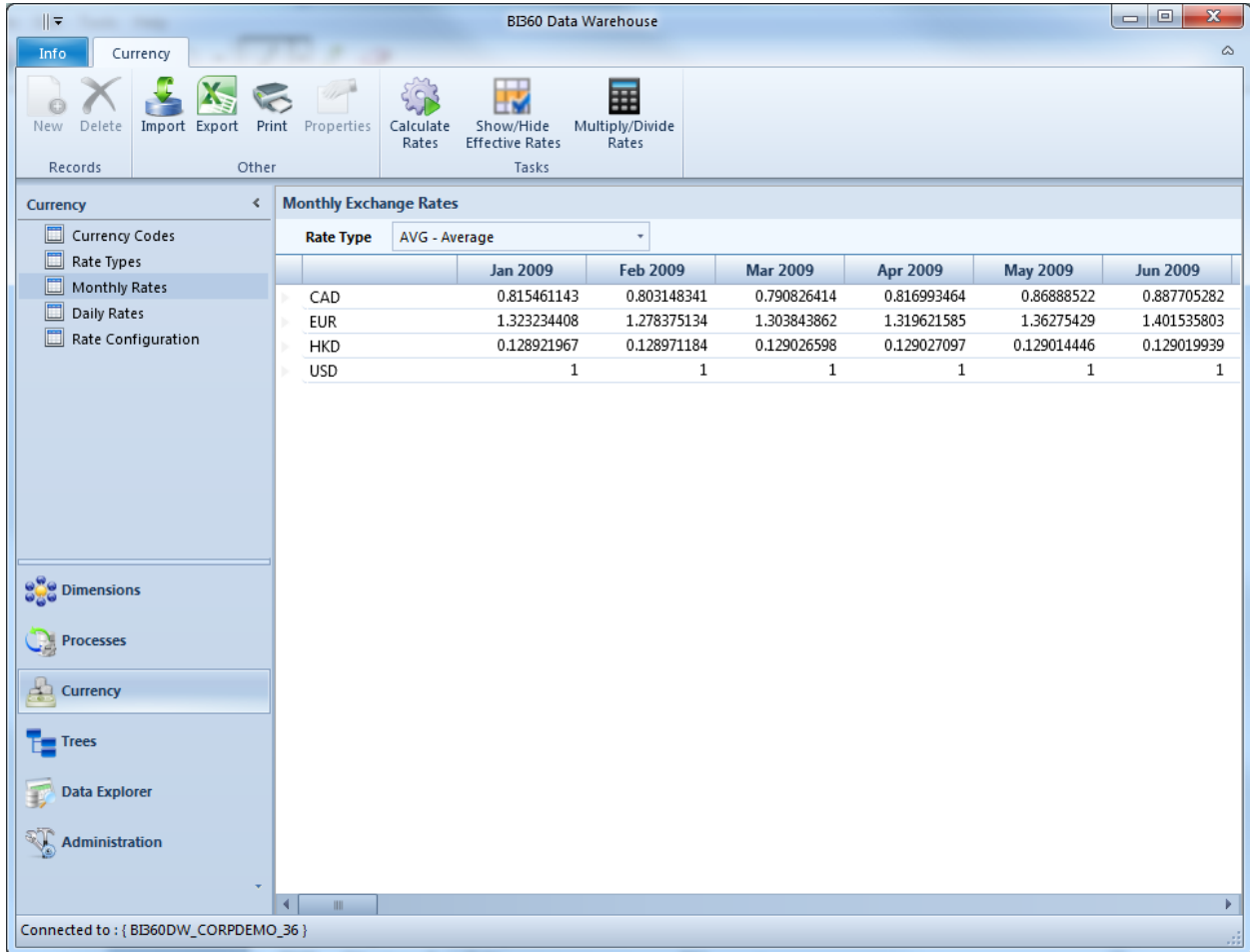


## Currency

The *Data Warehouse Manager* contains advanced currency features which work with the Business Rules to calculate foreign currency exchange. The tables accommodate daily rates, which can be loaded from a source system via SSIS integration or monthly rates can be directly entered into the *Data Warehouse Manager* interface. By combining several advanced backend features, the end user experience is very simple, yet advanced. The currency conversion process has several elements.

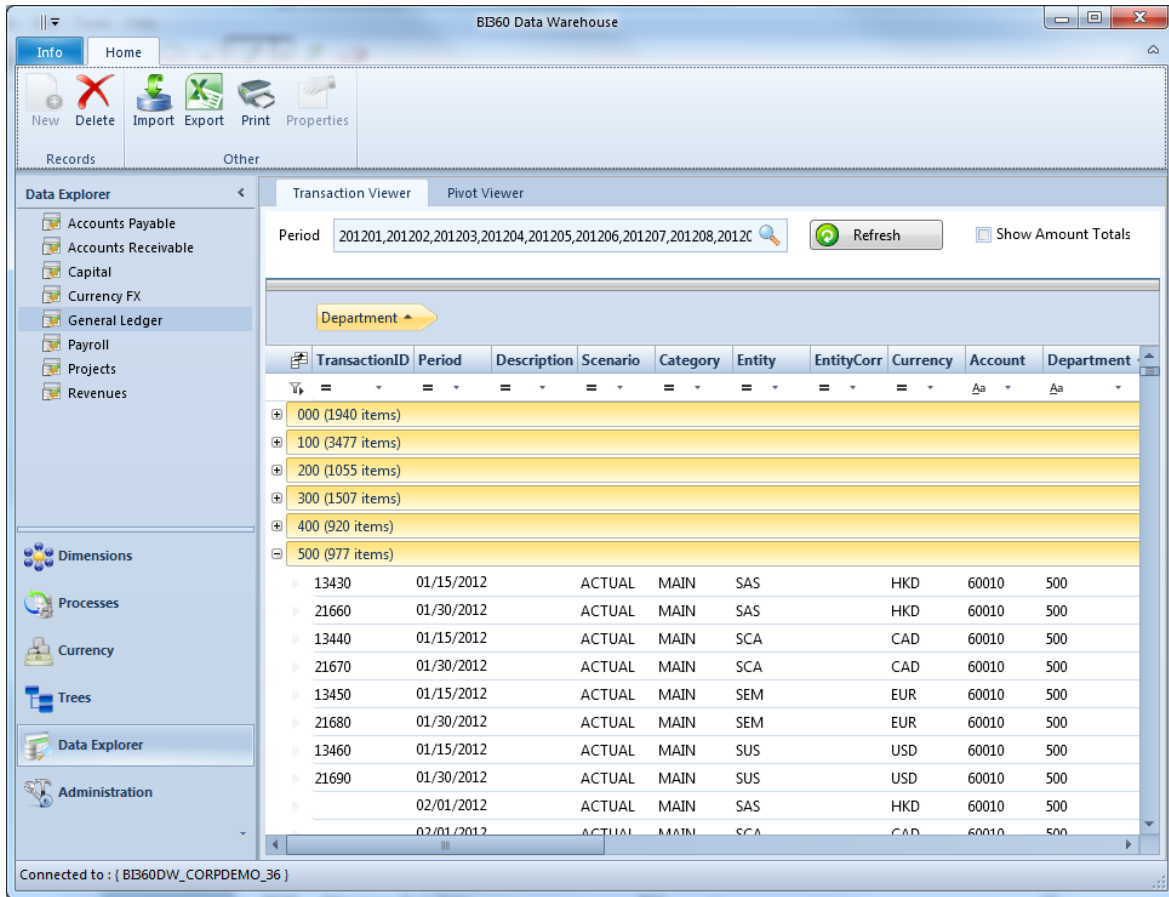
1. **Currency Codes:** Currency codes can be manually entered, directly imported or imported via SSIS integration.
2. **Rate Types:** Rate types can be manually entered, directly imported or imported via SSIS integration. Rate types such as Average, Closing, Historical, and Budget are typical.
3. **Daily\Monthly Rates:** If daily rates are loaded into the system, monthly rates can be automatically calculated. Otherwise, monthly rates can be manually loaded or imported.

- Rate Configuration: The configuration ties together the Account type, such as Balance Sheet, the Scenario such as Actual and the Rate Type such as Average.



## Data Explorer

The Data Explorer is used to filter, group and display data for each *Data Warehouse Manager* module so that an administrator, at any time, can see and validate what is stored in the *Data Warehouse Manager* without the need to perform SQL queries or running reports. The data can be viewed within the *Data Warehouse Manager* interface or exported to Excel.



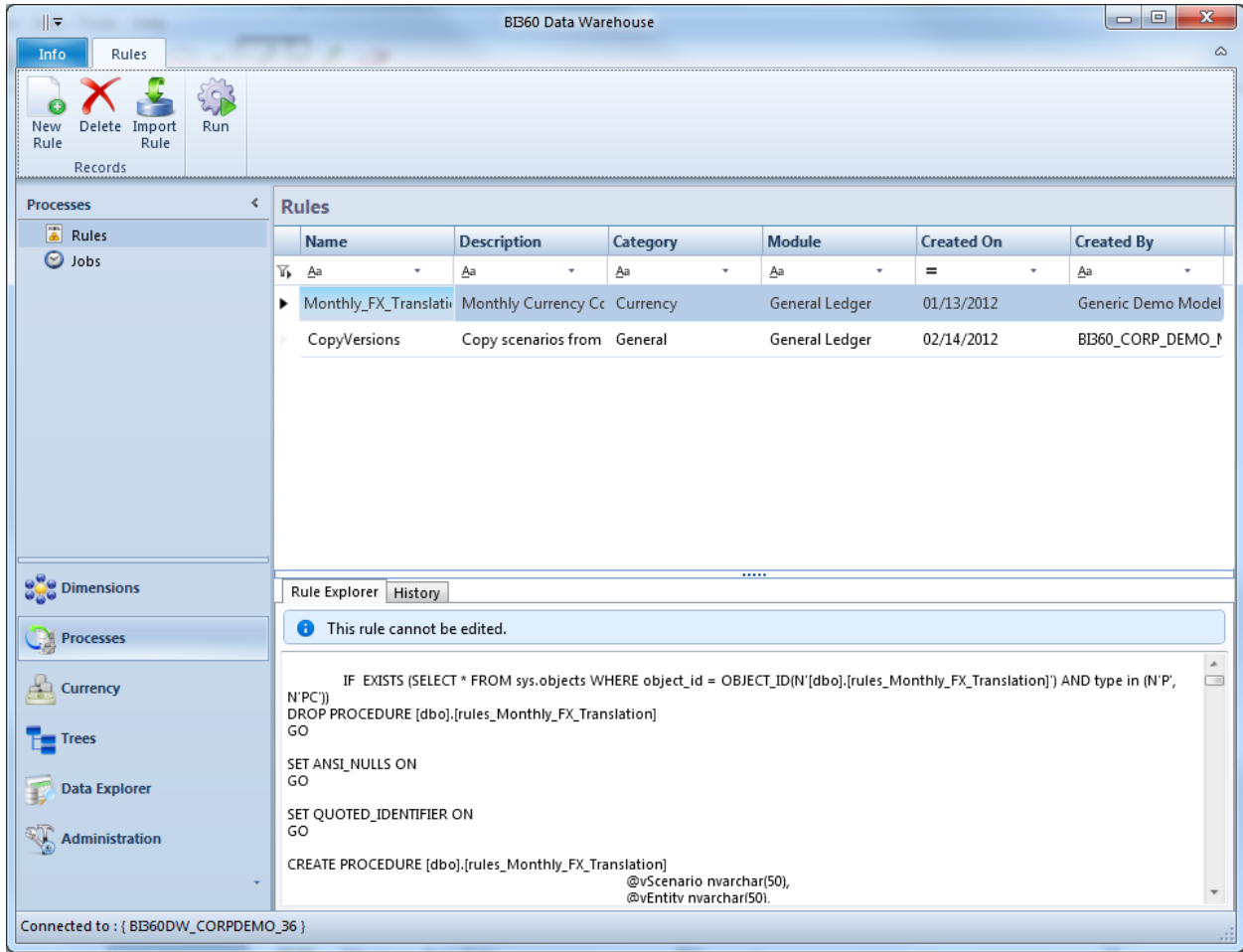
## Processes

The Processes are SQL procedure-based rules which enable data transformation and movement within the *Data Warehouse Manager*. Because the rules are created as stored procedures within the *Data Warehouse Manager*, the rules can be accessed and/or scheduled using external tools, such as the SQL Server Agent. Rules can be loaded into the warehouse via an XML file. The XML file contains information about the parameters used to execute the rule as well as the SQL for the data manipulation.

Practical applications of rules include:

- Currency conversion
- Data aggregation
- Entity eliminations
- Allocations
- Data movement
- Other transformations





## Practical Usage Examples

The following are examples of various practical uses of BI360's *Data Warehouse*:

| Data Warehouse   | Usage Examples   |
|--|--|
| Multi-source consolidation warehouse   | <ul style="list-style-type: none"> <li>• Load data from any ERP</li> <li>• Load data from external system (ADP, FAS, Ceridian, etc.)</li> <li>• Store multiple scenarios of budgets &amp; forecasts via <i>Data Warehouse Manager</i> or any other Interface.</li> </ul> |
| Instant Excel-based reporting, budgeting and modeling with BI360's <i>Reporting and Planning</i> modules | <ul style="list-style-type: none"> <li>• <i>Data Warehouse Manager</i> pre-defined integration to <i>Reporting and Planning</i> enables immediate reporting and budgeting to/from all <i>Data Warehouse Manager</i> modules.</li> </ul>                                  |

|   |  |
|---|--|
| Dimension & Data Integration                    | <ul style="list-style-type: none"> <li>• File Import/Export capabilities for all dimensions and data modules.</li> <li>• SSIS Import/Export capabilities to and from any system accessible via SQL Server Integration Services.</li> </ul>   |
| Customize Modules, Dimensions & attribute names | <ul style="list-style-type: none"> <li>• Labels can be modified to fit with the organizations requirements and to enhance the BI user experience.</li> </ul>   |
| Exchange Rate Administration                    | <ul style="list-style-type: none"> <li>• Import Daily spot rates from ERP via SSIS</li> <li>• Automatically calculate monthly Average and closing rates as well as any other user configured rates.</li> <li>• Enter monthly rates directly into the database (when daily rates are not loaded)</li> </ul> |
| Business Rules Engine                           | <ul style="list-style-type: none"> <li>• Complex currency calculations for multi-national organizations</li> </ul>   |
| Data Viewer                                     | <ul style="list-style-type: none"> <li>• Filter, view and validate data loaded into each module.</li> </ul>  |

## Typical Implementation Process

A typical implementation process of the *Data Warehouse Manager* would encompass the following:

1. Installation of the *Data Warehouse Manager* pre-configured SQL Server database.
2. Installation of *Data Warehouse Manager* desktop application on the administrators' machines.
3. Determine source systems and data to be integrated with each system and map to *Data Warehouse Manager* modules and dimensions (see below).
4. Configure *Data Warehouse Manager's* Period table and customize Module names, dimension names as requisite (or leave default).  
*Note: Custom labels will also be visible in Reporting, Planning & Dashboard modules.*
5. Configure currency (as requisite). Determine if daily rates will be loaded from external system or monthly Average/Closing, etc. rates will be loaded directly into the currency table. If it set up as an automated process, then configure the SSIS Integration to update daily spot rates in the *Data Warehouse Manager*.
6. Determine import method (automated with SSIS or manual imports):
7. SSIS: Configure SSIS packages [E.g. to Microsoft Dynamics] to previously determined module & dimension mappings. Or implement custom data integration to any data source.

8. Import: Manually import dimensions and data
9. Business Rules: If business rules, such as Currency Conversion are to be used, ensure step #5 is completed. No additional steps required, other than running rule.
10. Document integration decisions.
11. Quick training required (Walk through can be done in 1-2 hours).

## Module – Dimension Mapping

Each module contains module specific dimensions as well as ten free dimensions, which can be shared across all modules. Each module also contains several other standardized dimensions, e.g. Entity, Currency, Scenario, Category, etc. Additionally, each module contains many attributes, both system defined and user defined, including three amount fields, four text comment fields as well as other module specific attribute information.

| Module                     | Unique Dim | Unique Dim   | Unique Dim   |
|----------------------------|------------|--------------|--------------|
| <b>General Ledger</b>      | Account    |              |              |
| <b>Account Payable</b>     | Vendor     |              |              |
| <b>Accounts Receivable</b> | Customer   | Sales Person |              |
| <b>Capital</b>             | Asset      |              |              |
| <b>Human Resources</b>     | Employee   |              |              |
| <b>Project</b>             | Project    | Customer     | Item         |
| <b>Revenue</b>             | Customer   | Product      | Sales Person |
| <b>Other</b>               | Other      |              |              |