

KDCalc Designer User's Guide

For Developers and Business Users



KDCalc™ (Patents Pending)

Excel-Compatible Spreadsheet Add-In for
Java and .Net

Version 4.2.x

Knowledge Dynamics, Inc.

“The Killer-App for Software Development”™

Note: See the ***KDCalc Java/.NET Engine User's Guide*** for
Programming Information

Welcome to KDCalc™ V4.2.x by Knowledge Dynamics®, Inc. (Patents Pending)

“The Killer-App for Software Development”™

What is It?

KDCalc is a Microsoft Excel® compatible spreadsheet engine for Java and .Net.

Why do I need it?

KDCalc turns your Excel workbooks into enterprise applications for Java and .NET. With KDCalc, non-programmers can design, build, and test complex logic in Excel, then turn it into Web applications of all kinds. KDCalc generates enterprise class calculation engines and JSP, ASP.NET, and HTML user interfaces. It can also be paired with 3rd party grid and graph controls. KDCalc supports ‘Spreadsheet Programming’, where spreadsheets are used in transaction processing or other server side-processing. KDCalc can be deployed to any platform with a JVM or the .NET CLR. Excel is not needed on the end-user’s machine.

What do I get?

KDCalc includes an Excel plug-in that compiles Excel workbooks into a compressed executable, a runtime engine that supports most of Excel's

functions, and an API that allows consistent access to your spreadsheets’ functionality.

How do I use it?

The KDCalc plug-in installs right into the Microsoft Excel® toolbar. Click a few buttons to generate your application.

Access the spreadsheet data and calculations through simple API calls. Put values into data cells and get results from formula cells. Navigate Sheets by Name or Index. Access Cells and Ranges by Name or by Row, Col.

System Requirements

The KDCalc V4.2.x Designer requires the following:

- Microsoft Windows, 98, NT, 2000, or XP
- Microsoft Excel 2000 or XP for Windows
- 64MB RAM
- 15MB Free Hard Drive Space

The KDCalc Engine requires a JVM 1.1 or higher, or the .Net CLR 1.0 or higher.

Licensing

The following is a brief overview of the KDCalc V4.2.x licensing terms. See the attached End-User License Agreement for the full license terms.

For development KDCalc is licensed *per developer and per business user*. Each Developer that writes code that will directly or indirectly call the KDCalc runtime engine and each business user who maintains the spreadsheets must obtain a license. A Licensee is not permitted to generate modules for other persons who have not properly obtained a license.

For individual deployment the KDCalc Engine is royalty-free for deployment in interactive applications that execute in personal computer environments, such as stand-alone applications or applets in a web browser. Batch data processing applications are considered to be server deployments (see below), even if running on individual PCs or workstations.

For server deployment KDCalc Engine Server Edition is licensed *per CPU* for execution in server environments such as Web Services, JSP, or ASP.NET.

For embedded systems deployment KDCalc Engine Embedded Edition is licensed *per application* for deployment in applications that execute in embedded systems environments such as PDAs, embedded microprocessors, cell phones, etc.

Table of Contents

Table of Contents.....	2
Installation and Registration	1
Development System Requirements.....	1
Java Execution System Requirements	1
.Net Execution System Requirements	1
Installation	1
Registration	1
Introduction	2
Using the Generate Web UI Wizard	3
Developing with KDCalc (The Process).....	8
Using the KDCalc Designer Dialog	9
Custom Templates.....	15
Color Schemes	17
Defining Color Schemes	18
Feature Support Notes.....	20
KDCalc V4.2.x Supports These Excel Features	20
KDCalc V4.2.x Doesn't Support These Excel Features.....	20
Function Reference.....	21
Error Reference	22
Compilation warnings:	22
Compilation errors:.....	22
Best Practices	23
Frequently Asked Questions.....	23
End User License Agreement.....	23

KDCalc and Knowledge Dynamics are trademarks of Knowledge Dynamics, Inc. All other trademarks and Service marks are property of their respective owners.

Installation and Registration

Development System Requirements

The KDCalc V4.2.x Designer requires the following:

- Microsoft Windows 95, 98, NT, 2000, or XP
- Microsoft Excel 2000 or XP for Windows
- 64MB RAM
- 15MB Free Hard Drive Space

Java Execution System Requirements

The KDCalc V4.2.x runtime engine requires the following for Java deployment:

- Java JRE 1.1 or higher

.Net Execution System Requirements

The KDCalc V4.2.x runtime engine requires the following for .NET deployment:

- Microsoft .NET Framework version 1.0 or higher
- Visual J# .NET Redistributable Package

Installation

To install KDCalc V4.2.x, make sure Excel is not running and simply double-click on 'setup.exe'. It installs like any other Windows application.

Registration

The KDCalc Excel Plug-In installs as a size-limited demo. It will only compile cells A1-E10 on the first 2 Worksheets. KDCalc must be registered to unlock its full functionality.

To register KDCalc, click the 'About / Register' button on the main KDCalc window. Enter your License Code and registration information. An Internet connection is required.

Introduction

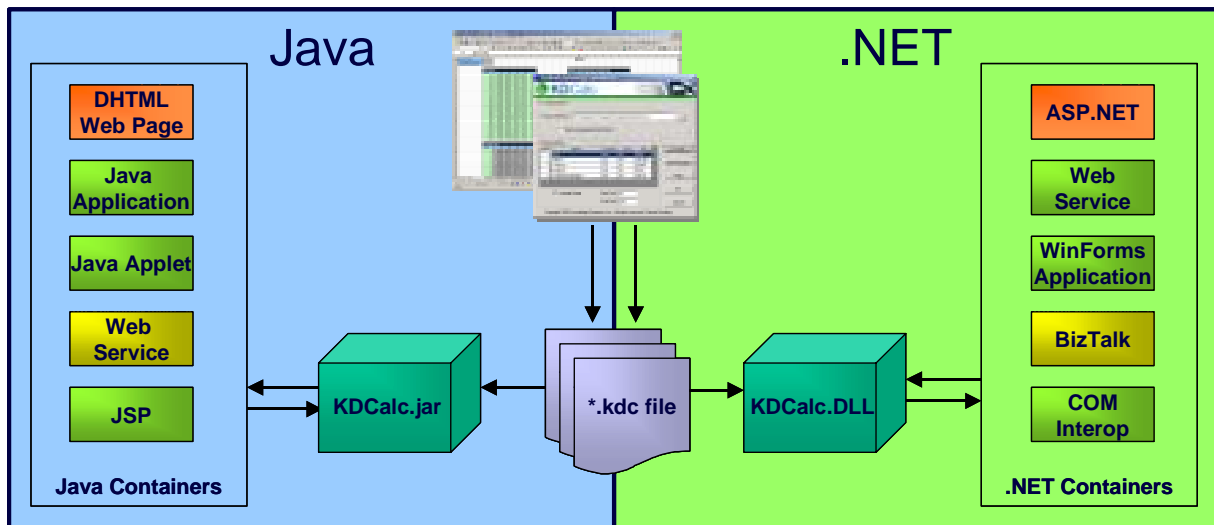
Banks, insurance companies, and all kinds of businesses have come to depend on Microsoft Excel for business decisioning, pricing algorithms, forecasting, and many other kinds of business logic. KDCalc turns these mission-critical spreadsheets into mission-critical applications suitable for use by individual end users or across the enterprise.

KDCalc converts your cell formulas and data into calculation engines that can be executed on Java and .NET without Excel. These calculation engines are fast, scalable, and secure. KDCalc can handle transactions volumes and parallel processing that goes far beyond Excel's capacity.

KDCalc can also generate a user interface from your spreadsheet. KDCalc can create ASP.NET, JSP, and HTML interfaces for end users to interact with your application.

KDCalc

The Only Excel Compatible Spreadsheet Engine for Java and .NET



- The KDCalc Designer turns Excel Workbooks into executable .KDC files. .KDC files are compatible with both the KDCalc Java and KDCalc.NET engines.
- The KDCalc Engines can be integrated into any type of client or server container for scalable, lightweight runtime execution. Excel is not required at runtime.
- The KDCalc Designer can optionally generate ASP.NET, JSP, and HTML web pages to act as the User Interface for your application.

The KDCalc Designer has two main functions: calculation engine generation and web user interface generation. For years KDCalc has been used by leading businesses around the world to generate calculation engines (.kdc files) for web services, batch transaction processing, and client/server applications. If the application required a user interface, developers could create rich UIs in Java, .NET, or even Flash, using KDCalc's robust API.

All of that functionality still remains, but with the release of KDCalc 4.0, KDCalc can also automatically generate HTML, JSP, and ASP.NET user interfaces from your Excel workbooks. This enables faster deployment of web applications while making KDCalc even easier for non-programmers.

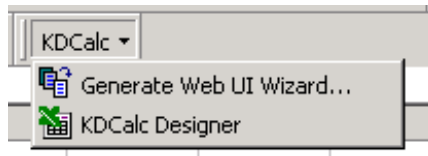
Beginners can use the KDCalc Wizard to generate web applications. Once you've become familiar with KDCalc, you can use the KDCalc Designer to exercise complete control over your web app.

Using the Generate Web UI Wizard

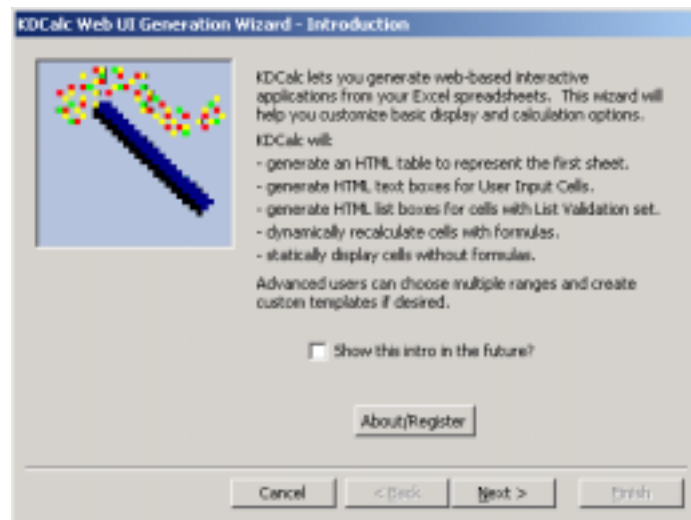
The Generate Web UI Wizard allows beginning KDCalc users to quickly create complete web applications from their Excel workbooks.

KDCalc installs as a button in the Excel toolbar. Clicking on "KDCalc" drops down two options: "Generate Web UI Wizard..." and "KDCalc Designer" as pictured below. Once "Generate Web UI Wizard..." is clicked it will pop-up the wizard's introductory screen.

Plugin:



Introduction:

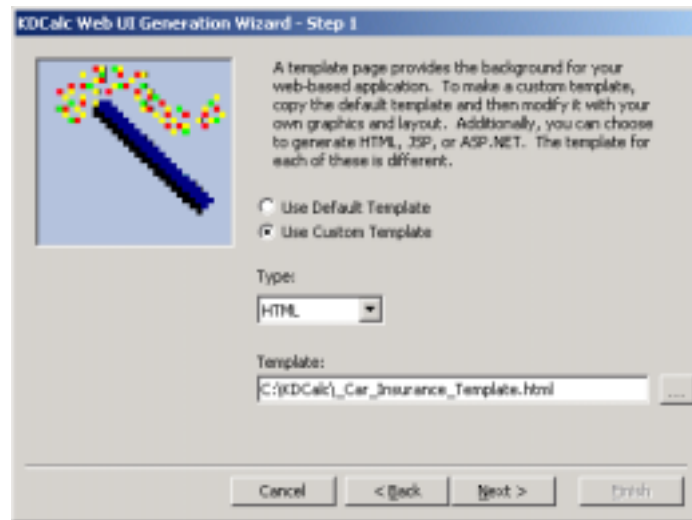


The introduction screen gives the user some basic information on how KDCalc will generate the resulting web application. You can choose to skip this screen in the future.

About / Register

Clicking 'About / Register' activates a window where you can register KDCalc or update your registration settings. KDCalc must be registered for full functionality. An Internet connection is required for registration.

Step 1: Configure Template and Application Type



Use Default/Custom Template:

KDCalc generates web UIs from Template files. A template file is shell JSP, ASP.NET, or HTML page. KDCalc will insert your compiled workbook into the template to create a new web UI. You can choose to use one of KDCalc's default templates, or you can create custom templates with your own graphics, buttons, links, and other features. Choose whether you want to use the default template or supply a custom template.

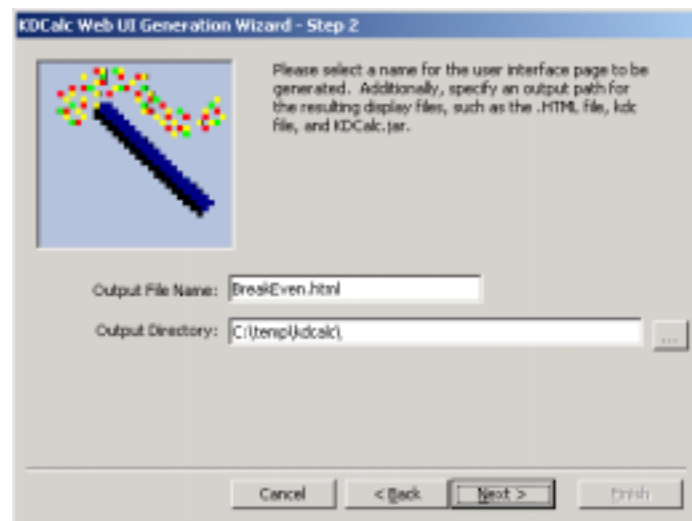
Web Application Type:

Choose the type of web application to create: HTML, JSP, or ASP.NET.

Template:

If choosing to use a Custom Template, select the location of this template.

Step 2: Configure Output



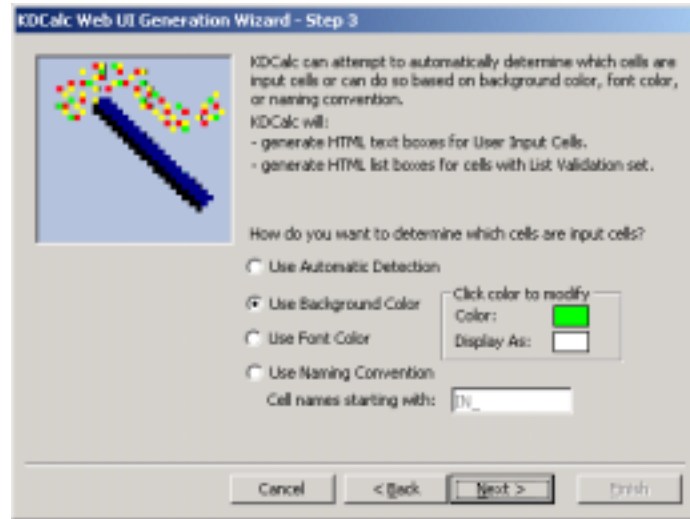
Output File Name

Specify the name of the resulting web application HTML, JSP, or ASP.NET file.

Output Directory

Specify the output directory for the web application to be written to.

Step 3: Configure Input Cells



The KDCalc Web Application Generator needs to know which cells represent user input cells. KDCalc can generate two kinds of user input controls for your web app: text boxes and combo-boxes. Regular user input cells will be turned into text boxes in your web app. If any of your User Input Cells have List Validation, they will be turned into drop-down boxes in the resulting web application (see Data->Validation in the Excel menu and look at 'List' in the 'Settings' tab for more info).

Choose the option that is most convenient for your spreadsheet:

- **Use Automatic Detection** – KDCalc will automatically determine which cells are input cells based on cell formula dependencies.
- **Use Background Color** – When creating the spreadsheet, choose a unique background color for input cells. All cells of this color will be assumed to be input cells. You can choose to display these cells in the same color in your web app, or you can change their runtime color by changing the "Display As:" color.
- **Use Font Color** – Same as background color, but looks at cell font color instead of background color.
- **Use named cells starting with __** – If you name cells (see *Insert->Name->Define* in Excel's menu) using a naming convention such as all input cells start with "IN_". KDCalc can use this to determine which cells are input cells.

Step 4: Set Web Application Options



Display Gridlines

If checked, will display light-gray gridlines for the borders of any cells not specifying borders.

Display Row/Col Headers

If checked, this will display A1-Style column and row headers.

Input Cell Border

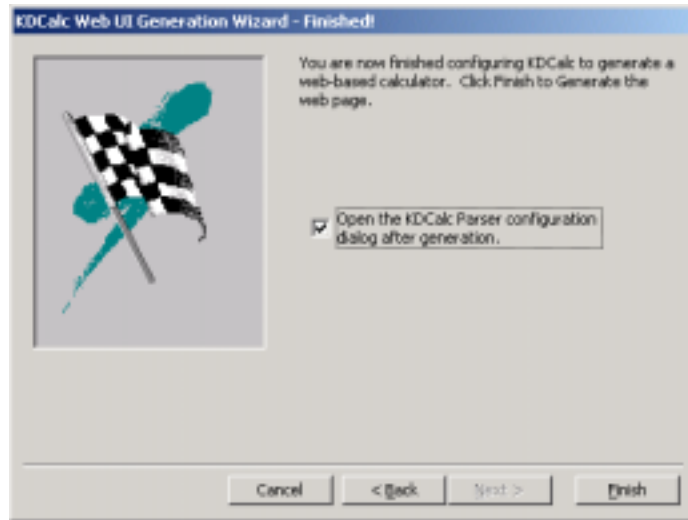
Choose the border for input cells. Options include the traditional 3D look of input text boxes, a Flat look with a single line border, or None for no border.

Table Width

Choose to change the width of the table produced:

- Automatic – tries to mimic the width of the used area on the first sheet.
- Fixed pixels – locks the width size in to a chosen pixel width.
- Fixed percentage – calculates a percent of the size of the used area on the first sheet.

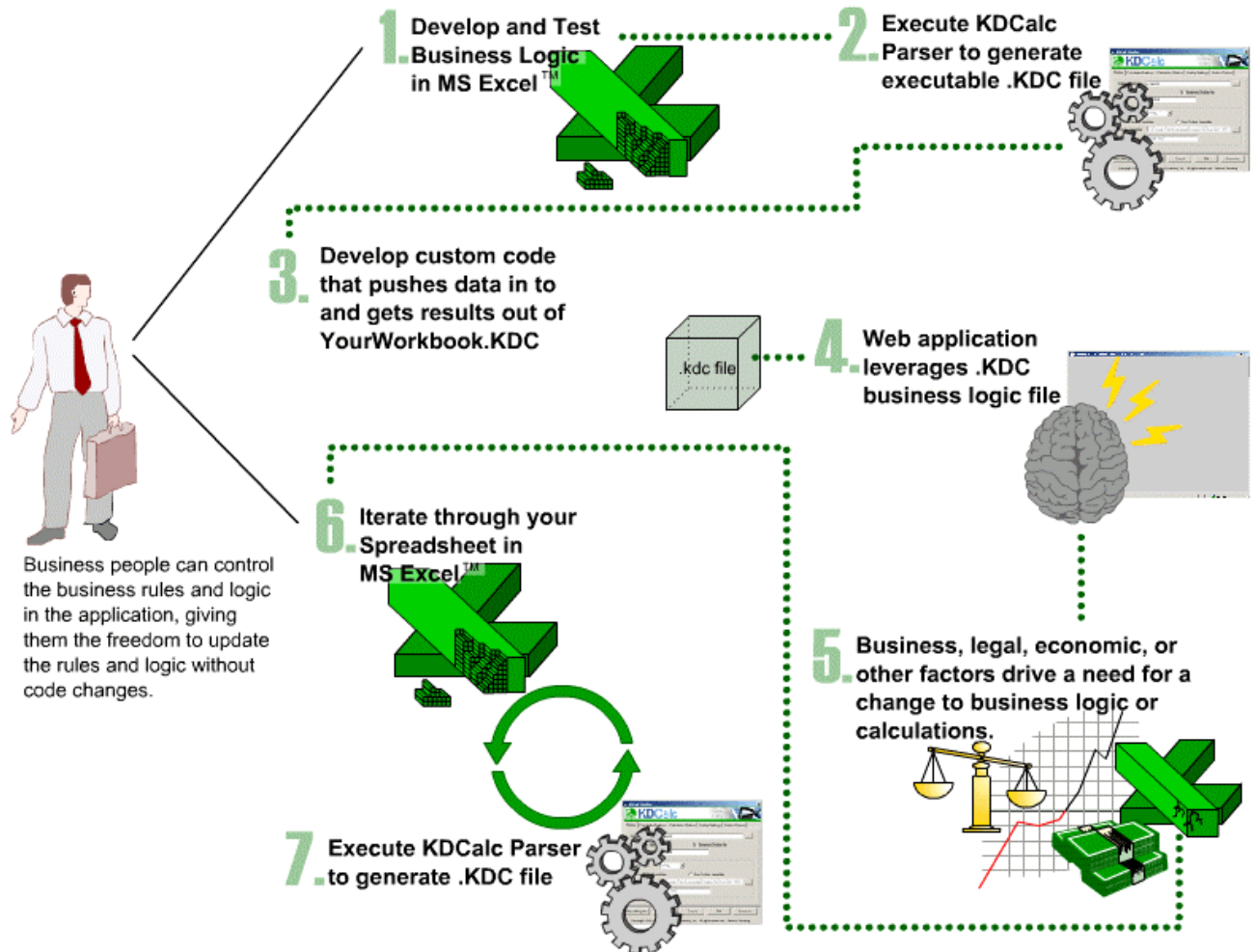
Step 5: Finish and Generate



After clicking finish on the Wizard's final screen, KDCalc will generate the web application as configured. You can choose whether or not to open the KDCalc Designer Configuration Dialog after generation is complete. The KDCalc Designer Configuration Dialog allows you to set more detailed configuration properties for finer control over web application generation.

Developing with KDCalc (The Process)

Developing Applications with KDCalc quick and easy as illustrated below:

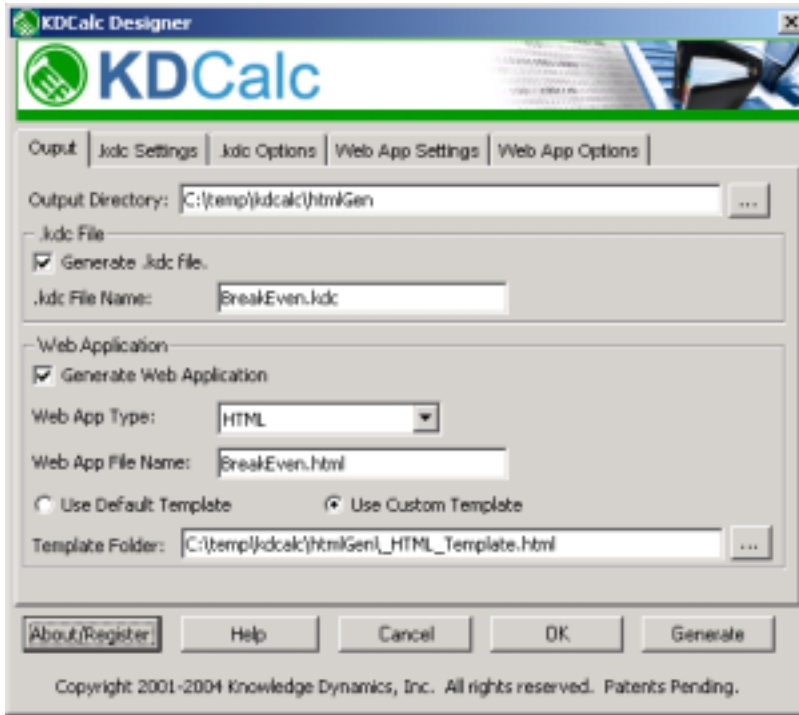


1. KDCalc lets you use the familiar, productive environment of Microsoft Excel to develop complex calculations and data transformations.
2. With a few mouse clicks, KDCalc turns your Excel cell formulas into a compressed, compiled executable format. The business rules in the spreadsheet are duplicated in the .KDC file with no chance of miscommunication between designers and developers.
3. All you have to write is the plumbing code that moves data in and out of KDCalc cells. This might be a GUI that gets data from the user, pushes it into the KDCalc Engine, pulls the results from the KDCalc Engine, and displays the results in a graph or table. It could also be an OLAP application running on a server, extracting data from a database and publishing the results to a web report. The 'setCellValue' and 'getCellValue' API methods make this integration simple. The list of potential applications is endless.
4. Once the application is built, business people can modify and enhance the spreadsheet in Excel and as often as desired.
5. The new business rules can be deployed to production without any code changes or recompilation of the application. The engine is small (about 100K) and support over 200 Excel functions. .KDC files are generally less than ¼ the size of the original .XLS file.

Using the KDCalc Designer Dialog

Start the KDCalc Designer dialog by clicking the 'KDCalc Designer' button in the Excel toolbar. The KDCalc Designer dialog pictured below will appear. The KDCalc Designer dialog allows you to generate calculation (.kdc) files and Web Applications with more options than the wizard.

Tab 1: Output Options



Output Directory

Specify the output directory for the generated calculation (.kdc) file.

.kdc File Section

Files with the .kdc extension generated by KDCalc are calculation files. These files contain all of cell information and calculation logic for the KDCalc Engine.

Generate .kdc file

Choose whether to generate a calculation (.kdc) file or not. This should remain checked in almost all circumstances. Only uncheck this if you have changed formatting in your spreadsheet that affects the user interface but not the calculations.

.kdc File Name

Specify the name of the .kdc file to be generated.

Web Application Section

KDCalc can generate HTML, JSP, ASP, and ASP.NET web applications.

Generate Web Application

Choose whether to generate a Web Application. This should only be checked if you plan to use KDCalc to generate a web-based application in HTML, JSP, ASP, or ASP.NET. If this option is unchecked, the associated properties are disabled and can be ignored.

Note: If you just plan on calling the KDCalc Engine's API directly, this option should be unchecked.

Web App Type

Choose to generate the web application as HTML, JSP, or ASP.NET.

Web App File Name

Specify the name of the web application's primary display file.

Use Default / Custom Template

KDCalc generates web applications from files stored in a Template folder. A template folder contains images, javascript, and JSP, ASP.NET, or HTML pages. You can choose to use one of KDCalc's default templates, or you can create custom templates with your own graphics, buttons, links, and other features.

Template Folder

If you choose to use a Custom Template, enter folder path and name in this field by typing in the name or browsing to the file.

In creating a custom template it is advised to copy and rename an existing template folder and then make modifications such as adding graphics to the original template. More information can be found in the **Custom Templates** section.

Tab 2: .kdc Settings

The table below displays each sheet's range for encoding into the calculation (.kdc) file. Select one or more of the rows to modify their properties.

Index	Name	Include	Start	End	Synch
1	Interface	Yes	A1	L43	Yes
2	Analysis	Yes	A1	BE216	Yes
3	Interface - old	No	A1	V51	Yes
4	UI Interface - big	No	A1	V51	Yes

☒ Include Sheet Start Cell:

☒ Keep End Cell synchronized with used area. End Cell:

Note: A "**" in the Index column means the End Cell does not match the sheet's last used cell.

Copyright 2001-2004 Knowledge Dynamics, Inc. All rights reserved. Patents Pending.

Sheet Settings

You can select one or more sheets at a time and make changes to the following properties:

Include Sheet

Indicates whether the worksheet will be compiled or not. Unchecked worksheets will be still be included in the file, but all of their cells will behave as if they were empty in the original Excel spreadsheet. At runtime, data values can still be programmatically entered into cells on sheets that were excluded at compile time.

Cells on included worksheets that refer to excluded names will have invalid formulas and will return errors at runtime.

Start Cell, End Cell

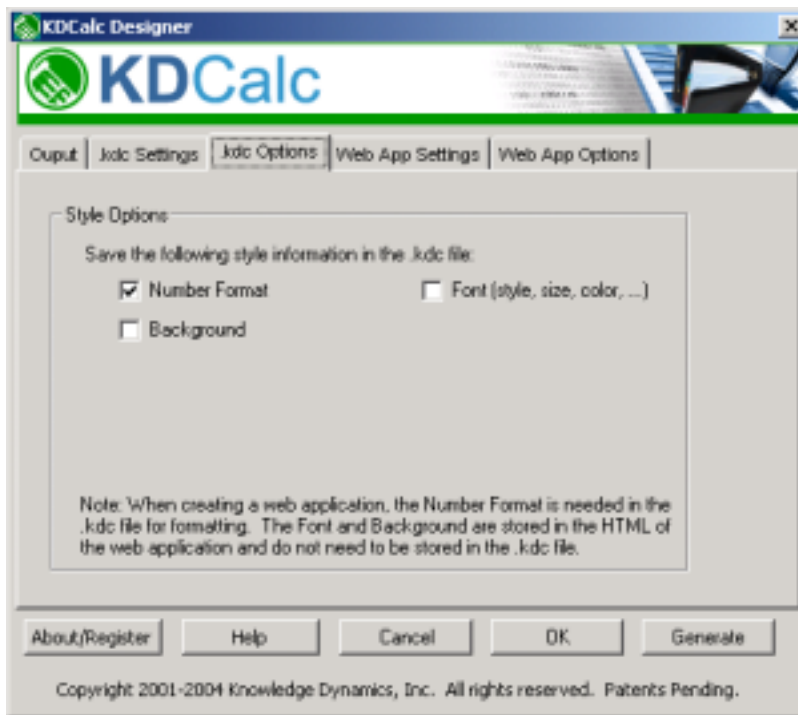
The Start Cell and End Cell specify the Top-Left and Bottom-Right cells of the parse area of included cells for each worksheet. Cells in this range will be compiled into the .KDC file. Cells outside of this range will be ignored. The End Cell defaults to each sheet's last used cell.

At runtime, cells outside of the parse area behave as if they were empty in the original spreadsheet. They are accessible, but will remain empty unless you put data in them programmatically.

Keep End Cell synchronized with used area

If this is checked, KDCalc, keeps the End Cell for the selected sheet(s) equal to the sheet's last used cell. If this is unchecked and the last used cell is different from the cell specified in the End Cell field, a “*” is placed by the sheet's index number and a warning is displayed.

Tab 3: .kdc Options



Style Options

The calculation file can store information pertaining to the style of cells. This information can be accessed through the KDCalc Engine's API. Currently, the **Number Format**, **Background**, and **Font** information can be stored in the calculation file.

Number Format is checked by default and is valuable when creating web applications so that KDCalc can display cell values using the correct format. Background and Font information only needs to be included if you plan to access the information through the API at runtime.

Tab 4: Web App Settings

Choose the range to be displayed in the user interface. This range can be different than the range included in the calculation file (.kdc).

Index	Name	Include	Start	End	Synch
1	Interface	Yes	A1	L43	Yes
2	Analysis	Yes	A1	BE216	Yes
3	Interface -old	Yes	A1	V51	Yes
4	UI Interface - big	Yes	A1	V51	Yes

☒ Include Sheet Start Cell:

☒ Keep End Cell synchronized with used area. End Cell:

Note: A "*" in the index column means the End Cell does not match the sheet's used area.

Buttons: About/Register, Help, Cancel, OK, Generate

Copyright 2001-2004 Knowledge Dynamics, Inc. All rights reserved. Patents Pending.

Sheet Settings

You can select one or more sheets at a time and make changes to the following properties:

Include Sheet

Indicates whether the web application will contain the sheet or not.

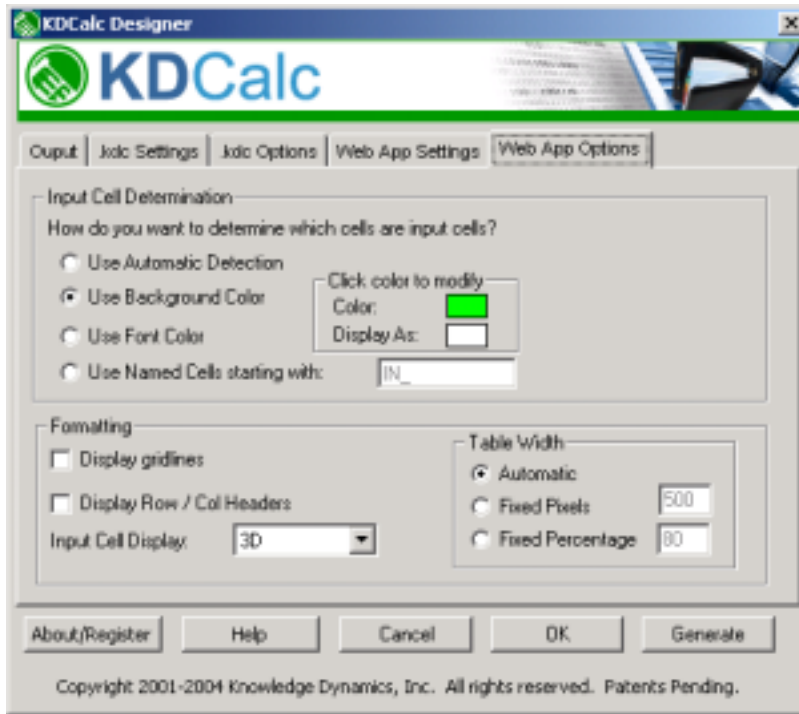
Start Cell, End Cell

The Start Cell and End Cell specify the Top-Left and Bottom-Right cells of the sheet to include in the display area. User interface code for the appropriate type of web application will be generated for cells in this range, while cells outside of this range will be ignored. The End Cell defaults to the sheet's last used cell. **NOTE:** This range is different from the range specified in the Calculation Settings and is more akin to the "Print Area". You will probably want to hide much of your calculations from end users and only display a part of your workbook. This feature lets you control what the users see.

Keep End Cell synchronized with used area

If this is checked, KDCalc keeps the End Cell for the selected sheet(s) equal to the sheet's last used cell. If this is unchecked and the last used cell is different from the cell specified in the End Cell field, a warning is displayed.

Tab 5: Web App Options



Input Cell Determination

The KDCalc Web Application Generator needs to know which cells represent user input cells. KDCalc can generate two kinds of user input controls for your web app: text boxes and combo-boxes. Regular user input cells will be turned into text boxes in your web app. If any of your user input cells have List Validation, they will be turned into drop-down boxes in the resulting web application (see Data->Validation in the Excel menu and look at 'List' in the 'Settings' tab for more info).

Choose the option that is most convenient for your spreadsheet:

- **Use Automatic Detection** – KDCalc will attempt to automatically determine which cells are input cells based on cell formula dependencies.
- **Use Background Color** – When creating the spreadsheet, choose a unique background color for input cells. All cells of this color will be assumed to be input cells. You can choose to display these cells in the same color in your web app, or you can change their runtime color by changing the "Display As:" color.
- **Use Font Color** – Same as background color, but looks at cell font color instead of background color.
- **Use named cells starting with** – If you name cells (see *Insert->Name->Define* in Excel's menu) using a naming convention such as all input cells start with "IN_". KDCalc can use this to determine which cells are input cells.

Display Gridlines

If checked this will display light-gray gridlines for the borders of any cells not specifying borders.

Display Row/Col Headers

If checked this will display A1-Style column and row headers.

Input Cell Border

Choose the border for input cells. Options include the traditional 3D look of input text boxes, a Flat look with a single line border, or None for no border.

Table Width

Choose to change the width of the table produced:

- Automatic – tries to mimic the width of the used area on the first sheet.
- Fixed pixels – locks the width size in to a chosen pixel width.
- Fixed percentage – calculates a percent of the size of the used area on the first sheet.

Overall Commands

About / Register

Clicking 'About / Register' activates a window where you can register KDCalc or update your registration settings. KDCalc must be registered for full functionality. An Internet connection is required for registration.

Help

Clicking 'Help' activates a web browser window pointed at the documentation root.

Cancel

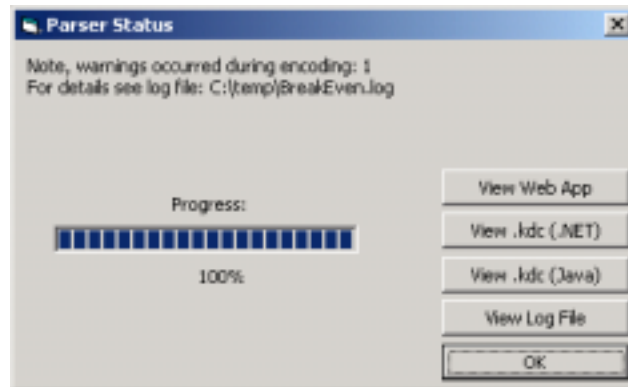
Clicking 'Cancel' will exit the KDCalc Designer window without generating a .KDC module and without saving changes to the compilation settings.

OK

Clicking 'OK' will exit the KDCalc Designer window without generating a .KDC module. But it will save any changes to the compilation settings.

Generate

Clicking '**Generate**' launches the encoding process and saves the spreadsheets settings. The process can take from a few seconds to a few minutes depending on the size and complexity of the spreadsheet being processed.



When the encoding process is complete a status dialog will notify the user if any errors or warnings were encountered during the generation process. Click "**View Log File**" to see these errors. Some errors are fatal and will prevent the .kdc file from working within the engine.

If the option to *Generate Display File* was checked, then the '**View Web Page**' button will be enabled. Clicking this button will launch a web browser and display the generated .HTML page.

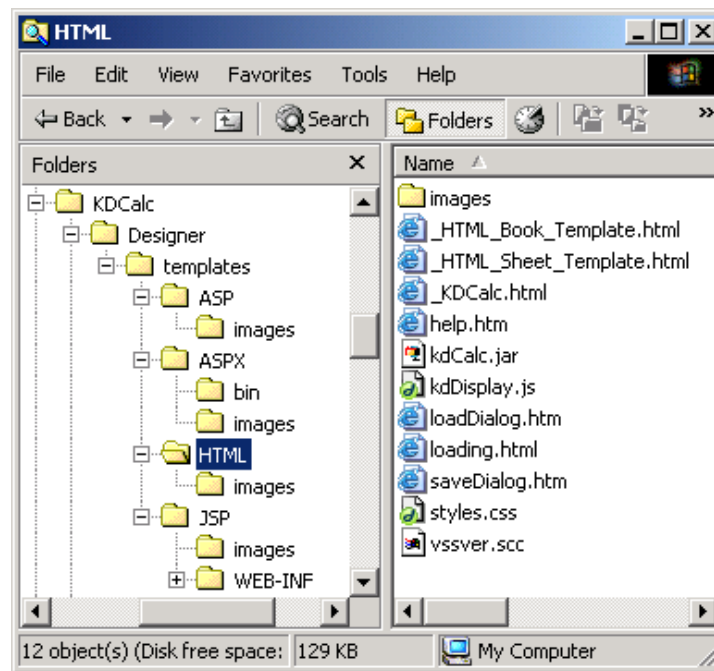
Click '**View .kdc (.NET)**' or '**View .kdc (Java)**' in order to view the .kdc file produced with the KDSpreadSheet Viewer in .NET or Java respectively.

Custom Templates

KDCalc generates web applications based on files stored in a template folder. A separate template folder exists for each type of web application that KDCalc can generate. These folders contain graphics, javascript, and JSP, ASP.NET, or HTML pages. You can choose to use one of KDCalc's default templates, or you can create custom templates with your own graphics, buttons, links, and other features.

You can choose to use one of KDCalc's default templates, or you can create a custom template with your own graphics, buttons, links, scripts, and other features. The figure below shows where the templates are located within the KDCalc directory and the files included. In creating a custom template it is advised to copy and rename an existing template folder and then make modifications such as changing the web page layout and adding graphics.

When you generate your web app, KDCalc copies all files and folders that do not start with "_" to the output directory. It then reads in all special files starting with "_" and replaces the special tags within them with generated code. KDCalc depends on these tags to know where to insert code for the generated web application. The code generated for each tag is different across HTML, JSP, or ASP.NET.



The table below lists the special tags and their meanings:

Tag	Description
<!--KD_CALC_FILENAME -->	Replaces the tag with the name of the Calculation File (.kdc file)
<!--KD_BOOK_NAME-->	Replaces the tag with the title of the Workbook. If original the workbook was named <i>Investments.xls</i> , then the title is <i>Investments</i> .
<!--KD_SHEET_NAME-->	Returns the current sheet name
<!--KD_SHEET_FILE_NAME-->	Returns the current sheet file name (ex: <i>Sheet1_Sheet.html</i>)
<!--KD_ZOOM_VAL-->	The zoom value for the current sheet.

<!--KD_GEN_STYLES-->	Replaces the tag with a list of all the styles used within the sheet.
<!--KD_GEN_TABS -->	Generate the sheet tabs for the sheet
<!--KD_GEN_DEFAULT -->	Replaces the tag with the cell code for the current sheet.

NOTE: The comment tags (<!--, -->) are part of the template tags and there are no spaces.

Example custom template:

The easiest way to make your own custom templates is to copy and edit one of the default templates. Open the templates directory under where you installed KDCalc (for example: *C:\Program Files\Knowledge Dynamics\KDCalc\Parser\Templates\HTML*)

Copy and paste the HTML folder somewhere and rename it (for example: *c:/temp/CustomHTMLTemplate*)

You can then change the files in this template to work the way you want. All of the files starting with "_" can be modified but not deleted. Open up the *_HTML_Sheet_Template.html* and add your company's logos and branding. You can create or modify the toolbar across the top as well. If you only want to show one sheet, you can remove the sheet tabs. You can even add your own javascript functions.

Color Schemes

In KDCalc, color schemes allow you to exercise fine-grained control over certain cell behaviors in the compilation process. This helps you optimize the size and performance of your compiled files.

KDCalc uses one set of color schemes to forcibly *exclude* cells from compilation and one set of color schemes to forcibly *include* cells in compilation. KDCalc also uses color schemes to specify cell value persistence characteristics for loading and saving cell values at runtime.

The color scheme of a cell is determined by the *font color* and the *background color* of the cell. Two cells have the same color scheme if they have exactly the same font color and background color.

KDCalc lets you specify color schemes using two reserved names: 'zCInclude', 'zCExclude'.

zCInclude

By default, KDCalc compiles all non-empty cells and ignores all empty cells. A quick and easy way to reduce the size of your .KDC file is to check the 'Omit Unreferenced Data Cells' option on the KDCalc Designer screen. The compilation process will ignore all cells that contain text or numeric values that are not referenced by other cells. This will eliminate 'label' cells that are needed during development of the spreadsheet, but not at runtime.

Your spreadsheet may have cells that are unreferenced that you still want to include in compilation. This way the calling application will have access to the data in those cells at runtime. The zCInclude color scheme is used for this purpose. zCInclude allows you to forcibly include non-formula cells that would otherwise be excluded from compilation.

The zCInclude color scheme is only useful when the 'Omit Unreferenced Data Cells' option is checked. It has no effect when the option is not checked.

The zCInclude color scheme is best used when you want to include data in your spreadsheet that is not used by any cell formula, but is used by the calling application.

Cells that are outside the parse area for the worksheet will be ignored, even if they have zCInclude color scheme.

See below for instructions on defining color schemes.

zCExclude

zCExclude allows you to forcibly exclude cells that would otherwise be included in compilation. Just as label cells are important during spreadsheet development, so too, are 'instrumentation cells'. These are cells that provide metrics about other cells in the spreadsheet that a designer uses to gage the behavior of the spreadsheet during design. If they are not needed at runtime they can be excluded using zCExclude.

zCExclude can also be used to exclude label cells when the 'Omit Unreferenced Data Cells' option is not checked.

The optimal combination of 'Omit Unreferenced Data Cells', zCExclude, and zCInclude will be different for different spreadsheets.

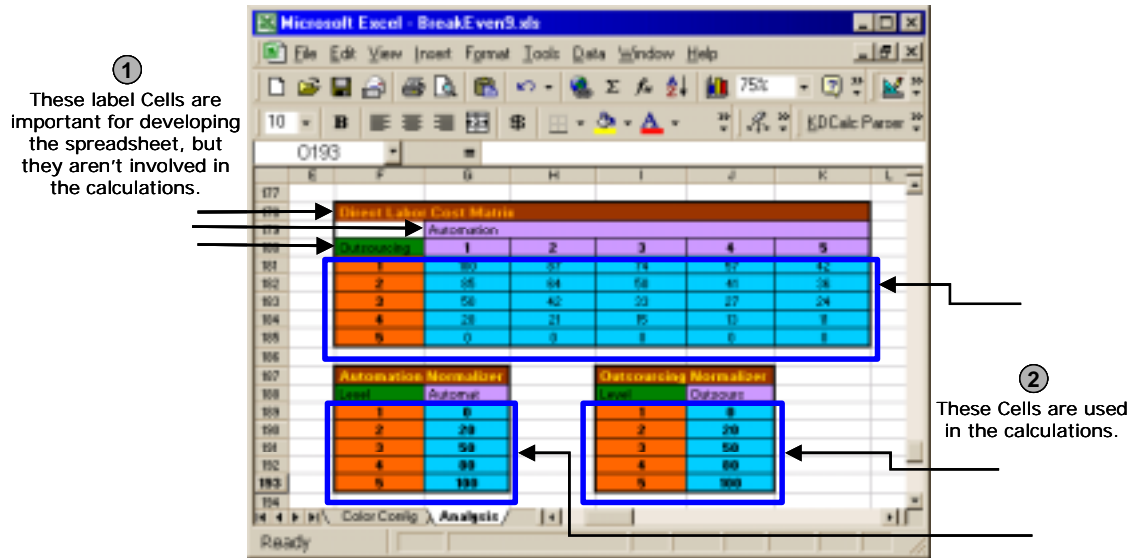
If a zCExcluded cell is referenced at runtime, the cell behaves as if it was empty in the original spreadsheet. It will be accessible, but it will remain empty unless you put data in it programmatically.

See below for instructions on defining color schemes.

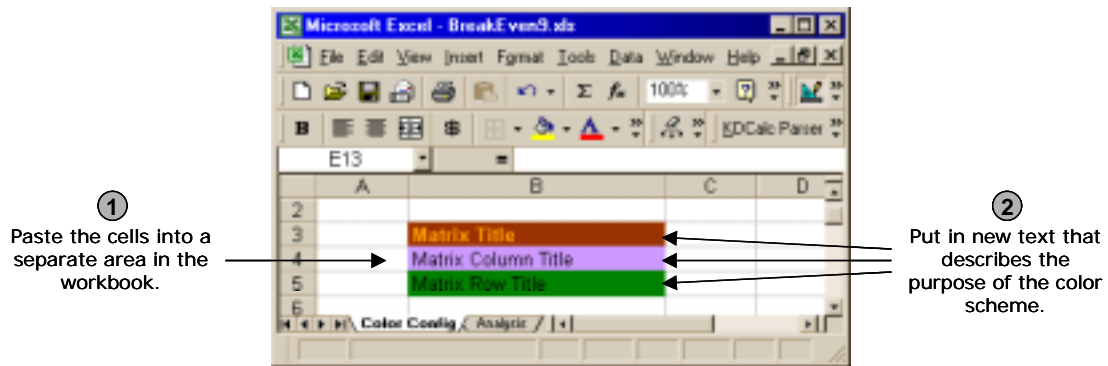
Defining Color Schemes

This example shows you how to define a set of color schemes. zCExclude is used in the example, but the same process is used to define color schemes for zCInclude.

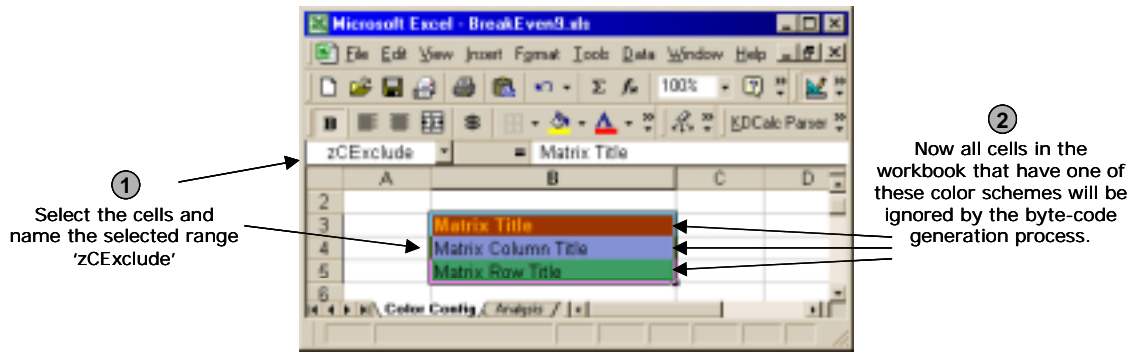
During development of a spreadsheet it is important to have visual cues so the spreadsheet is easier to edit. The 'label' cells in the worksheet below are important during spreadsheet development but are not used at runtime. They only fatten up the .KDC file. The label cells can be omitted from compilation by flagging them as zCExclude.



To mark the label cells as zCExclude, first copy one of the cells and paste it in a separate area. Do the same for any other color schemes that are not needed at runtime. This is illustrated below.



Next select the cells as illustrated below and name the range 'zCExclude' by typing 'zCExclude' into the Name Field and pressing 'Enter'.



Now any cells that have any of the specified color schemes will be ignored during compilation. The range named zCExclude can have as many color schemes in it as you want and it is completely configurable by you.

It's a good practice to settle on a standard set of color schemes for your development team. That way communication between developers and designers is easier and you can start each spreadsheet project from a common template with all the color schemes pre-defined.

Feature Support Notes

KDCalc v4.2.x Supports These Excel Features

- Over 200 of Excel's most commonly used functions (See functions reference for a full list)
- Nested parentheses
- Off-Sheet references
- Named Cells, Ranges, and Formulas
- Whole-column ranges or whole-row ranges like “=SUM(J:J)” or “=SUM(2:2)”

KDCalc v4.2.x Doesn't Support These Excel Features

- Off-book references (cached values are substituted and used)
- User defined functions (partially supported, contact us for details)
- Updating cell formulas at runtime (cell formulas can be overwritten with values at runtime)
- Reading cell formulas at runtime
- Non-rectangular named ranges (named sets of individual cells)
- Multi-sheet ranges like “=SUM(Sheet1:Sheet3!F5:G7)”
- Intersection syntax like “=SUM(H14:I18 I15:J18 I12:K15)” (no commas)
- Arrays
- Ctrl-Shift-Enter style array references
- Circular references

Function Reference

(See attached file: *KDCalcDi\docs\KDCalc Functions.pdf*)

Error Reference

In general, a Warning means that the calculation (.kdc) file or web application was successfully created, but might have errors or omissions in it. When an Error occurs, the calculation file or web application will not be successfully created. The error must be remedied to continue.

Compilation warnings:

Warning #	Warning	Notes
100	Unsupported Excel Function	A cell formula uses a known Excel function that is not supported by KDCalc.
110	Unknown Function	A cell formula uses an unknown Function.
120	Unknown Symbol	A cell formula uses an unknown Symbol. The most common cause of this error is a cell formula that refers to a cell or a name that is outside of the parse area.
130	Syntax Error	The cell formula cannot be parsed. The most common cause of this error is off-book references, which are not supported by KDCalc.
140	Arrays Not Supported	Ctrl-Shift-Enter style array references are not supported.
160	Circular Reference Option	Excel has the 'Iteration' option on. KDCalc does not support circular references. If the workbook has circular references it may encode, but may not execute properly. Uncheck the 'Tools->Options->Calculation->Iteration' option in the Excel toolbar and remove any circular references from the workbook.
170	Off-book Reference	KDCalc does not support off-book references. You may move the off-book references to another sheet in the same book or manually move the data between the books.

Compilation errors:

Error #	Error	Notes
1000	Compression Error	An error occurred with the compression library used by KDCalc.
1010	Output Directory Does Not Exist	The output directory for KDCalc to write to does not exist and KDCalc was unable to create it for unknown circumstances. Create the directory manually or use a different directory.
1020	Bad Path or Missing Template File	The path to the template file for creating a web application is incorrect or the file is missing.
Unlisted	Unknown / Unexpected Error	Please report the error to Knowledge Dynamics by sending the log file and .xls file which caused the error.

Best Practices

Standardize on a set of known color schemes.

See color schemes topic above.

Use Instrumentation sheets for development and testing that are excluded at runtime.

While developing a spreadsheet, it is often helpful to have cells that calculate metrics about the main cells. Values like Sums, Averages, and other calculations can help you do a 'reality check' on the core functionality you are trying to achieve. If these calculations are not needed at runtime, they can be excluded by KDCalc. The easiest way to do this is by keeping these cells on separate sheets and excluding the sheets at runtime.

Use zCExclude on user input cells to enable automated testing

Let's say cells C5:C20 on MainSheet are user input cells and the rest of your spreadsheet depends on them. You might test your spreadsheet by entering different values into the user input cells and checking the outcomes.

But this method of testing gets tedious, so you make multiple sets of test data and put it in a separate TestDataSheet worksheet. D5:D20, E5:E20, F5:F20 etc. on TestDataSheet all contain distinct sets of test data. Now you enter a formula in MainSheet!C5 so that it formulaically gets its value from one of the sets of test data by using OFFSET function. Likewise for C6:C20. You have a single cell that drives which set of test data to use and cells C5:C20 OFFSET by that amount.

Now you can overwrite the input cell formulas at runtime with user input data. You could also mark the user input cells as zCExclude. Then at runtime they will behave as if they were empty in the original spreadsheet and the compiled .KDC file size will smaller.

Frequently Asked Questions

I installed KDCalc but it's not showing up in my Excel toolbar. What should I do?

Look for 'KD Calc Designer' in the 'Tools' menu. If it's not there try reinstalling KDCalc. If it is there you can make it show up on the toolbar as well. Right-click on the Excel toolbar and click 'Customize'. Then click the 'Tools' menu and drag the 'KDCalc Designer' menu item to the Excel tool bar. If you hold down the 'ctrl' key when you release the mouse button, you will have access to KDCalc in both the 'Tools' menu and the Excel toolbar.

KDCalc was working fine. Without warning it started giving me warnings saying that the Product was not registered. What happened?

The files containing your License and Registration information may have been damaged or deleted. You can simply re-register with the same License Code and KDCalc will return to full functionality.

End User License Agreement

(See attached)