<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACsc( )</td>
<td>111</td>
</tr>
<tr>
<td>ACsch( )</td>
<td>111</td>
</tr>
<tr>
<td>Asc( )</td>
<td>112</td>
</tr>
<tr>
<td>ASec( )</td>
<td>112</td>
</tr>
<tr>
<td>ASecH( )</td>
<td>113</td>
</tr>
<tr>
<td>ASin( )</td>
<td>113</td>
</tr>
<tr>
<td>ASinh( )</td>
<td>114</td>
</tr>
<tr>
<td>ATan( )</td>
<td>114</td>
</tr>
<tr>
<td>ATanh( )</td>
<td>114</td>
</tr>
<tr>
<td>Binomial( )</td>
<td>115</td>
</tr>
<tr>
<td>Choose( )</td>
<td>115</td>
</tr>
<tr>
<td>Ceiling( )</td>
<td>116</td>
</tr>
<tr>
<td>Chr( )</td>
<td>116</td>
</tr>
<tr>
<td>Combin( )</td>
<td>117</td>
</tr>
<tr>
<td>Cos( )</td>
<td>117</td>
</tr>
<tr>
<td>CosH( )</td>
<td>118</td>
</tr>
<tr>
<td>Cot( )</td>
<td>118</td>
</tr>
<tr>
<td>CotH( )</td>
<td>118</td>
</tr>
<tr>
<td>Csc( )</td>
<td>119</td>
</tr>
<tr>
<td>Csch( )</td>
<td>119</td>
</tr>
<tr>
<td>Cumipmt(_)</td>
<td>120</td>
</tr>
<tr>
<td>Cumprinc(_)</td>
<td>120</td>
</tr>
<tr>
<td>Deg( )</td>
<td>121</td>
</tr>
<tr>
<td>erf( )</td>
<td>121</td>
</tr>
<tr>
<td>Exp( )</td>
<td>121</td>
</tr>
<tr>
<td>Exponent( )</td>
<td>122</td>
</tr>
<tr>
<td>Floor( )</td>
<td>122</td>
</tr>
<tr>
<td>Fv( )</td>
<td>123</td>
</tr>
<tr>
<td>gcf( )</td>
<td>123</td>
</tr>
<tr>
<td>If( )</td>
<td>124</td>
</tr>
<tr>
<td>Instr( )</td>
<td>124</td>
</tr>
<tr>
<td>Int( )</td>
<td>125</td>
</tr>
<tr>
<td>Inv( )</td>
<td>125</td>
</tr>
<tr>
<td>Ipmt(_)</td>
<td>125</td>
</tr>
<tr>
<td>LCase( )</td>
<td>126</td>
</tr>
<tr>
<td>Left( )</td>
<td>126</td>
</tr>
<tr>
<td>Len( )</td>
<td>127</td>
</tr>
<tr>
<td>Log( )</td>
<td>127</td>
</tr>
<tr>
<td>Log10( )</td>
<td>128</td>
</tr>
<tr>
<td>LogB( ), LogN()</td>
<td>128</td>
</tr>
<tr>
<td>LTrim( )</td>
<td>128</td>
</tr>
<tr>
<td>Mantissa( )</td>
<td>129</td>
</tr>
<tr>
<td>Max( )</td>
<td>129</td>
</tr>
<tr>
<td>Mid( )</td>
<td>130</td>
</tr>
<tr>
<td>Min( )</td>
<td>130</td>
</tr>
<tr>
<td>Nper(_)</td>
<td>130</td>
</tr>
</tbody>
</table>
Npv(_) .................................................................................................................. 131
Perm( ) .................................................................................................................. 131
Pmt(_) .................................................................................................................. 132
Ppmt(_) .............................................................................................................. 132
Prime( ) ............................................................................................................. 132
Pv() ...................................................................................................................... 133
Rad() .................................................................................................................... 133
Rand() ............................................................................................................... 134
Rate(_) ................................................................................................................. 134
Reduce() ............................................................................................................. 135
Right() ............................................................................................................... 135
Round() .............................................................................................................. 135
RoundSig() ....................................................................................................... 136
RTrim() ............................................................................................................... 136
Sec() .................................................................................................................... 137
SecH() ............................................................................................................... 137
Sgn() .................................................................................................................... 138
SigUnit() ............................................................................................................ 138
Sin() ..................................................................................................................... 138
SinH() ............................................................................................................... 138
Sqr() ..................................................................................................................... 139
Str() ..................................................................................................................... 140
StudentSt() ...................................................................................................... 141
Sum() .................................................................................................................... 141
Tan() ..................................................................................................................... 141
TanH() ............................................................................................................... 142
Trim() .................................................................................................................. 142
UCase() .............................................................................................................. 143
Unique() ............................................................................................................. 143
Val() ..................................................................................................................... 143
AND Operator .................................................................................................... 145
Concatenation Operator .................................................................................... 146
EQV Operator .................................................................................................. 146
IMP Operator .................................................................................................. 146
NOT Operator .................................................................................................. 147
OR Operator ..................................................................................................... 147
XOR Operator .................................................................................................. 147

Additional Resources ........................................................................................ 148
TheTestingCenter.Com ....................................................................................... 148
Software Updates .............................................................................................. 149
Technical Support ............................................................................................. 149

Product Feedback ............................................................................................. 150

Disclaimer .......................................................................................................... 151
Welcome to Wimba Diploma 6, the easy-to-use test-generation program packed with features that make evaluating and learning easier, more effective and more fun. With Wimba Diploma 6, you can quickly create tests, write and edit questions, and create study-sessions for students. Tests can be printed in a wide range of formats or administered to students online through TheTestingCenter.

This guide is separated into three main parts. The first is the Quick Tour, which takes you through the basics of creating questions and exams, saving, editing, and printing. This is followed by the In-Depth Help, with a more detailed look at Wimba Diploma 6, as well as explanations of its more advanced features. Finally, the Additional Resources section includes helpful links and information for your reference.
What's New in Wimba Diploma 6

The most significant enhancements for the most recent releases of Wimba Diploma are listed below.

New in Wimba Diploma 6.64

- Exporting for Moodle 1.7 - 1.9
- Diploma is now called Wimba Diploma

New in Diploma 6.61:

- Improved Wimba Plot functionality:
  - The ability to add a grid to plots
  - The ability to zoom and pan the axes from the plot preview
  - A drop-down menu within plot item parameters to provide easy access to functions
  - Expanded functions and operator syntax for parameterized plot items
  - Improved Wimba Plot window controls
  - Improved plot item rendering
- Added new Polynomial plot item functionality:
  - The ability to label individual points
  - The ability to define a Polynomial by critical points
  - The ability to define a Polynomial by zeros of second derivative
- Added a Number Line Coordinate type with the following plot items:
  - Point
  - Segment
  - Ray
  - Line
  - Box-and-whisker Plot
  - Vector

New in Diploma 6.60:

- Wimba Plot math plotting functionality with support for both Cartesian and Polar coordinate systems and the following plot types:
  - $y = f(x)$
  - Parabola
  - Circle
  - Ellipse
  - Hyperbola
  - Point
  - Segment
  - Polynomial
  - Parametric
  - Normal Curve
• Text Box
• \( r = f(\Theta) \)
• Point at \((r, \Theta)\)
• The ability to control plots using variables

**New in Diploma 6.5:**
• Support for Windows Vista and Internet Explorer 7.
• Spell check.
• Updated menu and toolbar styles.
• "Show Me How To" guides.
• Improved support for Unicode text.

**New in Diploma 6.4:**
• Exporting to WebCT Vista 3, WebCT Vista 4, WebCT CE6, and Angel 6.2+.
• Streamlined exporting process.
• Hyperlink support.
• New "Multiple Selection" question type.
• Fill-in-the-Blank and Short Answer support multiple answers.
• Matching questions can have additional distracters defined.
• New question editing features.
• New licensing system.

**New in Diploma 6.3:**
• New print engine to improve performance and reliability of printing and RTF exporting.
• Character Palettes for inserting special characters.
• Image List Variable for dynamically inserting images into question text.

**New in Diploma 6.2:**
• On-screen getting started guide.
• Preview pane for Question Bank and Exam windows.
• Integrated RTF Import.
• Dynamic equations export as a static image for all HTML exports.
• Manage default print settings.
• Support for standard instructions.

**New in Diploma 6.1:**
• Improved Windows XP compatibility.
• Advanced Find tool.
• Support for more graphics formats -- you can insert BMP, DIB, RLE, DXF, EPS, FPX, GIF, IMG, JPG, PCD, PCX, DCX, PNG, TGA, TIF, and WPG images into Diploma.
• Context-sensitive (right-click) menus.
• Better integration with MathType 5.
• Export to EDU, WebCT, and Blackboard.
• Exporting equations to MathML.
• JPG images are stored in their native format (no longer internally converted to PNG).
- Built-in check for software updates.
- Improved documentation.
- Enhanced performance.
Quick Tour

Creating an Exam

1. Open the Question Bank from which you wish to copy questions.
2. Click the File menu.

![New Exam Window]

4. Click the Save button. The Save As window opens.
5. Type the name you wish to use for the exam in the File name box.
6. Click Save.

**Note:** By default, the standard instruction is added to every new exam.

**See Also:**
- Standard Instructions

Writing Questions

You can add new questions to a Question Bank that can be copied to an exam later, or you can add new questions directly into the exam itself.

1. From either the Question Bank window or an existing exam, click the Question menu.
2. Select New Question. The New Question window opens.
3. Choose the settings for the new question by clicking the radio buttons in each section:
   a. Question Format lets you choose the type of question that you wish to add.
   b. Location lets you choose where the new question is inserted in relation to existing questions.
   c. Quantity lets you choose how many questions you wish to add.

4. Click **OK**. A new question titled "(Blank question -- double-click to edit)" is copied to the question list.

   ![Image of a question with instructions]

   **Exam1**

   ![Image of a question editor]

5. Double-click the new question. The question opens in the Question Editor window.

6. Edit the text of the question and its response(s).

7. Close the Question Editor window.
Editing Questions

1. Double-click the question that you wish to edit in the Question Bank or Exam window. The question opens in the Question Editor window.
2. Make your changes to the question.
3. Close the Question Editor window.
4. Click the File menu in the Question Bank window.
5. Select Save.

Tip: You can use the View drop-down menu, located in the top-right corner of the Question Editor, to access advanced options for question editing; this includes Feedback, Hints, Information Fields, Algorithms, and more.

Previewing a Question

1. Click the chapter that contains the question that you wish to view in the Chapter column.
2. Click the question that you wish to preview.
   a. If the Preview pane is enabled, the selected question appears below the question list.
b. If the Preview pane is disabled, click the **Preview** button to enable it.

### Copying Questions to an Exam

**Copying a Specific Question to an Exam**

1. Open the exam to which you wish to copy the question.
2. Click the desired chapter from the Chapter column.
3. Click the question that you wish to copy.
4. Click the **Copy To** button. The selected question is copied to the exam.

**Note:** The name of the **Copy To** button changes depending on whether you have an exam currently open. If an exam is open, the name of the exam appears after Copy To. If an exam is not open, the button says: **Copy To A New Exam**; clicking it creates a new exam, and copies the selected question to it.

**Copying a Series of Consecutive Questions to an Exam**

1. Open the exam to which you wish to copy the question.
2. Click the desired chapter from the Chapter column.
3. Click the first question in the series that you want to copy.
4. Hold the Shift key and click the last question in the series.

5. Click the **Copy To** button. The selected questions are copied to the exam.

**Note:** The name of the **Copy To** button changes depending on whether you have an exam currently open. If an exam is not open, the button says: **Copy To A New Exam**; clicking it creates a new exam, and copies the selected question to it.

**Copying a Group of Non-Consecutive Questions to an Exam**

1. Open the exam to which you wish to copy the question.
2. Click the desired chapter from the Chapter column.
3. Click the first question that you want to copy.
4. Hold the Ctrl key and click another question that you want to copy. Repeat until all the questions that you want to copy are selected.
5. Click the **Copy To** button. The selected questions are copied to the exam.

---

**Note:** The name of the **Copy To** button changes depending on whether you have an exam currently open. If an exam is not open, the button says: **Copy To A New Exam**; clicking it creates a new exam, and copies the selected question to it.

### Copying Questions Randomly to an Exam

1. Open the exam to which you wish to copy the question.
2. Click the desired chapter from the Chapter column.
3. Click the **Copy Randomly** toolbar button. The Randomly Copy Toolbar opens.
4. Type the number of questions you wish to randomly copy in the Randomly Copy box.
5. Click the **Copy Now** button. The questions are copied to the exam.
Note: If more than one question in a chapter is highlighted, the Copy Randomly feature limits its choices to the highlighted questions. You can take advantage of this feature to limit the range of questions that the program randomly chooses.

Saving an Exam

To save changes to an exam, click the Save button on the Exam window.

a. If you have not already saved the exam, a dialog appears prompting you for a name and location to save the file.

b. If you have already saved the exam, but you want to save your changes to a different file, click the File menu and select Save As.
Printing an Exam

1. Click the File menu in the Exam window.
2. Select Print. The Print window opens.
3. Adjust the print settings to meet your needs.
4. Click OK. The exam prints.
Scrambling Options

When printing an exam, there are two options to scramble the output order of your questions. Both of these options are accessible from the Print window.

- Check the **Scramble questions** box to shuffle the output order of the questions
- Click the **Scramble multiple-choice answers** box to shuffle the order of the choices for multiple-choice questions only.

Algorithm-Based Questions

Diploma 6 includes advanced features that allow questions to be controlled by algorithms, allowing a single question to act as a template for potentially thousands of variations.

The **Solve Variables** button is visible whenever a question that is displayed uses algorithms. Click this button to generate a variation of the question and display a student preview.
Note: When you click the **Solve Variables** button, the values are not substituted permanently. To see other values, click the **Solve Variables** button again. To see the original variables, click anywhere within the text of the question.

See Also:
Overview: Algorithms
Adding Variables
To Questions
Defining Variables
Using Variables In Equations
In-Depth Help

The Wimba Diploma 6 Interface

Open File Window
The Open File window is designed to make it easier to manage your question banks and exam documents. By default, when you start Diploma 6, you see the Open File window.

To Launch the Open File Window
1. Click the File menu in either the Question Bank or Exam window.
2. Select Open. The Open File window opens.

The Question Banks tab contains a list of all of the question banks you have installed. Anytime you open or save a Question Bank in Diploma 6, it is automatically added to this list for easy access in the future.

The Exams tab is similar to the Question Banks tab except that it contains a list of the exam files that you have created or opened in Diploma 6. Also, while Question Banks are sorted alphabetically, exam documents are sorted by the order in which the Exam was last opened, with the most recently opened file at the top of the list.

To open a file from the Open File window, either double-click the name of the file or select the file and click Open.

When opening a Question Bank, if the Start with a new exam box is checked, a new exam is created when the Question Bank is opened. Click the Browse... button to access a Question Bank or exam document that has not yet been added to the Open File window.
Tip: You can remove files from the Open File window list by right-clicking the name of an exam or Question Bank and selecting Remove from the pop-up menu. You are then prompted to choose whether you want to just remove the link to the file in the Open File window or delete the file from your computer altogether.

Exam Document Window

The Exam Document window is where you assemble, view, print, and save exams. You can access the Exam Document window either by opening an existing exam, or by creating a new exam. By default, when a new exam document is created, the Standard Instruction is placed at the beginning of the file. You can edit this Standard Instruction by double-clicking it.

See Also:
Standard Instructions

Question Bank Window

The Question Bank window is similar to the Exam Document window, except that it contains all of the questions that you have added to the open Question Bank instead of just those that you have copied to an exam. The Question Bank also lets you organize your questions into different chapters. You can view all of the questions in a particular chapter by clicking its name in the list on the left side of the window.
To preview a question from within the Question Bank window, select the question from the list and click the Preview button.

**Toolbars**

The Question Bank and Exam Document windows share a common set of toolbars which can each be enabled or disabled individually.

**To Enable Or Disable a Toolbar**

1. Click the View menu.
2. Select **Toolbars**.

   ![Toolbars](image)

   - **Preview**
   - **Shortcut**
   - **Exam Creation**
   - **Copy Randomly**
   - **Filter Questions**

3. Select the name of the toolbar that you wish to enable or disable.

**Shortcut Toolbar**

![Shortcut Toolbar](image)

The shortcut toolbar contains buttons for saving, printing, and exporting your documents.

**Exam Creation Toolbar**

![Exam Creation Toolbar](image)

The exam creation toolbar contains buttons used for building exams. The Copy To button allows you to copy selected questions into an open exam document. The Copy Randomly button opens the Copy Randomly toolbar and the Filter Questions toolbar opens the Filter Questions toolbar (see below).

**Note:** If more than one exam document is open, an additional drop-down button appears next to the Copy To button. This second button allows you to select which exam questions are copied to when you click the Copy to button.

**Copy Randomly Toolbar**

![Copy Randomly Toolbar](image)

The Copy Randomly toolbar allows you to include a sampling of a specific number of questions from the current selection into an exam. If you have only one question selected, the questions that are copied to the exam are randomly selected from all of questions in the current chapter.

**To Randomly Copy Questions to an Exam**

1. Open the exam of your choice.
2. Select the questions from the list that you wish to include in the random selection.
3. Enter the number of questions that you wish to randomly copy in the Randomly Copy box.
4. Click **Copy Now**. The questions are randomly copied to the open exam.

**Filter Questions Toolbar**

![Filter Questions Toolbar](image)

To make it easy to find the questions of your choice, Diploma 6 includes a filtering feature. Using the Filter Questions toolbar, you can filter the question list to include only a particular question format, or questions that contain a particular information field value. This toolbar is made up of a series of drop-down menus. The Filter By drop-down lets you select the other drop-down menus that appear in the toolbar. The Reset All Filters option lets you quickly disable all of the current filters; this can also be accomplished by disabling the Filter Question toolbar.

**Tip:** You can combine the question filter feature with the random selection feature to randomly copy from only a certain type of question. For example, you could set the question filter to display only Multiple Choice questions with a Difficulty value of Hard, then randomly copy questions to an exam only from that group.

If you need even more powerful search capabilities, you can use the Advanced Find tool to run detailed queries on an entire Question Bank.

**Columns**

The Columns feature allows you to show or hide information about the questions in the open Question Bank. When you enable a column, the information displays for the questions in the open chapter, making it easier for you to work with and sort through your questions.
To Enable Or Disable a Column

1. Click the View menu.
2. Select Columns.
3. Select the name of the column that you wish to enable or disable.

Standard Instructions

The default question that is added to new exam when it is first created is called a Standard Instruction. You can customize the Standard Instruction to make it easier to adapt it into the type of questions that you wish to create.

To Customize the Standard Instruction

1. Double-click the current Standard Instruction in the Exam Document window to open it in the Question Editor window.
2. Modify the question so that it appears as you would like your new Standard Instruction to appear.
3. Click the **Instruction Defaults** drop-down menu.
4. Select **Save as standard instruction**. The Standard Instruction is modified; the new Standard Instruction is automatically added the next time you create an exam.

**Tip:** If you do not want the Standard Instruction to automatically be added to new exams, you can disable this feature by un-checking the **Add standard instruction to new exams** box in the Exam Creation section of the Options window.

**Advanced Find**

Advanced Find is a powerful search tool that lets you find questions based on complex queries.

**To Access Advanced Find**

1. Click the **Tools** menu.
2. Select **Advanced Find**. The Advanced Find window opens.

Using Advanced Find, you can set specific criteria for searching across an entire Question Bank.
• The Text tab allows you to search for questions, references, and instructions that contain a specific phrase, or that contain tables, graphics, equations, or that use variables.

• The Information Fields tab allows you to search for questions with specific combinations of field values.

  Only include questions with the field values indicated below. Questions must match at least one of the values in each row.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difficulty</td>
<td>Easy</td>
<td>Moderate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Application</td>
<td>Knowledge</td>
<td>Synthesis</td>
<td></td>
</tr>
<tr>
<td>(Select Field)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  In the example above, a question must have a Difficulty value of Easy or Moderate, as well as have a Type value of Application, Knowledge, or Synthesis in order to be returned in the search. Any number of fields and any number of values may be added for your search.

• The More Choices tab allows you to specify the question formats that you would like to return in your search.

Once you complete your search, double-click a question in the results list to view or edit it. You can copy a question to an exam by using the Tools drop-down menu or by dragging it from the Advanced Find window to the Exam document window.

**Options Window**

The Options window contains global settings for the program, separated in to six sections: General, Exam Creation, Visual Style, Languages, Connection Settings and Export Profiles.

**To Access the Options Window**

1. Click the **Tools** menu.
2. Select **Options**. The Options window opens.
General

- **Disable undo functionality**
  Allows you to conserve system resources and improve speed by disabling the undo function.

- **Don't use non-breaking spaces for layout when printing**
  Normally, the answer key includes non-breaking spaces to prevent word-wrapping in inappropriate places. On some foreign-language systems, non-breaking spaces do not appear correctly. Check this box to use regular spaces instead.

- **Use tabs for layout when exporting RTF**
  Versions of Microsoft Word prior to Word 2000 do not support nested tables. By default, Diploma 6 uses tables with multiple columns to format questions. Checking this box allows you to use tabs instead of tables. The primary disadvantage to using tabbed layout is that tables inside question text cannot be displayed on the same line as the question number or answer designators. This option does not affect printed documents.

- **Show editing marks**
  Check this box to display hard paragraph breaks such as paragraph breaks and spaces in your question text.

- **Show welcome message on startup**
  Allows you to enable or disable the welcome balloon during startup.
- **Restore all caution messages**  
  Allows you to reset all "Don't ask me this again" dialogs

**Exam Creation**

- **Permit question duplication in exams**  
  Allows you to choose whether the same question can be added to an exam twice

- **Start with a new exam**  
  Allows you to choose whether a new exam file is automatically created when you open a Question Bank

- **Add standard instruction to new exams**  
  Allows you to choose whether new exam files start with the Standard Instruction

**Visual Style**

Allows you to customize the look and feel of Diploma 6’s menus and toolbars

**Languages**

Allows you to choose the language for the "Show Me How To" feature

**Connection Settings**

Allows you to set up your proxy server for Diploma 6’s connection to the Internet, as well control how automatic updates are handled

**Export Profiles**

Allows you to set up your connection information for TheTestingCenter.Com

**Question Editing**

**Question Editor Window**

Although you can preview a question from the Question Bank or Exam document window, to make changes to the question, you need to open it in the Question Editor window. You can do this by double-clicking the name of the question in either an exam or a Question Bank.
The main toolbar contains three items:

- The arrow buttons allow you to cycle through the questions in the current chapter or exam.
- The Copy To button allows you to copy the current question to an open exam.
- The View drop-down menu in the top-right corner of the window allows you to enable or disable different toolbars and editor tabs.

- The Menu Toolbar contains buttons relating to adding and editing content within the question itself.
- The Format Toolbar contains basic text-formatting commands.
- The Character Palettes menu provides access to mathematical and foreign-language symbols.
- The five selections below the Character Palettes menu provide access to tabs for editing specific elements of a question.
Tip: You can also right-click the status bar below the question content area to reveal a pop-up menu that allows you to show/hide additional information about the current question.

See Also:
- Question Editor Toolbars
- Special Character Palettes
- Advanced Editing

Question Editor Toolbars
There are two toolbars in the Question Editor window that can be enabled or disabled using the View drop-down menu.

Menu Toolbar

- The Edit drop-down menu contains commands for undo/redo and working with the clipboard.
- The Insert drop-down menu allows you to insert tables, images, equations, and other OLE objects.
- The Format drop-down menu contains commands for character formatting, paragraph formatting, borders and shading, image and object properties, and table formatting.
- The Answers drop-down menu contains commands for inserting and removing multiple-choice answers.
- The New drop-down menu allows you to insert new questions, instructions, and references.
- The Save button saves all changes made to the question bank or exam document.
- The Undo drop-down menu provides undo/redo functionality and commands for reverting changes to the saved version.

Format Toolbar

The Format toolbar contains common word-processing formatting commands:

- Font
- Font Size
- Bold
- Italics
- Underline
- Subscript
- Superscript
• Left, Center, and Right Orientation
• A Link to the Windows Character Map utility (if installed)
• Spellcheck

Special Character Palettes
Special Character Palettes are floating windows that contain specific character sets for foreign languages, math, and currency symbols.

To Use a Special Character Palette
1. Click the View drop-down menu.
2. Select Character Palettes.
3. Select the Character Palette that you wish to use. The selected Character Palette opens as a floating window.
4. Place your cursor where you wish to insert a special character.
5. Click the desired character in the special Character Palette. The character is inserted into your question.

Advanced Editing
There are five tabs in the Question Editor window that provide access to more advanced editing features. These tabs can be accessed from the View menu, or by clicking the More button.

Properties
The Properties tab lets you choose the question type, associate a reference to the question, and add a description to the question. In addition, you can disable the question from inclusion when the Randomly Copy feature is used. If a question is disabled, it is silently omitted if selected as part of a larger group of questions to be added to an exam using Randomly Copy. If only disabled questions are selected, a warning dialog appears and you are given the option to copy the questions or cancel the operation.
Note:
The warning for disabled questions only appears when using the Randomly Copy feature; disabled questions are always omitted without a warning when adding questions to an exam using copy-and-paste or drag-and-drop.

Information Fields
The Information Fields tab allows you to add virtually limitless details to characterize a question. A single information field can even be assigned multiple values. The Information Fields tab displays both the fields, and their values, as a table. You can edit a Field Value by clicking it, or edit a Field Name by double-clicking it. You can add a new field by double-clicking the New Field box and entering a title for the field. To add a new value to an existing field, simply double-click an empty box in one of the Additional Value columns; type the desired value, or select a value from the drop-down list.

Feedback
The Feedback tab allows you to edit the feedback (also known as responses or rejoinders) that the student receives about the question they have answered. For multiple-choice and true/false questions, feedback can be general, meaning that the same feedback is given for any response.

Note: When writing feedback for multiple-choice questions, it is common to refer to the answer letters of the correct or incorrect answer, which can result in inaccurate text when using multiple-choice answer scrambling. You can avoid this problem by using the special built-in variables _A, _B, _C, _D when referring to a multiple-choice answer letter.

Hints
The Hints tab allows you to give the student additional information that would suggest the correct answer. Multiple hints can be provided, in which case the first hint would generally be subtle with successive hints becoming more and more obvious.

Variables
The Variables tab allows you to access the features of Diploma 6’s algorithm engine, which allows you to write a single question with endless variations.

See Also:
Algorithm-Based Questions
Answer-Letter Variables

Answer Locking
Answer Locking is a feature that allows you to set the positions of the responses for Multiple Choice questions so that they do not change when the rest of the responses are scrambled. Diploma 6 automatically handles many common answer locking scenarios, but also makes it easy for you to maintain control over how your responses are ordered.
The answer locking status of a response is indicated by the padlock icon next to its radio button. An open padlock indicates that the position of the response is not locked and will be scrambled when the exam is exported. A closed padlock indicates that the answer's position is locked and will not be scrambled. To manually lock or unlock an answer, change the status of the answer to the desired state by clicking the padlock icon next to the answer's radio button.

In some cases, Diploma 6 automatically recognizes that a response should be locked. In the example above, response D has automatically been locked because it contains the phrase "None of the above." Other responses that are locked automatically are those that include "All of the Above" and those that contain the word "either," such as "Either A or B." Even if the answer locking status of a question has been set automatically you can still change it by clicking the padlock icon next to the question.

**Tip:** You can lock all the answers in a question at once by clicking the Answers drop-down menu and selecting Lock All Answers from the Answer Locking sub-menu.

**Equations**
Diploma 6 includes an Equation Editor from Design Science. Using the Equation Editor, you can create and edit high-quality equations that look great printed and render for the web using MathML.

**To Create a New Equation**
1. Click the **Insert** button in the Question Editor window.
2. Select **Equation**. A new equation is created and opens in a the Equation Editor window.
To Edit an Existing Equation

1. Double-click the equation in the Question Editor window. The equation opens in the Equation Editor window.
2. Make the desired changes to the equation.
3. Close the equation Editor window. The changes that you made are reflected in the Question Editor window.

Tip: Design Science, the company behind the Equation Editor in Diploma 6, also makes MathType: a more fully-featured version of Equation Editor. If you frequently include equations in your documents, you may wish to consider purchasing MathType for use with Diploma 6. You can use MathType 5 or higher to replace the Equation Editor inside Diploma 6. For more information, visit the Design Science website at http://www.dessci.com/.

See Also:
Using Variables in Equations

Importing and Exporting

Importing

Importing from RTF

The Import RTF feature in Diploma 6 lets you import questions into Diploma using Rich Text Formatting (RTF) documents created in either a word processing application or text editor. One of the advantages of adding questions to a Diploma Question Bank using this method is that you can easily re-purpose questions that you have in other text formats for use in Diploma 6.
General Formatting Guidelines

These guidelines for RTF import formatting apply to all question types:

- All questions must be separated by at least three blank lines.
- Changes to the Question Format must also be separated from other questions by three blank lines.
- "Keywords" are used to denote additional information, such as the correct answer, information fields, feedback, etc. Keywords generally end with a colon.
- Multiple-choice answers (as well as feedback) must appear after the multiple-choice question.
- Additional question information, such as the correct answer and information fields, may appear either before or after the question.

Formatting Multiple Choice Questions

To Format Text for RTF Import as a Multiple Choice Question:

1. Insert three blank lines after the previous question, or at the top of the page if this is your first question.
2. Type the text, "Multiple Choice".
3. Insert three more blank lines.
4. Type "1.", followed on the same line by the text of your question.
5. Type "A)", followed on the same line by the first response to the question.
6. Repeat Step 5 using different letters to add additional responses to the question.
7. On the line after the last response, type "Ans:", followed by the letter of the correct answer to the question.
8. On the same line as in Step 7, type "Difficulty:" followed by the question’s difficulty level.

**Example:**

Multiple Choice

1. Which of the following cities hosted the 1996 Summer Olympic Games?
   A) Atlanta, USA
   B) Athens, Greece
   C) Rome, Italy
   D) Moscow, Russia
   Ans: A Difficulty: Easy

**Keyword Reference:**
- Question Text -- set to "#.
- Multiple Choice Answers -- set to "A)"
- Correct Answer -- set to "Ans:"
- Start a New Chapter -- set to "Chapter:"
- Multiple Choice -- set to "Multiple Choice"
- Look for MC answers up to -- set to "E" or higher

**Things to Note:**
- The "Chapter:" keyword describes the name to give the chapter in the Question Bank.
- The question number (e.g. - "1.") indicates the beginning of the question text.
- Each question is separated by at least three blank lines; and the "Chapter:" and "Multiple Choice" keywords are also separated from other elements by three blank lines.
- If this is not the first question of this type in your text document, change the "1." to the next number in the sequence.
- A question can have up to 26 responses - one for each letter, A-Z. Each answer letter is followed by a closed parenthesis.
- A correct answer must be associated with every question. The "Ans:" keyword followed by a letter designates the correct answer.
Formatting True/False Questions

**To Format Text for RTF Import as a True/False Question:**

1. Insert three blank lines after the previous question, or at the top of the page if this is your first question.
2. Type the text, "True/False".
3. Insert three more blank lines.
4. Type "1.", followed on the same line by the text of your question.
5. On the next line, type "Ans:", followed by the correct answer to the question.
6. On the same line as in step 5, type "Difficulty:" followed by the question's difficulty level.

**Example:**

True/False

1. Atlanta hosted the 1996 Summer Olympic Games.
Ans: True Difficulty: Easy

**Keywords Reference:**
- Question Text -- set to ".#.
- True/False -- set to "True/false"
- Correct Answer -- set to "Ans:"
- Start a New Chapter -- set to "Chapter:

**Things to Note:**
- Each question is separated by at least three blank lines; and the "Chapter:" and "Multiple Choice" keywords are also separated from other elements by three blank lines.
- If this is not the first question of this type in your text document, change the "1." to the next number in the sequence.
- The "True/False" keyword indicates that the questions that follow are true/false.
- The text after "Ans:", that indicates the correct answer to the question, can be either "True", "False", "T", or "F", and is not case-sensitive.
Formatting Short Answer Questions

To Format Text for RTF Import as a True/False Question:

1. Insert three blank lines after the previous question, or at the top of the page if this is your first question.
2. Type the text, "Short Answer".
3. Insert three more blank lines.
4. Type "1.", followed on the same line by the text of your question.
5. On the next line, type "Ans:", followed by the correct answer to the question.
6. On the same line as in step 5, type "Difficulty:" followed by the question's difficulty level.

Example:

Short Answer

1. Which city hosted the 1996 Summer Olympic Games?
Ans: Atlanta Difficulty: Easy

Keywords Reverence:
- Short Answer -- set to "Short Answer"
- Question Text -- set to "#.
- Correct Answer -- set to "Ans:
- Start a New Chapter -- set to "Chapter:

Things to Note:
- Each question is separated by at least three blank lines; and the "Chapter:" and "Multiple Choice" keywords are also separated from other elements by three blank lines.
- If this is not the first question of this type in your text document, change the "1." to the next number in the sequence.
- The "Short Answer" keyword indicates that the questions that follow are short answer questions.
- The entire answer of the short answer question is imported into Diploma 6. The answer is designated by the "Ans:" keyword. An optional blank line can separate the "Ans:" keyword from the answer.
Formatting Essay Questions

To Format Text for RTF Import as an Essay Question:

1. Insert three blank lines after the previous question, or at the top of the page if this is your first question.
2. Type the text, "Essay".
3. Insert three more blank lines.
4. Type "1.", followed on the same line by the text of your question.
5. On the next line, type "Ans:", followed by the correct answer to the question.
6. On the same line as in step 5, type "Difficulty:" followed by the question's difficulty level.

Example:

Essay

1. The 1996 Summer Olympic Games were hosted in Atlanta. Discuss the changes to Atlanta’s infrastructure that occurred as a direct result of the Olympic Games.
   Ans: An articulate and well-written response. Difficulty: Easy

Keywords Reference:
- Essay -- set to "Essay"
- Question Text -- set to "#.
- Correct Answer -- set to "Ans:
- Start a New Chapter -- set to "Chapter:

Things to Note:
- Each question is separated by at least three blank lines; and the "Chapter:" and "Multiple Choice" keywords are also separated from other elements by three blank lines.
- If this is not the first question of this type in your text document, change the "1." to the next number in the sequence.
- The "Essay" keyword indicates that the questions that follow are essay questions.
- The entire answer of the essay question is imported into Diploma 6, including line breaks. Answers to essay questions tend to be lengthy, sometimes even spanning multiple pages.
Formatting Matching Questions

To Format Text for RTF Import as a Matching Question:

1. Insert three blank lines after the previous question, or at the top of the page if this is your first question.
2. Type the text, "Matching".
3. Insert three more blank lines.
4. Type "1.", followed on the same line by the first matching question.
5. On the next line, type "Ans:", followed by the first matching answer.
6. Repeat steps 3-5 to add more matching pairs.

**Example:**

Matching

1. Atlanta, USA
   Ans: 1996 Summer Olympic Games

2. Moscow, Russia
   Ans: 1980 Summer Olympic Games

Keywords Reference:

- Matching -- set to "Matching"
- Question Text -- set to "#.
- Correct Answer -- set to "Ans:
- Start a New Chapter -- set to "Chapter:

Things to Note:

- The "Matching" keyword indicates that the questions that follow are matching questions.
- Each question is separated by at least three blank lines; and the "Chapter:" and "Multiple Choice" keywords are also separated from other elements by three blank lines.
Formatting Fill-in-the-Blank Questions for RTF Import

To Format Text for RTF Import as a Fill-in-the-Blank Question:

1. Insert three blank lines after the previous question, or at the top of the page if this is your first question.
2. Type the text, "Fill-in-the-Blank".
3. Insert three more blank lines.
4. Type "1.", followed on the same line by the text of your question and a series of underscores (_____) 
5. On the next line, type "Ans:", followed by the correct answer to the question.
6. On the same line as in step 5, type "Difficulty:" followed by the question's difficulty level.

Example:

Fill-in-the-Blank

1. Atlanta hosted the Summer Olympic Games in the year _____
   Ans: 1996 Difficulty: Easy

Keywords Reference:
- Fill-in-the-Blank -- set to "Fill-in-the-Blank"
- Question Text -- set to ".#
- Correct Answer -- set to "Ans:
- Start a New Chapter -- set to "Chapter:

Things to Note:
- Each question is separated by at least three blank lines; and the "Chapter:" and "Multiple Choice" keywords are also separated from other elements by three blank lines.
- If this is not the first question of this type in your text document, change the "1." to the next number in the sequence.
- The "Fill-in-the-Blank" keyword indicates that the questions that follow are fill-in-the-blank questions.
- In practice, formatting fill-in-the-blank questions for import is identical to formatting short answer questions. The main difference is when exporting to some assessment tools, the user is presented with a fill-in box inline with the question. The series of underscores ("____") are used to denote the placement of the fill-in box.
Setting Answer Keys in Questions

Keywords to set:
No new keywords

Things to Note:
- The special "Answer Key" keyword effectively separates the area where the answers lie with where the questions lie. In the answer key, any field, answer, or reference information may exist just as they normally would along with the questions.
- When using answer keys, questions should be numbered consecutively; the question number should match the question number used in the answer key.
- In an answer key, keywords can be on the same line or on separate lines.

Setting Information Fields in Questions

Diploma 6 lets you automate the values of pre-defined Information Fields during the RTF import process. These Information Fields can contain any information that you wish to include about your questions. Some examples in all Question Banks included by default are: Difficulty, Subject, Topic, and Type. Although you must define an Information Field before you can give it a value during RTF import, once it is defined, you are not limited to a list of preset values.

To Define an Information Field In Your Question Bank
1. Click the Tools menu from either the Question Bank or Exam Document window.
2. Select Information Fields... The Information Fields window opens.
3. Click the Tools drop-down button.
4. Select Add Field. The Add Field window opens.
5. Type the name of the Information Field that you wish to add.
6. Click OK. The Information Field is created and can now be used as a tag during RTF import.

To Set the Value of an Information Field for RTF Import
1. On the same line as the Ans: tag, type the name of the Information Field, followed by a colon.
2. Type the value that you wish to set for the Information Field when the question is imported.

Things to Note:
- In the Keywords and Options... section of the Import RTF window, the Look for keywords that occupy the same line in the source file box must be checked.
- Once you have imported a question to your Question Bank, you can always change the values for an Information Field using the Question Editor window.
- Fields are always optional; just because a field is defined in a Question Bank, you do not have to set a value for it during RTF import.
- No more than one blank line should separate Information Fields and correct answers from the questions to which they belong.
• Diploma 6 supports a virtually unlimited number of fields.
• Names of Information Fields can contain spaces.

Reference: Tables, Graphics, and Equations

Tables
You can import tables with a question by inserting the table into the RTF document itself. Imported tables maintain their original formatting settings although some Table formatting may not be supported; nested tables in particular are not recommended. Tab settings are also not supported; if an RTF file contains Tab settings, the import feature will convert them into tables, but the formatting of the resulting table will likely need to be modified. Furthermore, questions should not be placed inside tables.

Note: The first line after a table does not count toward the three lines necessary between each question. As a general rule, insert a couple of extra lines after a table if it is the last element in a question.

Images
There are two ways to insert an image into your RTF file to be imported into a question. If you edit the RTF using a word processing application that supports images, you can simply place the image in the document where you want it to appear in the question and it is automatically imported into Diploma 6 with the text of question.
Example:

True/False

1. The image below is of the Italian flag.

Ans: True Difficulty: Easy

If your word processing application does not support images or you are using a simple text editor to edit your RTF file, you can also insert an image into your RTF file using the Image tag to reference an image file. In order to use the Image tag, the image file must be in the same directory as your RTF file.

Example:

True/False

1. The image below is of the Italian flag.
[Image: italian_flag.bmp]
Ans: True Difficulty: Easy
The recommended file format for importing images is WMF (Windows metafiles), although any image file supported by your word processing application can be used. When you import images as either WMF or JPG files, they are stored unaltered within your Diploma 6 chapter files. All other file formats are converted during import to either PNG or JPG format, depending on their size.

Equations
Equations are a special object type in Diploma 6. Diploma 6 includes an Equation Editor provided by Design Science, the makers of MathType. If your RTF document contains equations authored using the Word Equation Editor (included with Microsoft Word), the Brownstone Equation Editor (included with Diploma 6), or with MathType (available from Design Science), Diploma 6 automatically recognizes the equation format and imports it correctly. These equations can be driven by Diploma 6's algorithm engine, and can also be exported to MathML for use with Wimba's EDU.

Reference: Keywords & Options Window

While the instructions in this chapter show how to use the RTF Import feature with the standard keyword tags, it is also possible to change the keywords that Diploma 6 looks for when you import an RTF file. These options can be changed from the Keywords & Options window, which is accessible through the Import RTF window.
You can also remove a keyword by deleting the text; for example, the Instruction Text keyword is blank by default. This can be useful because the fewer keywords that the program has to look for, the faster it will process your file. You can also improve the import speed by limiting the number of multiple-choice answers to the maximum number of answers used in your document.

In addition to changing keywords, the Keyword & Options window lets you adjust settings relating to paragraph breaks and keyword location. To force the program to maintain the paragraph formatting of the RTF file, check the **Preserve paragraph breaks in the source text** box. To tell the program to recognize keywords that are not separated by line breaks, check the **Look for keywords that occupy the same line in the source file** box.

**Note:** If you have been importing questions using the instructions in this guide, it’s best to not change the default keywords because doing so will affect the way in which the program recognizes the text of your RTF documents.

**Saving an RTF File for Import**

The RTF (Rich Text File) format is a file format that can be created with many common word-processing and text-editing applications including Microsoft Word, WordPerfect, and WordPad, which is included with the Windows operating system. This topic describes the process to save an RTF document using WordPad; the process is very similar in other programs.

**To Save an RTF File Using WordPad**

1. Click the **File** menu.
2. Select **Save As**. The Save As window opens.
3. From the Save As type drop-down menu, select **Rich Text Format (RTF)**.
4. Click **Save**. Your document is saved as an RTF file and is ready for RTF Import in Diploma 6.

**Importing an RTF File into Diploma 6**

Once your document is formatted correctly and saved as an RTF file, you’re ready to import it into Diploma 6. The Import RTF dialog makes it easy for you to find and import your document.

**To Import an RTF File into Diploma 6**

1. Open the Question Bank to which you wish to import the questions.
2. Click the **File** menu.
3. Select **Import > RTF**. The Import RTF window opens.
4. Click the **Browse...** button. The file browser opens.
5. Locate the directory that contains the file that you wish to import.
6. Click **OK**. The RTF files in the directory that you selected are listed in the Import RTF window.
7. Select the file from the list that you wish to import.
8. Click the **Import Selected Files** button. The questions in the RTF file are imported and appear in the Question Bank window.

**Note:** If you would like to change keywords or other import settings, click the Keywords & Options button in the Import RTF window. You can learn more about the Keywords & Options window in the topic [Reference: Keywords & Options Window](#).

---

**Importing from Diploma 5.x**

Diploma 6 has the ability to easily import Question Banks and Exams that were created in Diploma 5. This feature allows you to reuse previously created content. You can import an exam file by simply selecting Open from the File menu. You can import a Question Bank file using the Import function.

**To Import a Question Bank file Created in Diploma 5.x for Windows**

1. Click the **File** menu.
2. Select **Import > More Formats**... The Import window opens.
3. Select **Import from Exam for Windows (5.x)** from the list. The file browser opens.
4. Locate the file that you wish to import.
5. Click **Open**. The file is imported.

**Notes:**

a) Diploma 6 does not support tab settings. When a Diploma 5 Question Bank file containing tab settings is imported, tab settings are converted to tables. It is important to verify the accuracy of this conversion.

b) Since Diploma 6 supports a virtually unlimited number of fields and field values, Diploma 6 no longer supports custom question types. When a Question Bank containing custom question types is imported, the question is imported as its base question; the custom format value is converted to a Custom Format field value. No content or structural data is lost in the process.

---

**Exporting**

**Exporting to a Content Package**

Diploma 6 can export an exam or a Question Bank chapter to a content package for easy upload to a course on your Course Management System (CMS).
The supported CMS targets are:

- Angel 7
- Blackboard LS 5
- Blackboard LS 6.0 - 6.2
- Blackboard LS 6.3
- Blackboard LS 7
- Blackboard LS 8
- Blackboard LS 9
- Blackboard LS CE 3 (formerly known as WebCT 3)
- Blackboard LS CE 4 & 6 (formerly known as WebCT 4 & 6)
- Blackboard LS Vista 3 & 4 (formerly known as WebCT Vista 3 & 4)
- Moodle 1.7
- Moodle 1.8
- Moodle 1.9
- Desire2Learn
- EDU 1
- EDU 2.5+
- IMS QTI 1.2

To Export an Exam or Question Bank Chapter to a Content Package

1. Open the exam or Question Bank chapter that you wish to export.
2. Click the File menu.
3. Select Export > More Formats... The Export window opens.
4. Select the target that you wish to export your file to.
5. Click Next.
6. Click Browse. The file browser opens.
7. Select a location to save your exported content package.
8. Click Save.
9. Click Export. Your content package is exported and is ready to be uploaded to your Course Management System.

Exporting for Print or Web

Diploma 6 can export an Exam or a Question Bank Chapter to file formats for print or web use.

The print and web targets are:

- RTF for word processors
- HTML/XHTML
- DHTML/DXHTML Interactive Quiz

To Export an Exam or Question Bank for Print or Web

1. Open the exam or Question Bank chapter that you wish to export.
2. Click the File menu.
3. Select Export > More Formats... The Export window opens.
4. Select the target that you wish to export your file to.
5. Click **Next**.
6. Click **Browse**. The file browser opens.
7. Select a location to save your exported content.
8. Click **Save**.
9. Click **Export**. Your file is exported.

**RTF for Word Processing Applications**
Diploma 6’s Export to RTF feature allows you to export your Question Bank Chapter or Exam to an RTF document with all of the same options and formatting settings available when printing; all answer key options and scrambling options are available, as well as all options available in the Page Setup dialog.

**Note:** If your questions contain tables, Word 2000 or greater is required to view your document correctly.

**HTML/XHTML**
The HTML/XHTML export is useful for exporting your exams as static HTML files for printing or viewing on the web.

**DHTML/DXHTML Interactive Quiz**
The DHTML/DXHTML Interactive Quiz feature can be used to create standalone, self-grading tests that can be uploaded to any web site. This format allows students to learn through self testing.

**Exporting for Importing**
Diploma 6 can export an Exam or a Question Bank Chapter to an RTF file for use with the Import RTF feature. This type of export can be used to learn how to properly format RTF files for import, or to share editable question files with individuals without access to Diploma 6.

**To Export an Exam or Question Bank for Print or Web**
1. Open the exam or Question Bank chapter that you wish to export.
2. Click the File menu.
3. Select Export > More Formats... The Export window opens.
4. Select RTF for importing
5. Click Next.
6. Click Browse. The file browser opens.
7. Select a location to save your exported file.
8. Click Save.
9. Click Export. Your file is exported.

Using Diploma Exports with Moodle

After setting the resource URL for your course (required if your content contains images, equations, or plots) and exporting a Moodle package, you are ready to upload the package to your Moodle server and begin using it with your course.

To Import a Diploma Package into your Moodle Course

1. Log in to your Moodle server.
2. Select the course to which you wish to import your content.
3. Click the Questions link in the Administration area.
4. Click the Import tab.
5. In the 'File format' area, select Moodle XML format.
6. In the 'Import from file already in course files...' area, click the Choose or upload a file... button. A new window opens.
7. Click the Upload a file button.
8. Click the Browse... button.
9. Select the Moodle .zip package that you exported from Diploma.
10. Click the Upload this file button. The .zip is uploaded and appears in the file list.
11. Click the Unzip link to the right of the uploaded file. The contents of the package are unzipped.
12. Click the OK button. The file list appears.
13. Click the Choose link to the right of the .xml file from the uploaded package. The window closes and you are returned to the previous page.
14. In the 'Import from file already in course files...' area, click the Import from this file button. A new page opens, showing the uploaded questions.
15. At the bottom of the page, click the Continue button. The uploaded content appears in the Questions list and are ready to be used in your course.

Notes:

a) For more information on setting the resource URL for your course, see Question Content Preferences for Moodle Exports.
b) Both these instructions and Diploma’s Moodle Export is intended for use only with Moodle 1.7-1.9.
Advanced Exporting Options

Choosing a Target Browser

When exporting to the HTML/XHTML or DHTML/DXHTML Interactive Quiz formats, you have a choice of what browser level you want to target. These options affect your choices for exporting images and equations, and the type of HTML that is produced.

**HTML (3.0 browsers)**

All equations and images are exported as JPEG images for maximum compatibility.

**HTML (4.0+ browsers)**

This option gives you the option to export equations as images, WebEQ applets, or MathPlayer objects. You can also export images as JPEG or PNG.

**XHTML (Netscape 7/Mozilla 1.1)**

With this option selected, XHTML is produced. XHTML is similar to HTML, but is more restrictive. At this time, only Netscape 7 and Mozilla 1.1 fully support XHTML. When choosing this option, you can export equations as images or native MathML for use with Netscape 7 and Mozilla 1.1. You can also export images as JPEG, PNG, or GIF.

Exporting Equations Using MathML

In the past, the only option for viewing an equation object over the Internet was to convert it to an image file. While this is sufficient for many uses, the viewing and printing quality of equations was never as good as the surrounding text. The solution to this problem that is used by Diploma 6 is MathML. MathML is an XML-based language for describing mathematical equations.

Currently, the export formats that take advantage of MathML are:

- DHTML/DXHTML
- HTML/XHTML
- EDU 1.x and 2.5+

For all other exports, equations must be exported as images.

Several key advancements in MathML technologies have occurred recently. Each technology targets certain segments of web users. Each technology has strengths and weaknesses, and each supports MathML to varying degrees. When exporting to DHTML or HTML, you have a choice between these technologies.
Netscape 7 / Mozilla 1.1
The open source Mozilla browser project has native support of MathML with version 1.1. Netscape 7.0 is based on Mozilla 1.1, so Netscape 7.0 also has native support of MathML. This MathML support covers users of Windows, Mac OS, and Linux. To use MathML with Mozilla or Netscape, your HTML files must be in XHTML format. XHTML is similar to, but more restrictive than, HTML, and Mozilla and Netscape 7 are the only browsers to embrace XHTML. While older browsers can read XHTML, they ignore MathML tags. If you are not sure that all your users will be viewing your content using either Netscape 7 or Mozilla 1.1, then this is the best choice.

MathPlayer
Design Science has provided MathPlayer, a free plug-in for Internet Explorer 5.5 on Windows, which provides MathML support for Internet Explorer. MathPlayer will also integrate with MathType (from Design Science) if it is installed. If you can be sure that all your users will be viewing your content using Internet Explorer 5.5 or greater on Windows, then this is a good choice.
To download and install the MathPlayer plugin, visit the MathPlayer Download Page.

WebEQ
Design Science also has WebEQ, one of the first MathML renderers, which uses free Java applet viewers to view MathML. Since WebEQ uses Java, it works with most of the older browsers, including Netscape 4 and Internet Explorer 4 and 5. To download the WebEQ viewer applet, visit the WebEQ Download Page.

Limitations
Nearly all characters and templates available in the Equation Editor (as well as most that are available in MathType) export correctly to MathML. There are some features of Equation Editor and MathType that do not have a MathML equivalent, and there are many characters that are not well supported across all rendering technologies. Some Equation Editor and MathType features are designed for more precise control over print layout and are not supported by MathML. These include nudging, line spacing, and tab stops. In addition, overstriking of characters and templates are not supported in MathML. Some of the border controls (such as the box template and the outer border of matrices) have less granular control in MathML than they do in Equation Editor and MathType.

The conversion process also cannot convert MathType-defined sizes into MathML, due to the nature of how MathML handles sizes but you can supply an explicit point size in your equations. Color is not currently supported at this time. Other features, such as the long division template and relational operator or decimal point alignment, are defined in MathML 2.0 but not yet supported by any of the supported rendering technologies, and so they are currently not translated.

While every character available to Equation Editor can be converted correctly to MathML, not all rendering technologies support all characters. Further, since Equation Editor and MathType allow you to insert characters from any font, it is very easy to experience situations in which a MathML renderer cannot locate a character for display. This situation should improve as MathML renderers improve, and when the STIX project is completed, this situation should improve dramatically.

Some server-based assessment products, such as Wimba’s EDU, provide native MathML processing, including rendering equations dynamically to any browser on any platform.
When exporting to server-based assessment products that do not support MathML, such as Blackboard and WebCT, all equations are exported as images because all of the supporting technologies for MathML require special page-
level HTML code (or use of XHTML) or server-side hosting of java applets to function properly. If you're not sure which format to use for Equation exporting, we suggest trying each to determine which would work better for your users, and to determine which MathML viewing technology will work better for your data.

Further Information
For more information on MathML, see the official W3C specifications for MathML 1.01 and MathML 2.0 at the W3C Math Home.

Exporting Images
Diploma 6 supports many types of images, including vector images (using WMF files) and raster images (such as PNG, JPG, and GIF files). When exporting to HTML, all of your images (including any equations that will be exported as images) are converted to a raster format compatible with major browsers -- either PNG, JPG, or GIF because no browser currently supports WMF graphics.

JPEG (JPG)
JPEG graphics are the safest graphic format for export. They have excellent compression, and are supported by all web browsers. The only downside to JPEG is that it is a lossy compression format -- which means that when you export images that contain text, or other line drawing types of images, your images appear slightly distorted.

GIF
GIF graphics are one of the oldest graphic file formats still commonly found on the web, but they also have the most limitations. They are limited to 256 colors, so most high resolution graphics will lose a lot of quality when rendering in GIF. Only use GIF when it is absolutely necessary. If you need lossless compression, consider using PNG instead.

Portable Network Graphics (PNG)
PNG graphics are supported by Internet Explorer 4 and Netscape 4, as well as all modern browsers. PNG is similar to the GIF format in that it is a lossless compression format -- which means that the image does not appear distorted. The main disadvantage to PNG files is that they usually have larger files sizes than JPEG files for large images, images that contain pictures, and images with lots of colors.

Auto Select
The Auto Select option gives you the best of both formats. For any image inserted into Diploma 6 as a JPEG, that image is exported as it was inserted. For all other images, the PNG format is used, unless the PNG is too large (over 50 KB), in which case it is saved as a JPEG. Use this option to ensure that your image sizes are manageable for use on the web.

Question Content Preferences for Moodle Exports
Before exporting any content for Moodle that contains images, equations, or plots, you must set the resource URL to the location where the images will reside on your Moodle server. There are two steps to this process: determining the resource URL on your Moodle server and setting the resource URL in Diploma before generating the Moodle package. If your content does not contain any images, equations, or plots, you can leave the resource URL undefined.
Determining the Resource URL on your Moodle Server

You can determine the resource URL for a course on your Moodle server by observing the URL that appears in your browser when you access the course and then inserting specific information from that URL into the resource URL form. Resource URLs on Moodle systems use the following form:

http://[server name]/[moodle folder]/file.php/[course ID]

Resource URLs are specific to individual courses, so when obtaining specifics on your Moodle server, ensure that you access the same course to which you intend to later import content from Diploma.

To determine the resource URL for a course, log in to your Moodle server, and select the desired course. When the main course page loads, the URL in your browser will be similar to this:
http://www.myschool.edu/moodle/course/view.php?id=3

In this case, the server name is “www.myschool.edu”, the moodle folder is “moodle”, and the course id is “3”. So the Resource URL would be:
http://www.myschool.edu/moodle/file.php/3

Note:
Some installations may not have a folder for Moodle. If this is the case, simply omit it from the resource URL form.

For example, a course URL of:
http://moodle.myschool.edu/course/view.php?id=2

Would have a resource URL of:
http://moodle.myschool.edu/file.php/2

Setting the Resource URL in Diploma

Once you have determined the resource URL of the course with which you wish to use your Diploma content, you must specify this URL when exporting the package for Moodle.

To Set the Resource URL in Diploma

1. Click the File menu.
2. Select Export > More Formats... The Export window opens.
3. Select Moodle 1.7 - 1.9
4. Click the Next >> button.
5. In the 'Question Content Preferences' area, click the **Change...** button. The Questions Content Preferences window opens.

![Question Content Preferences window]

6. Type the resource URL in the Resource URL field.
7. Click the **OK** button.

After exporting to the Moodle package, follow the steps outlined in [Using Diploma Exports with Moodle](#) to ensure that the images will be viewable from your Moodle course.

### Printing Documents

**Printing**

Diploma 6 provides a number of options and features to meet different printing needs. These options are found in the Print window, accessible from the File menu. By default, if you open the Print window from a Question Bank, the current chapter is printed. If you open the Print window from an exam, the entire exam is printed.
The Print window contains several sections:

**Printer**
The drop-down list contains a list of the printers that you have installed on your computer. The Properties button displays a printer-specific dialog that allows you to set properties for your printer such as paper source, print quality, and additional printer-specific features. Use the Page Setup button to define additional print settings, including layout options and question appearance.

**Ranges**
Adjust the settings under the Ranges heading to specify which elements of your document to print. You can specify a page range or a question range.

**Versions & Copies**
Use the tools under Versions & Copies to scramble questions and/or multiple-choice answers when printing. Scrambling options can also be applied. You can also print multiple versions of your exams if necessary. If you have any questions that make use of Diploma 6's algorithm features, each version of your document can contain different versions of those questions when printed.

**Page Setup**
Diploma 6's Page Setup settings allow you to specify properties for use when printing. Page Setup preferences are stored with your Question Bank or exam document and are accessed through the Page Setup window in the File menu.
Margins & Paper
The Margins & Paper contains options for paper size, orientation, and margin settings.

Fonts
The Fonts panel contains settings for adjusting the default font and justification for your text, as well as font and justification for header and footer text. To reduce the number of pages required when printing, the standard font size can be reduced. It is recommended that you use fonts that have all extended characters available, such as Arial, Courier New, Georgia, Tahoma, Times New Roman, and Verdana.

Spacing & Numbering
The Spacing & Numbering panel lets you specify the number of lines inserted between questions. You can also specify the starting question and page number, as well as the ability to restart question numbering after every instruction.

Headers & Footers
The Headers & Footers panel allows you to specify the content that prints in the header and footer of each page.

Answer Key
The Answer Key panel allows you to specify which details are printed with the answer key.
Default Settings

When adjusting Page Setup and Question Appearance options, you have the option of storing your current settings as the default settings by clicking the **Defaults** drop-down button. These settings are used anytime you create a new exam. You can also load up your default settings into any Question Bank or Exam using the Load default Page Setup settings option.

Scrambling Options

When scrambling questions in an exam, questions always stay within the boundaries set up by any instructions you have added. If desired, you can configure scrambling to keep questions of the same question format or other information field values together when scrambling.

**To Access the Scrambling Options Dialog**

1. Click the **File** menu.
2. Select **Print**. The Print window opens.
3. Click the **Options**... button. The Scrambling Options window opens.

![Scrambling Options Dialog](image)

**Tip:** You can use the **Defaults** drop-down button to save the current settings as the default.

Question Appearance

The Question Appearance section allows you to adjust the printed appearance of your questions and is accessed through the Page Setup window in the File menu.
Each panel contains different settings. Many settings only affect a single question format. The preview area displays the changes your settings make to each question format.

**Default Settings**

When adjusting Page Setup and Question Appearance options, you have the option of storing your current settings as the default settings by clicking the **Defaults** drop-down button. These settings are used anytime you create a new exam. You can also load up your default settings into any Question Bank or exam using the Load default Page Setup window.

**Conserving Paper**

Diploma 6 features a number of settings that can be used to reduce the amount of paper required when printing exams.

Some suggestions for conserving paper are:

- Reduce the size of the margins. (see **Page Setup**)
- Use the paragraph setting for multiple-choice questions. (see **Question Appearance**)
- Allow page breaks within question text. (see **Question Appearance**)
- Minimize the amount of blank space around questions. (see **Page Setup**)
- Reduce the standard font size. (see **Page Setup**)

---

56
Plotting

About Wimba Plot

Wimba Plot allows you to create and edit math plots for insertion into your questions. You can add plots to questions much in the same way that you insert other elements, such as tables, images, and equations. You then perform edits within the Wimba Plot window. You can control the parameters of your plot items using variables to create a virtually unlimited number of questions using a single plot.

Wimba Plot supports Cartesian, Polar, and Number Line coordinate systems, and includes the following plot items:

- \( y = f(x) \)
- Parabola
- Circle
- Ellipse
- Hyperbola
- Point
- Line Segment
- Polynomial
- Parametric
- Normal Curve
- Label
- \( r = f(\theta) \)
- Point at \( (r, \theta) \)
- Ray
- Box-and-Whisker Plot
- Vector

Inserting a New Plot Into a Question

You can insert a plot into any type of question, just as you would insert tables, images, and equations. The Wimba Plot window allows you to add, edit, and manage plot items.

To Insert a New Plot Into a Question

1. Place your cursor where you wish to insert the plot in your question.
2. Click the **Insert** drop-down menu.

   ![Insert drop-down menu]

3. Select **Plot**, followed by the type of plot you would like to insert: **Cartesian**, **Polar**, or **Number Line**. The Wimba Plot window opens, showing the Add... drop-down menu.

4. Select a plot item. The item is added to your plot.

5. Adjust the plot item parameters as desired.

6. Click **OK**. The Wimba Plot window closes and the plot is inserted into your question.

**Wimba Plot Window**

The Wimba Plot window is where you add, edit, and manage your plot and plot items. The window is made up of four main sections: the Plot Item List, the Plot Display, Plot Commands, and Plot Item Parameters.
Plot Item List

The Plot Item List is where you can see all of the plot items that you have added to your plot. You can use the buttons to the right of the Plot Item List to create, delete, and duplicate plot items. In addition, you can change the order in which the plot items are displayed on the plot; click the Move Up button to move the selected plot item closer to the foreground of a plot; click the Move Down button to move it further into the background.

Plot Display

The Plot Display shows a preview of the plot as it will appear when you insert it into a question. Changes you make in the Plot List or Plot Parameters are automatically reflected in this display. You can change the display area for the plot by clicking and dragging in the preview area. You can zoom the display in and out by clicking the zoom buttons at the bottom left-hand corner of the display, or by using the scroll wheel on your mouse. Any changes you make to the plot display are reflected in the Axes Setup window.

Plot Commands

Below the preview of the plot, there are four buttons. The Axes Setup... button lets you access the Axes Setup window. Clicking the Solve button displays a new version of any plot item that uses variables. Clicking the Help button opens this user guide to the Wimba Plot chapter. Clicking the More drop-down button reveals the Load from Plot Template... and Save to Plot Template... buttons, which allow you to import previously exported plot states or save the current plot state for future use.

Plot Item Parameters

The Plot Item Parameters section is initially empty, until you add a plot item; it then displays all of the editable parameters for the selected item. In addition, the function of the plot item is displayed at the top of the section, along with a checkbox that allows you to reverse the function (if applicable), and a drop down menu that allows you to apply inequalities to the plot item.

Tip:
The different sections of the Wimba Plot window can be individually resized using the grab handles between the Plot Item List and Plot Display, and between the Plot Display and Plot Item Parameters sections.

Axes Setup Window

The Axes Setup window, which is accessible from the Wimba Plot window, allows you to set the basic characteristics of your plot. In addition to selecting the coordinate system for a plot, you can adjust a number of parameters relating to its size, labeling, and display. The two main sections of the window, Horizontal axis and Vertical axis, contain an identical set of parameters that control their respective axes.
The Coordinate System pull-down menu allows you to choose between a standard Cartesian coordinate system, a Polar coordinate system, and a Number Line system. All three are two-dimensional plotting systems. The Polar and Cartesian coordinate systems are similar, except that the Polar coordinate system uses plotting based on a radial and an angular coordinate. The Number Line system allows you to add plot items to a single-axes line. In Diploma 6, the type of coordinate system you choose affects the type of plot items that are available to you; there are more plot items available for the Cartesian coordinate system than for the Polar and Number Line coordinate systems.

Horizontal and Vertical Axes

Label
The Label field allows you to choose the characters that label the respective axes. By default, these labels are $x$ and $y$ for a Cartesian coordinate plot and $r$ for a Polar coordinate plot, but you can change them to whatever you wish.
Minimum
The Minimum field allows you to specify how much of the plot to display in the negative direction for each axis. This does not affect the size of the plot; increasing the Minimum of a plot, without increasing its Width (in inches), makes everything on the plot appear smaller. You can also adjust how much of the plot is displayed by dragging and zooming the plot preview in the Wimba Plot window.

Maximum
The Maximum field allows you to specify how much of the plot to display in the positive direction for each axis. This does not affect the size of the plot; increasing the Maximum of a plot without increasing its Width (in inches) makes everything on the plot appear smaller. You can also adjust how much of the plot is displayed by dragging and zooming the plot preview in the Wimba Plot window.

Major Ticks
The Major Ticks field allows you to specify the frequency of the larger, major tick marks along each axis of the plot.

Minor Ticks
The Minor Ticks field allows you to specify the frequency of the smaller, minor tick marks along each axis of the plot.

Font...
Clicking the Font... button opens your system’s font library and allows you to specify the font that is used to display labels on the plot.

Show Tick Marks
The Show Tick Marks checkbox allows you to enable or disable the display of tick marks on each axis.

Label Tick Marks
The Label Tick Marks checkbox allows you to specify whether or not tick marks are labeled by number along each axis.

Width (inches)
The Width (inches) field allows you to set the size of the plot. This is the absolute size that is also maintained when the plot is printed.

Grid

Show Grid
The Show Grid checkbox allows you to display a grid of dotted lines on the plot. This option is disabled by default.
Draw Grid Over Plot Items
The Draw Grid Over Plot Items checkbox allows you to control whether or not the grid is drawn on top of the plot items on the plot.

Polar Angle
The Polar angle field allows you to set the polar angle of the grid for plots using the polar coordinate system.

Other

Maintain Aspect Ratio
The Maintain Aspect Ratio checkbox allows you to specify whether or not the scales of the two axes remain proportional to each other. For most mathematical plotting, it is best to leave this box checked. However, for certain types of data plotting (such as economic plotting, in which the two axes do not need to remain in proportion), you may want to uncheck this box.

Plot Item Samples
The Plot Item Samples field allows you to control the sample level at which plot items are drawn on the plot. If the curves of plot items do not appear smooth, increase the plot item sample level.

Using Wimba Plot With Variables
Like other aspects of Diploma 6, you can use variables in questions to control the parameters of a plot items within a plot. To do this, you must first define at least one variable within a question, and then link the value of that variable to a plot item parameter within the Wimba Plot window. Calling a variable within a parameter field in the Wimba Plot window is just like calling a variable anywhere else in Diploma 6.

To Call a Variable from within the Wimba Plot Window
1. Define the variable within the question into which you wish to insert the plot.
2. Insert a new plot into the question.
3. Add a new plot item to the plot.
4. Type \[\text{[the name of the variable]}\] in any parameter field that you wish to control with the variable.

5. Click Solve. The plot item is regenerated using the value of the variable.

**Tip:**
You can call a variable from any parameter field, not just the function field.

**See Also:**
- Adding Variables to Questions
- Using Variables in Equations
- Defining Variables

**Wimba Plot Syntax**
The syntax for using functions and operators in Wimba Plot is very similar to the syntax for using functions and operators elsewhere in Diploma 6. The primary differences are that you cannot use string manipulation functions or sum() functions in Wimba Plot. Also, using the rand() function is not recommended. See the Algorithms chapter for more information on using functions and operators in Diploma 6.
Cartesian Plot Items

\[ y = f(x) \]

The \( y = f(x) \) plot item is a simple function that displays the value of \( y \) as defined by a function of \( x \).
Parameters:

\( y = \)

The \( y = \) field allows you to define the function that provides the value of the \( y \) variable to be plotted. The default value is \( x \), which generates a diagonal line as shown above. If the function is reversed, \( y = \) becomes \( x = \). Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the field.

Bounds

The Bounds parameter allows you to restrict the range along the \( x \) axis in which the equation is displayed on the plot. The default setting is \(-\infty \) to \( \infty \), meaning that the plot item is displayed along the entire area of the plot. Entering numeric values defines the area in which the plot item is displayed. If the function is reversed, the Bounds parameter is also reversed, and is defined based on the \( y \) axis instead of the \( x \) axis.

Style

The parameters in the Style section define the appearance of the plot item. Line size allows you to set the thickness (in pixels) of plotted lines. Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. The none style makes the line invisible and can be useful for displaying inequalities. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot. Fill color allows you to set the color of the shaded area for plots using inequalities.

Inequalities

While the default setting for the \( y = f(x) \) plot item is that of an equality function, you have the option of changing it to an inequality function using the Inequalities drop-down menu. Changing the equality to a less than, greater than, less than or equal to, or greater than or equal to function results in a shaded area on the plot, representing the inequality.

Function Reversal

The \( x = f(y) \) checkbox, located below the Inequalities drop-down menu, allows you to reverse the variables in the function so that the value that is plotted is \( x \) as defined by a function of \( y \).

See Also:

- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

Parabola

The Parabola plot item displays a conic based on the value of the following equation:

\[(y - k)^2 = 4p(x - h)\]
Parameters:

**h =**
The h = field allows you to define the x the coordinate of the parabola’s vertex. If the function is reversed, h = defines the y coordinate of the vertex.

**k =**
The k = field allows you to define the y coordinate of the parabola’s vertex. If the function is reversed, k = defines the x coordinate of the vertex.

**p =**
The p = field allows you to define the distance from the vertex to focus. Visually, setting a higher p value makes a parabola wider, while a lower p value makes a parabola narrower.

Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the fields.
**Bounds**

The **Bounds** parameter allows you to restrict the range along the x and y axes in which the parabola is displayed on the plot. The default settings are \(-\infty\) to \(\infty\), meaning that the parabola is displayed along the entire area of the plot. Entering numeric values defines the area in which the parabola is displayed.

**Style**

The parameters in the **Style** section define the appearance of the parabola on the plot. Line size allows you to set the thickness (in pixels) of plotted lines. Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. The none style makes the line invisible and can be useful for displaying inequalities. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot. Fill color allows you to set the color of the shaded area for plots using inequalities.

**Inequalities**

While the default setting for the Parabola plot item is that of an equality function, you have the option of changing it to an inequality function using the Inequalities drop-down menu. Changing the equality to a less than, greater than, less than or equal to, or greater than or equal to function results in a shaded area on the plot, representing the inequality.

**Function Reversal**

The \(x = f(y)\) checkbox, located below the Inequalities drop-down menu, allows you to reverse the x and y variables in the function so that the conic is based on the following equation:

\[
(x - k)^2 = 4p(y - h)
\]

---

See Also:
- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

**Circle**

The Circle plot item displays a conic based on the value of the following equation:

\[
(x^2 - h) + (y^2 - k) = r^2
\]
Parameters:

**h =**
The h = field allows you to define the x coordinate of the circle's center.

**k =**
The k = field allows you to define the y coordinate of the circle's center.

**r =**
The r = field allows you to define the radius of the circle.

Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu \( \frac{\text{function}}{\text{operator}} \) to easily insert functions into the fields.

**Bounds**
The Bounds parameter allows you to restrict the range along the x and y axes in which the circle is displayed on the plot. The default settings are -inf to inf, meaning that the circle is displayed along the entire area of the plot. Entering numeric values defines the area in which the circle is displayed.
Style

The parameters in the Style section define the appearance of the circle on the plot. Line size allows you to set the thickness (in pixels) of plotted lines. Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. The none style makes the line invisible and can be useful for displaying inequalities. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same graph. Fill color allows you to set the color of the shaded area for plots using inequalities.

Inequalities

While the default setting for the Circle is that of an equality function, you have the option of changing it to an inequality function using the Inequalities drop-down menu. Changing the equality to a less than, greater than, less than or equal to, or greater than or equal to function results in a shaded area on the plot, representing the inequality.

Function Reversal

The $x = f(y)$ checkbox, located below the Inequalities drop-down menu, allows you to reverse the x and y variables in the function so that the conic is based on the following equation:

$$(y^2 - h) + (x^2 - k) = r^2$$

See Also:

- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

Ellipse

The Ellipse plot item displays a conic based on the value of the following equation:

$$(x - h)^2 / a + (y - k)^2 / b = 1$$
Parameters:

\( h = \)
The \( h = \) field allows you to define the \( x \) coordinate of the ellipse’s center.

\( k = \)
The \( k = \) field allows you to define the \( y \) coordinate of the ellipse’s center.

\( a = \)
The \( a = \) field allows you to define the radius of the ellipse along the \( x \) axis. If the function is reversed, \( a = \) defines the radius of the ellipse along the \( y \) axis.

\( b = \)
The \( b = \) field allows you to define the radius of the ellipse along the \( y \) axis. If the function is reversed, \( b = \) defines the radius of the ellipse along the \( x \) axis.

Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the fields.
**Bounds**

The Bounds parameter allows you to restrict the range along the x and y axes in which the ellipse is displayed on the plot. The default settings are $-\infty$ to $\infty$, meaning that the ellipse is displayed along the entire area of the plot. Entering numeric values defines the area in which the ellipse is displayed.

**Style**

The parameters in the Style section define the appearance of the ellipse on the plot. Line size allows you to set the thickness (in pixels) of plotted lines. Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. The none style makes the line invisible and can be useful for displaying inequalities. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot. Fill color allows you to set the color of the shaded area for plots using inequalities.

**Inequalities**

While the default setting for the Ellipse plot item is that of an equality function, you have the option of changing it to an inequality function using the Inequalities drop-down menu. Changing the equality to a less than, greater than, less than or equal to, or greater than or equal to function results in a shaded area on the plot, representing the inequality.

**Function Reversal**

The $x = f(y)$ checkbox, located below the Inequalities drop-down menu, allows you to reverse the variables in the function so that the conic is based on the following equation:

\[
\frac{(y - k)^2}{a} + \frac{(x - h)^2}{b} = 1
\]

**See Also:**
- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

**Hyperbola**

The Hyperbola plot item displays a conic based on the value of the following equation:

\[
\frac{(x - h)^2}{a^2} - \frac{(y - k)^2}{b^2} = 1
\]
Parameters:

h =
The h = field allows you to define the x coordinate of the hyperbola's center. If the function is reversed, h = defines the y coordinate of the hyperbola's center.

k =
The k = field allows you to define y coordinate of the hyperbola's center. If the function is reversed, k = defines the x coordinate of the hyperbola's center.

a =
The a = field allows you to define the semi-major axis of the hyperbola, which is half the distance between the two arms of the hyperbola at its center (the major axis). If the function is reversed, a = defines the semi-minor axis.

b =
The b = field allows you to define the semi-minor axis of the hyperbola. Visually, setting a higher b value makes the arms of the parabola curve away from each other less, while setting a lower b value makes the arms curve away from each other more. If the function is reversed, b = defines the semi-major axis, which is half the distance between the two arms of the hyperbola at its center (the major axis).
Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the fields.

**Bounds**
The Bounds parameter allows you to restrict the range along the x and y axes in which the hyperbola is displayed on the plot. The default settings are \(-\infty\) to \(\infty\), meaning that the hyperbola is displayed along the entire area of the plot. Entering numeric values defines the area in which the hyperbola is displayed.

**Style**
The parameters in the Style section define the appearance of the hyperbola on the plot. Line size allows you to set the thickness (in pixels) of plotted lines. Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. The none style makes the line invisible and can be useful for displaying inequalities. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot. Fill color allows you to set the color of the shaded area for plots using inequalities.

**Inequalities**
While the default setting for the Hyperbola plot item is that of an equality function, you have the option of changing it to an inequality function using the Inequalities drop-down menu. Changing the equality to a less than, greater than, less than or equal to, or greater than or equal to function results in a shaded area on the plot, representing the inequality.

**Function Reversal**
The \(x = f(y)\) checkbox, located below the Inequalities drop-down menu, allows you to reverse the variables in the function so that the conic is based on the following equation:

\[
\frac{(y - k)^2}{a^2} - \frac{(x - h)^2}{b^2} = 1
\]

**See Also:**
- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

**Point**
Point displays a single point on the plot based on two coordinates.
**Parameters:**

**Point**
The Point parameters allow you to define the location and appearance of the point on the plot. The x: and y: fields define the coordinates of the point. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point’s symbol; the size field controls the overall size of the point’s symbol. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the x and y field.

**Label**
The Label parameters allow you to define the location and appearance of the label that accompanies the point on the plot. The type drop-down menu allows you choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system’s font manager and allows you specify a font for the label.
Line Segment

The Line Segment plot item allows you to plot two points with a line between them. The points, their labels, and the line are all fully customizable.

![Diagram of a line segment with points (1, 1) and (3, 1) on the x-axis.]

**Parameters:**

**Start Point**
The Start Point parameters allow you define the location and appearance of the starting point of the line segment. The x: and y: fields define the coordinates of the point on the plot. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point’s symbol; the size field controls the overall size of the point’s symbol. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the x and y fields.
Start Label
The Start Label parameters allow you to define the location and appearance of the label that accompanies the Start Point on the plot. The type drop-down menu allows you choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system's font manager and allows you specify a font for the label.

End Point
The End Point parameters allow you define the location and appearance of the ending point of the line segment. The x: and y: fields define the coordinates of the point on the plot. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point’s symbol; the size field controls the overall size of the point’s symbol. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the x and y fields.

End Label
The End Label parameters allow you to define the location and appearance of the label that accompanies the End Point on the plot. The type drop-down menu allows you choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system’s font manager and allows you specify a font for the label.

Style
The parameters in the Style section define the appearance of the line on the plot. Line size allows you to set the line's thickness (in pixels). Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.

See Also:
Using Wimba Plot with Variables
Wimba Plot Syntax
Axes Setup Window

Polynomial
The Polynomial plot item displays the line created by connecting up to eight separate plotted points.
Parameters:

Define Using
The Define Using drop-down menu allows you to choose a method for defining the polynomial. If you choose Points, the polynomial crosses through each defined x and y coordinate pair. If you choose Zeros, the polynomial crosses the x axis at each defined x coordinate. If you choose Critical points, the polynomial has a critical point (i.e. the slope is zero at each defined x coordinate). If you choose Zeros Of The 2nd Derivative, the polynomial has a point of inflection, i.e. the slope will change sign, at each defined x coordinate.

Points
The Points fields are where you enter the coordinates of the points that you wish to plot.

Scale
If you set Define Using to Zeros, Critical Points, or Zeros of the 2nd Derivative, you can use the Scale field to define the slope of the lines as they approach each curve; the larger the scale value, the narrower each curve.

y-intercept
If you set Define Using to Critical Points or Zeros of the 2nd Derivative, you can use the y-intercept field to set where the polynomial intersects with the y axis.
If you set Define Using to Zeros of the 2nd Derivative, you can use the slope at x=0 field to set the severity of the slope at x=0.

Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the fields.

**Labeling Point**

The drop-down menu allows you to choose one of the points that you have plotted to add a label to.

**Point [Labeling Point Value]**

The Point parameters allow you define the location and appearance of the point selected in the Labeling Point drop-down menu. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point’s symbol; the size field controls the overall size of the point’s symbol.

**Label [Labeling Point Value]**

The Label parameters allow you to define the location and appearance of the label that accompanies the point. The type drop-down menu allows you choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system’s font manager and allows you specify a font for the label.

**Bounds**

The Bounds parameter allows you to restrict the range along the x axis in which the plot item is displayed on the plot. The default setting is -inf to inf, meaning that the it is displayed along the entire area of the plot. Entering numeric values defines the area in which the plot item is displayed.

**Style**

The parameters in the Style section define the appearance of the plot item. Line size allows you to set the thickness (in pixels) of the line. Line Pattern allows you to set the style of the line. In addition to the standard solid line, you can choose dashed, dotted, or none styles. The none style makes the line invisible and can be useful for displaying inequalities. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.

**See Also:**

- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window
Parametric

The Parametric plot item displays the value of the equations $x = f(t)$, $y = f(t)$. Parametric plotting generates $x$ and $y$ values independently based on a range of values for $t$.

![Parametric Graph]

**Parameters:**

**t low**
The $t$ low field allows you to define the low value for $t$. The default value is $-\pi$.

**t high**
The $t$ high field allows you to define the high value for $t$. The default value is $\pi$.

**x**
The $x =$ field allows you to define the value for the $x$ equation. The default value is $\cos(3 \cdot t) \cdot 4$.

**y**
The $y =$ field allows you to define the value for the $y$ equation. The default value is $\sin(5 \cdot t) \cdot 4$. 
Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the x and y fields.

**Style**

The parameters in the Style section define the appearance of the plot item. Line size allows you to set the thickness (in pixels) of lines. Line Pattern allows you to set the style of lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.

**See Also:**
- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

**Normal Curve**

The Normal Curve displays the value of an equation resulting in what is commonly referred to as a Bell Curve or a Gaussian Curve.
Parameters:

Mean (μ) =
Setting the mean value for μ allows you to define the center point of the curve on the x axis.

Std. Deviation (sigma) =
Setting the standard deviation affects the distribution represented by the curve. A lower standard deviation results in a curve narrowly distributed around the mean. A higher standard deviation results in a curve more widely distributed around the mean.

Shade from sigma * to sigma *
Setting numeric values for these fields allows you to shade a range from one multiple of the standard deviation to another multiple of the standard deviation.

Scale
The Scale field allows you to define the overall scale of the curve.

Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the parameters fields.

Style
The parameters in the Style section define the appearance of the curve on the plot. Line size allows you to set the thickness (in pixels) of the line. Line Pattern allows you to set the style of the line. In addition to the standard solid line, you can choose dashed, dotted, or none styles. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot. Fill color allows you to set the color of the shaded areas within the curve (if applicable).

See Also:
Using Wimba Plot with Variables
Wimba Plot Syntax
Axes Setup Window

Label
Label is not a plot item based on an equation, but instead is a line of text on the plot centered at a location defined by two coordinates.
**Parameters:**

**Point**
The Point parameters allow you define the location of the label using the x and y fields. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the fields.

**Label**
The Label parameters allow you to define the appearance and the text. The type drop-down menu allows you choose what type of label is displayed at the coordinates defined. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system's font manager and allows you specify a font for the label.

**See Also:**
- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window
Polar Plot Items

\[ r = f(\theta) \]

The \( r = f(\theta) \) plot item displays the value of \( r \) as defined by a function of \( \theta \).

**Parameters:**

**Function**
The function field allows you to define the function that \( r \), the radius, is derived from. The default value is \( t \), representing the angle theta. Use \( t \) to represent theta in your polar expressions. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu \( \sqrt{a} \) to easily insert functions into the field.

**Bounds**
The Bounds parameter allows you to restrict the range along the x axis in which the plot item is displayed on the plot. The default setting is \( 0 \) to \( 2 \times \pi \). Entering numeric values defines the area in which the plot item is displayed.

**Style**
The parameters in the Style section define the appearance of the plot item. Line size allows you to set the thickness (in pixels) of the line. Line Pattern allows you to set the style of the lines. In addition to the standard solid line, you
can choose dashed, dotted, or none styles. The none style makes the line invisible and can be useful for displaying inequalities. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.

See Also:
Using Wimba Plot with Variables
Wimba Plot Syntax
Axes Setup Window

Point at \((r, \theta)\)

Point at \((r, \theta)\) displays a single point on the plot based on two coordinates. It is similar to the Point on a Cartesian plot, but uses \(r\) and \(\theta\) instead of \(x\) and \(y\) coordinates.

Parameters:

Point
The Point parameters allow you define the location and appearance of the point on the plot. The \(r\) and \(\theta\) fields, the radius and angle fields respectively, define the polar coordinates of the point on the plot. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point’s symbol; the size field controls the
overall size of the point’s symbol. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the r and θ fields.

**Label**

The Label parameters allow you to define the location and appearance of the label that accompanies the point on the plot. The type drop-down menu allows you choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system’s font manager and allows you specify a font for the label.

---

**See Also:**

- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

**Label at (r, θ)**

Label is not a plot item based on an equation, but instead is a line of text on the plot centered at a location defined by two coordinates.
Parameters:

Point
The Point parameter allow you set the location of the label on the plot by using the \( r \) and \( \theta \) fields, the radius and angle fields respectively, define the polar coordinates of the point on the plot. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu \( \sqrt{\text{ }} \) to easily insert functions into the fields.

Label
The Label parameters allow you to define the appearance and the text. The type drop-down menu allows you choose what type of label is displayed at the coordinates defined. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system's font manager and allows you specify a font for the label.

See Also:
Using Wimba Plot with Variables
Wimba Plot Syntax
Axes Setup Window

Number Line Plot Items
Point
Point displays a single point on the plot based on a given x coordinate on the number line.

Parameters:

Point
The Point parameters allow you define the location and appearance of the point on the plot. The \( x \) field defines the coordinate of the point. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point's symbol; the size field controls the overall size of the point's symbol. Wimba Plot syntax supports most of the...
functions and operators available elsewhere in Diploma 6. You can use the drop-down menu \[ \sqrt{} \] to easily insert functions into the x field.

Label
The Label parameters allow you to define the location and appearance of the label that accompanies the point on the plot. The type drop-down menu allows you to choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system’s font manager and allows you to specify a font for the label.

**See Also:**
- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

Segment
The Segment plot item allows you to plot two points with a line between them on the number line. The points, their labels, and the line are all fully customizable.

\[
\begin{array}{ccccccc}
-5 & -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

**Parameters:**

Start Point
The Start Point parameters allow you to define the location and appearance of the starting point of the line segment. The x: field defines the coordinate of the point on the plot. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point’s symbol; the size field controls the overall size of the point’s symbol. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu \[ \sqrt{} \] to easily insert functions into the x field.
**Start Label**

The Start Label parameters allow you to define the location and appearance of the label that accompanies the Start Point on the plot. The type drop-down menu allows you choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the **Font...** button opens your system’s font manager and allows you to specify a font for the label.

**End Point**

The End Point parameters allow you define the location and appearance of the ending point of the line segment. The x: field defines the coordinate of the point on the plot. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point’s symbol; the size field controls the overall size of the point’s symbol. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu \( \sqrt{x} \) to easily insert functions into the x field.

**End Label**

The End Label parameters allow you to define the location and appearance of the label that accompanies the End Point on the plot. The type drop-down menu allows you choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the **Font...** button opens your system’s font manager and allows you to specify a font for the label.

**Style**

The parameters in the Style section define the appearance of the line on the plot. Line size allows you to set the line’s thickness (in pixels). Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.

---

**See Also:**
- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

**Ray**

The Ray plot item allows you to plot a line on the number line with that originates at a given point and continues infinitely in one direction. The point, its label, and the line are all fully customizable.
Parameters:

Point
The Point parameters allow you define the location and appearance of the origin point on the plot. The x: field defines the coordinate of the point. The style drop-down menu allows you to set the symbol type for the point. The color field allows you to set the color of the point. The width field controls the width (in pixels) of the lines that make up the point's symbol; the size field controls the overall size of the point's symbol. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu \( \sqrt{x} \) to easily insert functions into the x field.

Label
The Label parameters allow you to define the location and appearance of the label that accompanies the point on the plot. The type drop-down menu allows you choose what type of label is displayed next to the point. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system's font manager and allows you specify a font for the label.

Direction
The direction drop-down menu allows you to set whether you want the ray to go to the left or the right of the point.

Style
The parameters in the Style section define the appearance of the line on the plot. Line size allows you to set the line's thickness (in pixels). Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.

See Also:
Using Wimba Plot with Variables
Wimba Plot Syntax
Axes Setup Window
Line
The Line plot item allows you to plot a line on the number line that continues infinitely in both directions. You can customize the size and appearance of the line.

![Line plot example]

Style
The parameters in the Style section define the appearance of the line on the plot. Line size allows you to set the line’s thickness (in pixels). Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.

See Also:
Axes Setup Window

Box-and-whisker Plot
The Box-and-whisker plot item (sometimes referred to as a Box plot) allows you to plot data based on its five-number summary, consisting of the lower extreme, lower quartile, median, upper quartile, and upper extreme.

![Box-and-whisker plot example]

Parameters
The parameters allow you to define the positions of the elements that make up the Box-and-whisker plot item. Lower extreme defines the left edge of the left whisker. Lower quartile defines the left edge of the box. Median defines the location of the tick inside the box. Upper quartile defines the right edge of the box. Upper extreme defines the right edge of the right whisker. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the fields.
**Style**

The parameters in the Style section define the appearance of the lines on the plot. Line size allows you to set the line’s thickness (in pixels). Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.

**See Also:**

- Using Wimba Plot with Variables
- Wimba Plot Syntax
- Axes Setup Window

**Vector**

The Vector plot item (also known as a spatial vector) allows you to plot a graphical representation of the magnitude and direction between two points.

![Vector Diagram](image)

**Range**

$x_0$ defines the point of origin of the vector. $x_1$ defines the end point of the vector. Wimba Plot syntax supports most of the functions and operators available elsewhere in Diploma 6. You can use the drop-down menu to easily insert functions into the fields.

**Label**

The Label parameters allow you to define the location and appearance of the label that accompanies the plot item. The type drop-down menu allows you to choose what type of label is displayed. If you select a label containing text, you can type the text in the box below the drop-down menu. The color field allows you to set the color of the label. The radio buttons in the site field allow you to set the location of the label on the plot in relation to the point. Clicking the Font... button opens your system’s font manager and allows you specify a font for the label.

**Style**

The parameters in the Style section define the appearance of the line on the plot. Line size allows you to set the line’s thickness (in pixels). Line Pattern allows you to set the style of plotted lines. In addition to the standard solid line, you can choose dashed, dotted, or none styles. Line color allows you to set the color of the line, which can be especially useful for distinguishing between multiple plot items on the same plot.
Algorithms

Overview: Algorithms

Diploma 6 includes features that allow questions to be controlled by algorithms. This permits a question to act as a template for potentially thousands of variations.

A Solve Variables button is visible whenever a question is being displayed that uses algorithms. Clicking this button results in the program generating a variation of the question and displaying it as it would appear to a student. By clicking the More button and the Variables tab, you can see how the individual variables used in the question are defined.
**Note:** When you click the Solve Variables button, values are not substituted permanently. You can see other values by clicking the Solve Variables button again. To see the original variables, click anywhere within the text of the question.

**Adding Variables to Questions**

Any question can be turned into an algorithm-based question by adding variables to its text and then defining the variables.

**To Add a Variable to a Question:**

1. Open the question to which you wish to add variables.
2. Select the word in the text of the question that you wish to turn into a variable by double-clicking it.
3. Press F12 on your keyboard. The selected word changes color to blue and is underlined.

4. Click the View drop-down menu
5. Select Variables. The Variables tab appears.

**Defining Variables**

After adding a variable to a question, you can use the Variables tab to define or modify it.
To Define a Variable:

1. Highlight the variable that you wish to define by selecting it from the list on the left side of the Variables tab. The variable's definition appears on the right side of the Variables tab.

2. If the type of the variable has not already been defined, choose a variable type from the Type box:
   - To confine the variable to a range of consecutive numeric values, select **Range-Based**.
   - To confine the variable to a list of possible values, select **List-Based**.
   - To make the value of the variable a function of an equation, select **Equation-Based**.
   - To make the variable insert a dynamically selected image, select **Image-List**.
   - To make the value of the variable a constant, select **Constant**.

3. After defining the type of the variable, you can enter in the values and expressions for that variable type, as well as define formatting attributes.

4. Click the **Solve Variables** button to verify the values that are substituted in place of the variable.

Using Variables In Equations

Variables can also be used inside of equations authored using the Brownstone Equation Editor (included with Diploma 6) or MathType (available from Design Science). This unique capability allows you to dynamically generate equations with values from Diploma 6's algorithm feature.
To insert a variable into an equation, enclose the variable between \[$\] and \]$.

A popular use of this feature is to generate dynamic fractions, as shown in the example below. The Reduce() function is designed to ease this task.

The above text, when solved, is converted to the following:

**Types of Definitions**

**Range-Based Variables**

When a numerical variable's value is confined between a lower and an upper limit, it can be defined as a range-based variable.
The Low Value and the High Value are required. The Low Value indicates the lowest possible value. The value can never be higher than the High Value. You may specify how values for the variable are incremented. In the example above, the variable X is being incremented in units of .5. (Unless otherwise specified, range-based variable values are incremented by "1.")

You may also indicate values within the range that should not be selected. These values should be typed into the Disallow area. The Low Value, High Value, Increment, and Disallow Value(s) can all be expressions that use other variables, as long as the expression evaluates to a number.

See Also:
Formatting Attributes.

List-Based Variables

When a variable's value is confined to a set of text values, and/or when the value of the variable should correspond with the value of another, the variable should be defined as a list-based variable.

You must specify at least one item. Items can be any type of number, string, or expression, and must be separated by commas. You can use the Group to make one or more list-based variables correspond to each other. For example, you might have a variable "SideA" with the values 3, 6, and 8, and a variable "SideB" with the values 4, 8, and 15. If you wanted the variables to correspond the elements (so that 3 and 4 are used together, 6 and 8 are used together, and 8 and 15 are used together), you can give them both the group value of "Sides".

See Also:
Formatting Attributes.
Equation-Based Variables

When a numerical variable’s value is determined by performing a computation, the variable should be defined as an equation-based variable.

Equations can be built from a combination of mathematical operators, functions, constants, and other variables. In the example above, Total is an equation-based variable that uses the addition operator, "+", the multiplication operator, "\times", and two other variables: X and Y.

See Also:
Formatting Attributes.

Image List Variables

When different images need to be displayed within a question, an image-list variable can be defined. Image-list variables operate very similarly to list-based variables, except instead of selecting from a pre-defined list of text or numbers, image-list variables randomly select from a list of pre-defined graphic images.

To add images to the list, click the Add Image button. The order of the images can be adjusted by dragging the image names up or down. You can use the Group option to make an image-list variable correspond to other image-list variables and/or to list-based variables. Note how the values of the list-based variable below correspond to the images above. Note especially that both variables belong to the same group: Objects.
Constants

Some academic disciplines make frequent use of the same fixed values (constants). These can be defined within Diploma 6. By defining a constant, an often lengthy numeric value can be replaced with a simple name.

Constants behave just like variables, except that their value cannot contain expressions. A constant's value may be a simple number, a number in scientific notation (as in the example above), or a string.

Constants can be defined in a "global" scope so that they can be used by multiple questions.

User-Defined Functions

User-defined functions can be added as needed to Diploma 6's list of built-in functions.

In the example above, SideC is an equation-based variable that makes use of a user-defined function called Hypotenuse. The name of the function is followed immediately by a set of parentheses. The parentheses contain the arguments that are used by the function. Note that the argument names also appear in the function’s definition. When multiple arguments are passed to a user-defined function, separate them with commas in the function’s name as shown.

Functions that are generic are best used in a "global" scope, so that they can be used by other questions.
Conditions

Sometimes conditions must be met before the solution to an algorithm can be considered acceptable. Condition statements make it relatively easy to address this need.

Condition statements usually define a relationship between two expressions: equality (\( = \)), inequality (\( <> \)), or relative magnitude (\( < \), \( > \), \( <= \), or \( >= \)). If the condition’s expression evaluates to a non-zero or true value, then the condition is satisfied. If the expression evaluates to zero or false, then the condition is not satisfied, and the entire question is re-solved with a different set of values. In the example above, the condition statement says that the value of A cannot be equal to the value of B. Look at the definition for C in the example above. If A and B were ever to have the same value, the denominator in C’s definition would be zero -- causing a division-by-zero problem. The Condition statement prevents this from happening. An algorithm-based question can use as many condition statements as it needs.

It is possible to write conditions that are so restrictive that very few or even no possibilities exist for the question. If Diploma 6 detects that it’s taking too long to generate a series of values that satisfy all conditions, an error is be raised. When this happens, you should modify your definitions. If necessary, you may want to convert some of your variables to list-based variables, with which you can specify a series of “acceptable” values, and remove the need for conditions.

Formatting Attributes

All types of variables and constants can have formatting attributes applied to them. Formatting attributes don’t alter the value of the variable or constant, but they do affect how the value is formatted when the variable is replaced inside of the text of the question.

You can edit a variable’s formatting attributes while editing the variable’s definition.
In the example above, the variables A1, A2, and A3 use the value of pi. A1 uses the default formatting. A2 and A3 set the decimal amounts explicitly. When you solve a question that contains these variables, their text is “3.141593”, “3”, and “3.142”. By default, Diploma 6 displays no more than six decimal places. To set the number of decimal places used when displaying a variable, use the **Decimal Places** option when defining a variable:

![Decimal Places Option](image)

In the example above, the variables B1, B2, and B3 use the value 10000000 (10 million). B1 uses the default formatting. B2 uses scientific notation, while B3 uses the "separated by thousands" feature.

When you solve a question that contains these variables, their text is “10000000”, “1e7”, and “10,000,000”. To set the formatting used when displaying a variable, use the **Format** option when defining a variable:

![Format Option](image)

**See Also:**

*Formatting Coefficients and Polynomials*

**Formatting Coefficients and Polynomials**

Diploma 6 provides formatting attributes to make it easier to create dynamic polynomial expressions. These polynomials can also be used inside equations.

![Format Option](image)

These attributes make it easy to turn a question defined like this:
To format coefficients, simply define the formatting attribute of each variable according to how the coefficient appears in the expression - whether it is a leading coefficient, a coefficient in the middle of an expression, or a constant coefficient at the end of an expression. In each case, the sign is added automatically, and zero is automatically disallowed. In the question text, the "+" signs are added as part of the variable for readability only. They could have been omitted and the result would have been exactly the same.

In the case above, the definitions would look like this:

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>-10 to 10 [lead coefficient]</td>
</tr>
<tr>
<td>b</td>
<td>-10 to 10 [coefficient]</td>
</tr>
<tr>
<td>c</td>
<td>-10 to 10 [constant coefficient]</td>
</tr>
<tr>
<td>da</td>
<td>2^a [lead coefficient]</td>
</tr>
<tr>
<td>db</td>
<td>b [constant coefficient]</td>
</tr>
</tbody>
</table>

**Formatting Scientific Notation**

**Using a Custom Style for Displaying Scientific Notation**

Typically, Diploma displays numbers in scientific notation in a common computer programming style using E notation instead of exponential notation. This format is the same as the one used to define scientific notation in Diploma’s algorithm engine. For example: Avogadro’s constant would be defined as 6.02214179e23 in the Diploma engine. By default Diploma uses six decimal places, and automatically uses scientific notation for very large (or small) numbers. For example, using the constant directly displays the following output:

6.022142e23
You can adjust the number of decimal places by adjusting the formatting for the variable you’re using. For instance, the following is the same number formatted to three decimal places:

6.022e23

If you want to create a custom style for displaying scientific notation, such as for displaying it in exponential style, you can use the mantissa() and exponent() functions to format the number in any way that you like. For example:

mantissa(avogadro) = 6.022142
exponent(avogadro) = 23

By then combining the output of these functions, you can create your own custom format. The basic form for standard scientific notation would be: mantissa(x) × 10\(^{\text{exponent}(x)}\). For example:

mantissa(avogadro) × 10\(^{\text{exponent}(avogadro)}\) = 6.022142 × 10\(^{23}\)

**Using Custom Scientific Notation with Variables and Equations**

When you mark an expression as a variable (using F12, or Format -> Character -> Variable on the menu), all other formatting is lost. To retain formatting when the expression is solved, select the expression after marking the variable and format it as superscript a second time.

You can also use a similar approach within an equation. For example, an equation defined like this:

\[
\frac{\text{mantissa}(\text{avogadro}) \times 10^{\text{exponent}(\text{avogadro})}}{15}
\]

displays as:

\[
\frac{6.022142 \times 10^{23}}{15}
\]

**Formatting Reduced Fractions**

Reduced fractions are handled in Diploma using the reduce() function. For example, if we have two variables, a and b, and we want to put them into a fraction inside an equation, this equation:

\[
\frac{\text{a}}{\text{b}}
\]

displays as:

\[
\frac{20}{25}
\]
However, if we want the fraction to display in reduced form. We can specify this by adding the reduce() function to both the numerator and the denominator, causing this equation:

\[
\frac{\text{reduce}(1, a, b)}{\text{reduce}(2, a, b)}
\]

to display as:

\[
\frac{4}{5}
\]

**Answer-Letter Variables**

When writing feedback or hints for multiple-choice questions, it is common to refer to the answer letters of the correct or incorrect answer. When using multiple-choice answer scrambling, this can cause incorrect text. When referring to a multiple-choice answer letter, use the special pre-defined variables \(_A\), \(_B\), \(_C\), \(_D\), etc. These variables are assigned the answer letter that the answer is using in the version of the question being displayed. So if answers A through D are scrambled in reverse order, then \(_A = \text{“D”}, \_B = \text{“C”}, \_C = \text{“B”},\) and \(_D = \text{“A”}\). These built-in variables are available for each multiple-choice question, and one is defined for each answer choice in the question.

To convert a regular "A" to the variable \(_A\), simply type an underscore before the A, highlight the underscore and the A, and press **F12** (create variable). The appearance changes to indicate that it is a variable.

**Algorithm Reference**

**Legal Names for Variables, Constants, and Functions**

The names of variables, constants, and functions can contain letters, digits, and underscore characters. The names should not begin with a digit or an underscore, nor should the names contain operators, spaces, or other forms of punctuation.

<table>
<thead>
<tr>
<th>Legal Names</th>
<th>Illegal Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnswerA, TotalPrice, Student_Name, Player_2</td>
<td>Answer A, Total-Price, Student.Name, 2ndPlayer</td>
</tr>
</tbody>
</table>

**Note:** The names of variables, constants, and functions are not case-sensitive. A variable written as "AverageSize" could be referred to as "AVERAGEsize" with the same results.
Order of Execution

In determining the order of execution of functions and operators in an expression, Diploma 6’s algorithm engine considers five general levels of precedence:

1. Expressions inside of the inner-most parentheses are evaluated first.
2. Functions are evaluated next.
3. Arithmetic operators are evaluated next (in their order of precedence).
4. Relational operators are evaluated next (in their order of precedence).
5. Boolean logic operators are evaluated last (in their order of precedence).

As indicated, additional layers of precedence are embedded within some of the general levels (for example, in the arithmetic operator level, division operations are carried out before subtraction operations). Operators located at the same level are evaluated in left-to-right order (for example, multiplication and division are located at the same level and would be evaluated in order of appearance when read left-to-right).

Pre-Defined Constants

A number of constants have been pre-defined for use within Diploma 6. These constants can be used in the algorithms of any question in any question bank. Their names and values are listed below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pi</td>
<td>3.14159265358979323846264338327950</td>
</tr>
<tr>
<td>e</td>
<td>2.71828182845904523536028747135266</td>
</tr>
<tr>
<td>True, Yes, On</td>
<td>1</td>
</tr>
<tr>
<td>False, No, Off</td>
<td>0</td>
</tr>
<tr>
<td>endash</td>
<td>&quot;−&quot;</td>
</tr>
</tbody>
</table>

In addition to these constants, platform-specific constants are also available, so that you can customize your algorithms depending on what context they are running in:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_win</td>
<td>Set to 1 when running on a Windows platform.</td>
</tr>
<tr>
<td>_web</td>
<td>Set to 1 when running on a Web platform (such as Brownstone’s EDU server).</td>
</tr>
<tr>
<td>_mac</td>
<td>Set to 1 when running on a Macintosh platform.</td>
</tr>
</tbody>
</table>

Scope of Variables, Constants, and Functions

Scope refers to how far the definition of a variable, constant, or function can be applied. The scope of definitions can be either "local" or "global".

Current Scope

Definitions with a "local" scope in a question can only be used in that question. Definitions with a "local" scope in a reference can be used by that reference and by questions that are associated with that reference.

Global Scope

Definitions whose scope are "global" are defined once and then can be used in any question or reference anywhere in a question bank or exam document they are defined in. Only constants and functions can be declared in a "global" scope.
**Tip:** If a constant or function will be used across multiple questions, it is best to define it with "global" scope.

### Functions

#### Conversion Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asc()</td>
<td>Convert a String Character into an ASCII Value</td>
</tr>
<tr>
<td>Chr()</td>
<td>Convert an ASCII Value into a String Character</td>
</tr>
<tr>
<td>Deg()</td>
<td>Convert Radians into Degrees</td>
</tr>
<tr>
<td>Rad()</td>
<td>Convert Degrees into Radians</td>
</tr>
<tr>
<td>Str()</td>
<td>Convert a Value into a String</td>
</tr>
<tr>
<td>Val()</td>
<td>Convert a String into a Value</td>
</tr>
<tr>
<td>LCase()</td>
<td>Convert a String into Lower Case</td>
</tr>
<tr>
<td>UCase()</td>
<td>Convert a String into Upper Case</td>
</tr>
</tbody>
</table>

#### Logic Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose()</td>
<td>Branching (1-N)</td>
</tr>
<tr>
<td>If()</td>
<td>Branching (True/False)</td>
</tr>
<tr>
<td>Unique()</td>
<td>Returns true if all items are unique</td>
</tr>
</tbody>
</table>

#### Math Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs()</td>
<td>Find the Absolute Value of a Value</td>
</tr>
<tr>
<td>Binomial()</td>
<td>Computes binomial coefficients</td>
</tr>
<tr>
<td>Deg()</td>
<td>Convert Radians into Degrees</td>
</tr>
<tr>
<td>Exp()</td>
<td>Find the Natural Exponent of a Value</td>
</tr>
<tr>
<td>gcf()</td>
<td>Returns the greatest common factor</td>
</tr>
<tr>
<td>Inv()</td>
<td>Determine the Inverse of a Value</td>
</tr>
<tr>
<td>Log()</td>
<td>Find the Natural Logarithm of a Value</td>
</tr>
<tr>
<td>Log10()</td>
<td>Find the Base 10 Logarithm of a Value</td>
</tr>
<tr>
<td>LogB()</td>
<td>Find the (User-Defined) Base Logarithm of a Value</td>
</tr>
<tr>
<td>Max()</td>
<td>Find the Maximum Value in a Sequence</td>
</tr>
<tr>
<td>Min()</td>
<td>Find the Minimum Value in a Sequence</td>
</tr>
<tr>
<td>Prime()</td>
<td>Return the nth prime number</td>
</tr>
<tr>
<td>Rad()</td>
<td>Convert Degrees into Radians</td>
</tr>
<tr>
<td>Reduce()</td>
<td>Reduces a fraction</td>
</tr>
<tr>
<td>Sgn()</td>
<td>Determine the Sign of a Value</td>
</tr>
<tr>
<td>Sqr()</td>
<td>Determine the Square Root of a Value</td>
</tr>
</tbody>
</table>
### Financial Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fv()</td>
<td>Find the future value of an investment</td>
</tr>
<tr>
<td>Pv()</td>
<td>Find the present value of an investment</td>
</tr>
<tr>
<td>Pmt()</td>
<td>Find loan payments for constant payments and interest rates</td>
</tr>
<tr>
<td>Nper()</td>
<td>Find the number of periods for an investment with constant payments and interest rates</td>
</tr>
<tr>
<td>Npval()</td>
<td>Find the net present value of an investment</td>
</tr>
<tr>
<td>Ppmt()</td>
<td>Find the payment on the principal of an investment during a specific period</td>
</tr>
<tr>
<td>Ipmt()</td>
<td>Find interest payments for an investment during a specific period</td>
</tr>
<tr>
<td>Cumipmt()</td>
<td>Find the cumulative interest paid between two specific periods</td>
</tr>
<tr>
<td>Cumprinc()</td>
<td>Find the cumulative principal paid on a loan between two specific periods</td>
</tr>
<tr>
<td>Rate()</td>
<td>Find the interest rate per period of annuity</td>
</tr>
</tbody>
</table>

### Rounding Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ceiling()</td>
<td>Round Up</td>
</tr>
<tr>
<td>Floor()</td>
<td>Round Down</td>
</tr>
<tr>
<td>Int()</td>
<td>Round Towards Zero</td>
</tr>
<tr>
<td>Round()</td>
<td>Round to Nearest Whole Number</td>
</tr>
<tr>
<td>RoundSig()</td>
<td>Round to a significant digit</td>
</tr>
<tr>
<td>SigUnit()</td>
<td>Process a significant digit</td>
</tr>
</tbody>
</table>

### Statistical Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combin()</td>
<td>Determine the Number of Combinations</td>
</tr>
<tr>
<td>erf()</td>
<td>Computes cumulative normal distribution</td>
</tr>
<tr>
<td>Perm()</td>
<td>Determine the Number of Permutations</td>
</tr>
<tr>
<td>Rand()</td>
<td>Generate a Random Value</td>
</tr>
<tr>
<td>StudentSt()</td>
<td>Computes students t-test distribution</td>
</tr>
<tr>
<td>Sum()</td>
<td>Sums the result of an expression across a range</td>
</tr>
</tbody>
</table>

### String Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asc()</td>
<td>Convert a String Character into an ASCII Value</td>
</tr>
<tr>
<td>Chr()</td>
<td>Convert an ASCII Value into a String Character</td>
</tr>
<tr>
<td>Instr()</td>
<td>Find the Location of a String in another String</td>
</tr>
<tr>
<td>LCase()</td>
<td>Convert a String into Lower Case</td>
</tr>
<tr>
<td>Left()</td>
<td>Isolate Characters in the Left Side of a String</td>
</tr>
<tr>
<td>Len()</td>
<td>Find the Length (Number of Characters) of a String</td>
</tr>
<tr>
<td>LTrim()</td>
<td>Eliminate Space Characters from the Left Side of a String</td>
</tr>
<tr>
<td>Mid()</td>
<td>Isolate Characters in the Middle of a String</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>Right( )</strong></td>
<td>Isolate Characters in the Right Side of a String</td>
</tr>
<tr>
<td><strong>RTrim( )</strong></td>
<td>Eliminate Space Characters from the Right Side of a String</td>
</tr>
<tr>
<td><strong>Str( )</strong></td>
<td>Convert a Value into a String</td>
</tr>
<tr>
<td><strong>Trim( )</strong></td>
<td>Eliminate Space Characters from the Sides of a String</td>
</tr>
<tr>
<td><strong>UCase( )</strong></td>
<td>Convert a String into Upper Case</td>
</tr>
<tr>
<td><strong>Val( )</strong></td>
<td>Convert a String into a Value</td>
</tr>
</tbody>
</table>

**Note:** The concatenation operator ("&") can be used to join strings together.

**Trigonometric Functions**

**Standard Trigonometric Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cos( )</strong></td>
<td>Cosine</td>
</tr>
<tr>
<td><strong>Cot( )</strong></td>
<td>Cotangent</td>
</tr>
<tr>
<td><strong>Csc( )</strong></td>
<td>Cosecant</td>
</tr>
<tr>
<td><strong>Sec( )</strong></td>
<td>Secant</td>
</tr>
<tr>
<td><strong>Sin( )</strong></td>
<td>Sine</td>
</tr>
<tr>
<td><strong>Tan( )</strong></td>
<td>Tangent</td>
</tr>
</tbody>
</table>

**Trigonometric Conversion Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deg( )</strong></td>
<td>Convert Radians into Degrees</td>
</tr>
<tr>
<td><strong>Rad( )</strong></td>
<td>Convert Degrees into Radians</td>
</tr>
</tbody>
</table>

**Inverse Trigonometric Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACos( )</strong></td>
<td>Arc Cosine</td>
</tr>
<tr>
<td><strong>ACot( )</strong></td>
<td>Arc Cotangent</td>
</tr>
<tr>
<td><strong>ACsc( )</strong></td>
<td>Arc Cosecant</td>
</tr>
<tr>
<td><strong>ASec( )</strong></td>
<td>Arc Secant</td>
</tr>
<tr>
<td><strong>ASin( )</strong></td>
<td>Arc Sine</td>
</tr>
<tr>
<td><strong>ATan( )</strong></td>
<td>Arc Tangent</td>
</tr>
</tbody>
</table>

**Hyperbolic Trigonometric Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CosH( )</strong></td>
<td>Hyperbolic Cosine</td>
</tr>
<tr>
<td><strong>CotH( )</strong></td>
<td>Hyperbolic Cotangent</td>
</tr>
<tr>
<td><strong>CscH( )</strong></td>
<td>Hyperbolic Cosecant</td>
</tr>
<tr>
<td><strong>SecH( )</strong></td>
<td>Hyperbolic Secant</td>
</tr>
<tr>
<td><strong>SinH( )</strong></td>
<td>Hyperbolic Sine</td>
</tr>
<tr>
<td><strong>TanH( )</strong></td>
<td>Hyperbolic Tangent</td>
</tr>
</tbody>
</table>

**Inverse Hyperbolic Trigonometric Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACosH( )</strong></td>
<td>Arc Hyperbolic Cosine</td>
</tr>
<tr>
<td><strong>ACotH( )</strong></td>
<td>Arc Hyperbolic Cotangent</td>
</tr>
<tr>
<td><strong>ACscH( )</strong></td>
<td>Arc Hyperbolic Cosecant</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------</td>
</tr>
<tr>
<td>ASecH()</td>
<td>Arc Hyperbolic Secant</td>
</tr>
<tr>
<td>ASinh()</td>
<td>Arc Hyperbolic Sine</td>
</tr>
<tr>
<td>ATanH()</td>
<td>Arc Hyperbolic Tangent</td>
</tr>
</tbody>
</table>
Reference

Abs()  
The Abs() function returns the absolute value of the expression that is passed to it.

Form:
Abs( Expression )

Samples:
X = Abs( AltitudeChange )
X = Abs( Car1Price - Car2Price )

See Also:
Sgn()  
Math Functions

ACos()  
The ACos() function returns the angle whose cosine was passed to the function. The value returned by the ACos() function is expressed in radians. Use the Deg() function to convert radians to degrees when needed.

Form:
ACos( Expression )

Samples:
X = ACos( Y )
X = Deg( ACos( .5 ) )

See Also:
Cos()  
Deg()  
Trigonometric Functions
ACosH()

The ACosH() function returns the angle whose hyperbolic cosine was passed to the function. The value returned by the ACosH() function is expressed in radians. Use the Deg() function to convert radians to degrees when needed.

Form:
ACosH( Expression )

Samples:
X = ACosH( Y )
X = Deg( ACosH( .5 ) )

See Also:
CosH()
Deg()
Trigonometric Functions

ACot()

The ACot() function returns the angle whose cotangent was passed to the function. The value returned by the ACot() function is expressed in radians. Use the Deg() function to convert radians to degrees when needed.

Form:
ACot( Expression )

Samples:
X = ACot( Y )
X = Deg( ACot( .5 ) )

See Also:
Cot()
Deg()
Trigonometric Functions

ACotH()

The ACotH() function returns the angle whose hyperbolic cotangent was passed to the function. The value returned by the ACotH() function is expressed in radians. Use the Deg() function to convert radians to degrees when needed.
**Form:**
\[ \text{ACotH( Expression )} \]

**Samples:**
\[ X = \text{ACotH}( Y ) \]
\[ X = \text{Deg}( \text{ACotH}( .5 ) ) \]

**See Also:**
- \text{CotH( )}
- \text{Deg( )}
- \text{Trigonometric Functions}

---

**ACsc( )**

The ACsc( ) function returns the angle whose cosecant was passed to the function. The value returned by the ACsc( ) function is expressed in radians. Use the Deg( ) function to convert radians to degrees when needed.

**Form:**
\[ \text{ACsc( Expression )} \]

**Samples:**
\[ X = \text{ACsc}( Y ) \]
\[ X = \text{Deg}( \text{ACsc}( .5 ) ) \]

**See Also:**
- \text{Csc( )}
- \text{Deg( )}
- \text{Trigonometric Functions}

---

**ACscH( )**

The ACscH( ) function returns the angle whose hyperbolic cosecant was passed to the function. The value returned by the ACscH( ) function is expressed in radians. Use the Deg( ) function to convert radians to degrees when needed.

**Form:**
\[ \text{ACscH( Expression )} \]
**ASC**

The **ASC** function returns the ASCII value of a string character. For example, the return value of ASC("A") is 65, the return value of ASC("B") is 66, and so forth. If a string is passed to the ASC() function instead of a single character, the ASCII value of the first character is returned.

**Form:**

ASC( Character )

**Samples:**

X = ASC( StudentName )
X = ASC( Address1 + Address2 )

**See Also:**

- **Chr()**
- **Conversion Functions**
- **String Functions**

**ASec()**

The **ASec()** function returns the angle whose secant was passed to the function. The value returned by the **ASec()** function is expressed in radians. Use the **Deg()** function to convert radians to degrees when needed.

**Form:**

ASec( Expression )

**Samples:**

X = ASec( Y )
X = Deg( ASec(.5) )
The ASecH() function returns the angle whose hyperbolic secant was passed to the function. The value returned by the ASecH() function is expressed in radians. Use the Deg() function to convert radians to degrees when needed.

Form:
ASecH( Expression )

Samples:
X = ASecH( Y )
X = Deg( ASecH( .5 ) )

The ASin() function returns the angle whose sine was passed to the function. The value returned by the ASin() function is expressed in radians. Use the Deg() function to convert radians to degrees when needed.

Form:
ASin( Expression )

Samples:
X = ASin( Y )
X = Deg( ASin( .5 ) )
ASinH( )

The ASinH( ) function returns the angle whose hyperbolic sine was passed to the function. The value returned by the ASinH( ) function is expressed in radians. Use the Deg( ) function to convert radians to degrees when needed.

Form:
ASinH( Expression )

Samples:
X = ASinH( Y )
X = Deg( ASinH( .5 ) )

See Also:
SinH()
Deg()
Trigonometric Functions

ATan( )

The ATan( ) function returns the angle whose tangent was passed to the function. The value returned by the ATan( ) function is expressed in radians. Use the Deg( ) function to convert radians to degrees when needed.

Form:
ATan( Expression )

Samples:
X = ATan( Y )
X = Deg( ATan(.5) )

See Also:
Tan()
Deg()
Trigonometric Functions

ATanH( )

The ATanH( ) function returns the angle whose hyperbolic tangent was passed to the function. The value returned by the ATanH( ) function is expressed in radians. Use the Deg( ) function to convert radians to degrees when needed.
**Form:**
ATanH( Expression )

**Samples:**
X = ATanH( Y )
X = Deg( ATanH( .5 ) )

**See Also:**
TanH()
Deg()
Trigonometric Functions

---

**Binomial( )**

The Binomial( ) function computes the binomial coefficient. This function returns the n\textsuperscript{th} coefficient of (x+y)^m.

**Form:**
Binomial( m, n )

**Samples:**
X = binomial(8, 3)

**See Also:**
Math Functions

---

**Choose( )**

The Choose( ) function requires at least two arguments: an integer expression, and a comma-delimited list of values. The integer expression must have a value between 1 and N (where N is the number of values in the comma-delimited list). The function returns the value from the list specified by the integer expression.

Choose( ) is a special function in that it does not evaluate all parameters of the function when evaluating the function. This allows you to perform special processing that would cause errors in some cases.

**Form:**
Choose( Expression, Item1, Item2, Item3, ..., ItemN )

**Samples:**
X = Choose( Weekday, "Sun", "Mon", "Tues", "Wed", "Thurs", "Fri", "Sat"
X = Choose( Rand(1, 5), 10, 20, 30, 40, 50 )
Ceiling()

The Ceiling() function rounds non-integer expressions to the next highest integer. Rounding occurs in the direction of the positive infinity: Ceiling(3.5) yields 4, and Ceiling(-3.5) yields -3.

**Form:**
```
Int( Expression )
```

**Samples:**
```
X = Ceiling( CarPayment )
X = Ceiling( MonthlyCost * 12 )
```

Chr()

The Chr() function is the inverse of the Asc() function. The Chr() function returns the string character whose ASCII value was passed to it. For example, the return value of Chr(65) is "A", the return value of Chr(66) is "B", and so forth. Only integers having a value from 0 to 255 may be passed to the Chr() function.

**Form:**
```
Chr( Expression )
```

**Samples:**
```
X = Chr( 32 )
X = Chr( Y + 64 )
```
Combin( )

The Combin( X, Y ) function returns the number of unique subsets containing exactly Y elements that can be created from a set of X elements.

Form:
Combin( AvailableElements, SubsetSize )

Samples:
X = Combin( StudentBody, ClassSize )
X = Combin( PhoneListings, 2 )

See Also:
Perm()
Statistical Functions

Cos( )

The Cos( ) function returns the cosine of an expression. The value that is passed to the Cos( ) function must be expressed in radians. Use the Rad() function to convert degrees to radians when needed.

Form:
Cos( Expression )

Samples:
X = Cos( Y )
X = Cos( Rad(30) )

See Also:
Sin()
Rad()
Trigonometric Functions
**CosH()**

The CosH() function returns the hyperbolic cosine of an expression. The value that is passed to the CosH() function must be expressed in radians. Use the Rad() function to convert degrees to radians when needed.

**Form:**
CosH( Expression )

**Samples:**
X = CosH( Y )
X = CosH( Rad(30) )

**See Also:**
- SinH()
- Rad()
- Trigonometric Functions

**Cot()**

The Cot() function returns the cotangent of an expression. The value that is passed to the Cot() function must be expressed in radians. Use the Rad() function to convert degrees to radians when needed.

**Form:**
Cot( Expression )

**Samples:**
X = Cot( Y )
X = Cot( Rad(30) )

**See Also:**
- Sec()
- Rad()
- Trigonometric Functions

**CotH()**

The CotH() function returns the hyperbolic cotangent of an expression. The value that is passed to the CotH() function must be expressed in radians. Use the Rad() function to convert degrees to radians when needed.
Form:
CotH( Expression )

Samples:
X = CotH(Y)
X = CotH(Rad(30))

See Also:
SecH()
Rad()  
Trigonometric Functions

Csc()

The Csc() function returns the cosecant of an expression. The value that is passed to the Csc() function must be expressed in radians. Use the Rad() function to convert degrees to radians when needed.

Form:
Csc( Expression )

Samples:
X = Csc(Y)
X = Csc(Rad(30))

See Also:
TanH()
Rad()  
Trigonometric Functions

CscH()

The CscH() function returns the hyperbolic cosecant of an expression. The value that is passed to the CscH() function must be expressed in radians. Use the Rad() function to convert degrees to radians when needed.

Form:
CscH( Expression )

Samples:
X = CscH(Y)
X = CscH(Rad(30))
Cumipmt(_)

The Cumipmt() function returns the cumulative interest paid on a load between two periods.

**Form:**
cumipmt(rate, number of payments, pv, start period, end period, type)

**Note:**
Type is an optional parameter. If no value is specified, the default value is set to 0.

**Sample:**
cumipmt(0.08/52, 4*52, 3000, 1, 12, 1)

See Also: Financial Functions

Cumprinc(_)

The Cumprinc() function returns the cumulative principal paid on a loan between two loan periods.

**Form:**
cumprinc(rate, number of payments, pv, start period, end period, type)

**Note:**
Type is an optional parameter. If no value is specified, the default value is set to 0.

**Sample:**
cumprinc(0.08/52, 4*52, 3000, 3, 12, 0)

See Also: Financial Functions
**Deg( )**

The Deg( ) function converts radians to degrees.

**Form:**
Deg(Radians)

**Samples:**
X = Deg(AngleY)

**See Also:**
Rad()
Pre-Defined Constants
Conversion Functions
Math Functions
Trigonometric Functions

**erf( )**

The erf( ) function computes the cumulative distribution of a standard normal distribution.

**Form:**
erf(x)

**Samples:**
X = erf(0)
X = erf(5)

**See Also:**
StudentSt()
Statistical Functions

**Exp( )**

The Exp( ) function returns the natural exponent of an expression.

**Form:**
Exp(Expression)
Samples:
X = Exp( Diameter )
X = Exp( 2 * Radius )

See Also:
Log()
Pre-Defined Constants
Math Functions

Exponent()

The Exponent( ) function returns the exponent of a floating-point number (base 10). This is useful when you need to create an alternate representation of scientific notation.

Form:
Exponent( Expression )

Samples:
Exponent( 100000000 ) would give a result of 8
Exponent( 3.45e-10 ) would give a result of -10

See Also:
Mantissa()

Floor()

The Floor( ) function rounds non-integer expressions to the next lowest integer. Rounding occurs in the direction of the negative infinity: Floor(3.5) yields 3, and Floor(-3.5) yields -4.

Form:
Floor( Expression )

Samples:
X = Floor( Weight )
X = Floor( ProjectDuration / 7 )
**Fv( )**

The Fv( ) function returns the future value of an investment based on a given interest rate and a constant payment schedule.

**Form:**
fv(rate, number of payments, payment, pv, type)

**Note:**
pv and type are optional parameters. If no value is specified, the default value for both is set to 0.

**Sample:**
fv(0.08/52, 2*12, -300, -8000, 1)

See Also: Financial Functions

**gcf( )**

The gcf( ) function returns the greatest common factor of two numbers.

**Form:**
gcf(a, b)

**Samples:**
X = gcf(10, 20)
X = gcf(11, 13)

See Also: Reduce(), Prime(), Math Functions
**If()**

The If() function requires three arguments: an expression to evaluate, a value to return if the expression is true (non-zero), and a value to return if the expression is false (zero).

If() is a special function in that it does not evaluate all parameters of the function when evaluating the function. This allows you to perform special processing that would cause errors in some cases.

For example:

```plaintext
X = If (a <> 0, 1/a, 0)
```

When a is zero, then second expression is not evaluated and the entire statement will not cause an error.

**Form:**

`If ( Expression, True Value, False Value)`

**Samples:**

```plaintext
X = If (Gender="Male", "Men", "Women")
X = If (N, 100, 10)
X = If (Y>Z, Y*3, Z*3)
```

**See Also:**

- Relational operators
- Logic Functions

**Instr()**

The Instr() function returns the location of the first occurrence of a search string in a target string. For example, the return value of Instr("Denver Colorado", "Color") is 8. Zero is returned when the search string can't be found in the target string.

**Form:**

`Instr( TargetString, SearchString )`

**Samples:**

```plaintext
X = Instr( StudentFullName, StudentLastName )
X = Instr( BookTitle, "by" )
```

**See Also:**

- String Functions
**Int()**

The Int() function returns the integer portion of an expression. Rounding occurs in the direction of the origin: Int(3.5) yields 3, and Int(-3.5) yields -3.

**Form:**
Int( Expression )

**Samples:**
X = Int( GroceryBill )
X = Int( AgeInDays / 365 )

**See Also:**
- Round()
- Ceiling()
- Floor()
- Rounding Functions

**Inv()**

The Inv() function returns the inverse of an expression. Inv(x) is the same as 1/x.

**Form:**
Inv( Expression )

**Samples:**
X = Inv( GallonsPerMile )
X = Inv( MinutesPerMile / 60 )

**See Also:**
- Math Functions

**Ipmt()**

The Ipmt() function returns the interest payment for an investment based on a given interest rate and a constant payment schedule.

**Form:**
ipmt(rate, period, number of payments, pv, fv, type)
**Note:**
fv and type are optional parameters. If no value is specified, the default value for both is set to 0.

**Sample:**
\[ \text{ipmt}(0.08/52, 30, 4*52, 8000, 0, 1) \]

**See Also:** Financial Functions

---

**LCase()**

The LCase() function returns a lower-case version of a string. For example, the return value of LCase("50 Main Street") is "50 main street".

**Form:**
\[ \text{LCase}( \text{String} ) \]

**Samples:**
\[ X = \text{LCase}( \text{StudentName} ) \]
\[ X = \text{LCase}( \text{Address1} + \text{Address2} ) \]

**See Also:**
- UCase()
- Conversion Functions
- String Functions

---

**Left()**

The Left() function returns the specified number of characters from the left side of a string. For example, the return value of Left("Robert Browne", 6) is "Robert".

**Form:**
\[ \text{Left}( \text{String}, \text{Length} ) \]

**Samples:**
\[ X = \text{Left}( \text{"500 Fifth Avenue"}, 3) \]
\[ X = \text{Left}( \text{FullName}, \text{FirstNameLength} ) \]
Len( )

The Len( ) function returns the length of a specified string. For example, the return value of Len("Denver") is 6.

Form:
Len( String )

Samples:
X = Len( StudentName )
X = Len( BookTitle )

See Also:
String Functions

Log( )

The Log( ) function returns the natural logarithm of an expression. The function can also be expressed: Ln().

Form:
Log( Expression )

Samples:
X = Log( Diameter )
X = Log( 2 * Radius )

See Also:
Exp( )
LogB( )
Log10( )
Math Functions
**Log10( )**

The Log10( ) function returns the base 10 logarithm of an expression.

**Form:**

Log10( Expression)

**Samples:**

\[ X = \text{Log10( Diameter )} \]
\[ X = \text{Log10( 2 * Radius )} \]

**See Also:**

Log( ), LogB( ), Math Functions

**LogB( ), LogN()**

The LogB( ) function returns the base logarithm of an expression where the base is specified by the user. This function can also be expressed as LogN( ).

**Form:**

LogB( Expression, Base )

**Samples:**

\[ X = \text{LogB( Diameter, 2 )} \]
\[ X = \text{LogB( 2 * Radius, 10 )} \]

**See Also:**

Log( ), Log10( ), Math Functions

**LTrim( )**

The LTrim( ) function removes the leading space characters from a string. For example, the return value of LTrim("500 Main Street   ") is "500 Main Street   ".

**Form:**

LTrim( String )
Samples:
X = LTrim( Address1 )
X = LTrim( Address1 + Address2 )

See Also:
Trim()
RTrim()
String Functions

Mantissa()

The Mantissa() function returns the mantissa of a floating-point number (base 10). This is useful when you need to create an alternate representation of scientific notation.

Form:
Mantissa( Expression )

Samples:
Mantissa( 100000000 ) would give a result of 1
Mantissa( 3.45e-10 ) would give a result of 3.45

See Also:
Exponent()

Max()

The Max( ) function returns the maximum value from a comma-delimited list of expressions.

Form:
Max( Expression1, Expression2, Expression3, ... , ExpressionN )

Samples:
X = Max( Class1Average, Class2Average )
X = Max( City1Population, City2Population, City3Population )

See Also:
Min()
Math Functions
Mid( )

The Mid( ) function returns a portion of a string. Three arguments are passed to the function: the initial string, the location in the initial string (number of characters) where the desired portion starts, and the length of the desired portion (also in number of characters). For example, the return value of Mid("300 Main Street", 5, 4) is "Main".

Form:
Mid( String, Starting Location, Length )

Samples:
X = Mid( Address, X, L )
X = Mid( "300 Main Street, Townsville, Texas", CityPosition, CityLength )

See Also:
Left( )
Right( )
String Functions

Min( )

The Min( ) function returns the minimum value from a comma-delimited list of expressions.

Form:
Min( Expression1, Expression2, Expression3, ... , ExpressionN )

Samples:
X = Min( Car1Speed, Car2Speed, Car3Speed )
X = Min( Test1Score, Test2Score, Test3Score, MidtermScore )

See Also:
Max( )
Math Functions

Nper( )

The Nper( ) function returns the number of periods for an investment with a given interest rate and a constant payment schedule.

Form:
nper(rate, payment, pv, fv, type)
**Note:**
fv and type are optional parameters. If no value is specified, the default value for both is set to 0.

**Sample:**
\[ nper(0.08/52, -300, -8000, 0, 1) \]

**See Also:** Financial Functions

---

**Npv( )**

The Npv( ) function returns the net present value of an investment based on a series of future payments and income.

**Form:**
\[ \text{npv(discount rate, value1, value2, ..., value_n)} \]

**Sample:**
\[ \text{npv(0.05, 5000, 3000, 4000, 1200)} \]

**See Also:** Financial Functions

---

**Perm( )**

The Perm( X, Y ) function returns the number of unique arrangements of exactly Y elements that can be created from a set of X elements.

**Form:**
\[ \text{Perm(AvailableElements, ArrangementSize)} \]

**Samples:**
\[ X = \text{Perm(StudentBody, ClassSize)} \]
\[ X = \text{Perm(PhoneListings, 2)} \]

**See Also:**
- Combin( )
- Statistical Functions
**Pmt( )**

The `Pmt( )` function returns the payment amount for a loan with a given interest rate and a constant payment schedule.

**Form:**

\[ pmt(rate, \text{number of payments}, pv, fv, type) \]

**Note:**

fv and type are optional parameters. If no value is specified, the default value for both is set to 0.

**Sample:**

\[ pmt(0.08/52, 2*12, -8000, 0, 1) \]

**See Also:** Financial Functions

**Ppmt( )**

The `Ppmt( )` function returns the payment on the principal based on a given interest rate and a constant payment schedule.

**Form:**

\[ ppmt(rate, period, \text{number of payments}, pv, fv, type) \]

**Note:**

fv and type are optional parameters. If no value is specified, the default value for both is set to 0.

**Sample:**

\[ ppmt(0.08/52, 20, 2*52, 8000, 0, 1) \]

**See Also:** Financial Functions

**Prime( )**

The `Prime( )` function returns the nth prime number (up to n=1028).

**Form:**

\[ Prime(n) \]
Samples:
Prime(1) yields 2
Prime(5) yields 11
Prime(10) yields 29

See Also:
gcf()  
Reduce()  
Math Functions

Pv()

The Pv() function returns the present value of an investment based on a given interest rate and a constant payment schedule.

Form:
\[ pv(\text{rate}, \text{number of payments}, \text{payment}, \text{fv}, \text{type}) \]

Note:
fv and type are optional parameters. If no value is specified, the default value for both is set to 0.

Sample:
\[ pv(0.08/52, 2*12, -300, -8000, 1) \]

See Also: Financial Functions

Rad()

The Rad() function converts degrees to radians.

Form:
\[ \text{Rad( Degrees )} \]

Samples:
\[ X = \text{Rad( AngleY )} \]
Rand()

The Rand() function returns an integer that is randomly selected from a range of integers specified by a low expression and a high expression.

Form:
Rand( Low expression, High expression)

Samples:
X = Rand( FirstRaffleTicketNumber, LastRaffleTicketNumber )
X = Rand( 1, 365 )

Rate(_)

The Rate() function returns the interest rate for a period of annuity.

Form:
rate(number of payments, payment, pv, fv, type, estimate)

Note:
Type and estimate are optional parameters. If no value is specified, the default values are set to 0 and 0.10, respectively.

Sample:
rate(2*12, -300, 8000, 0, 1, )

See Also: Financial Functions
**Reduce( )**

The `Reduce( )` function aids in reducing a fraction for display.

**Form:**
`Reduce( Part, Numerator, Denominator )`
If Part = 1, then the Numerator is returned. If Part = 2, then the Denominator is returned.

**Samples:**
Numerator = `Reduce(1, 24, 96)`
Denominator = `Reduce(2, 24, 96)`

**See Also:**
- `gcf( )`
- `Prime( )`
- `Math Functions`

**Right( )**

The `Right( )` function returns the specified number of characters from the right side of a string. For example, the return value of `Right("Robert Browne", 6)` is "Browne".

**Form:**
`Right( String, Length )`

**Samples:**
X = `Right("500 Fifth Avenue", 12 )`
X = `Right( FullName, LastNameLength )`

**See Also:**
- `Mid( )`
- `Left( )`
- `String Functions`

**Round( )**

The `Round( )` function rounds an expression to the nearest multiple. If no multiple is provided, 1 is assumed.

**Form:**
`Round( Expression, Multiple )`
**Samples:**

\[ X = \text{Round}(\text{Mileage}, 1000) \]
\[ X = \text{Round}(\text{StockPrice}, .125) \]

**See Also:**
- Int()
- Ceiling()
- Floor()
- Rounding Functions

---

### RoundSig()

The `RoundSig()` function rounds an expression to a specified significant digit.

**Form:**

\[ \text{RoundSig}(\text{Expression, SignificantDigit}) \]

**Samples:**

\[ \text{RoundSig}(12.3456, 4) \text{ yields } 12.35 \]

**See Also:**
- SigUnit()
- Round()
- Rounding Functions

---

### RTrim()

The `RTrim()` function removes the trailing space characters from a string. For example, the return value of `RTrim("500 Main Street   ")` is "500 Main Street".

**Form:**

\[ \text{RTrim}(\text{String}) \]

**Samples:**

\[ X = \text{RTrim} (\text{Address1}) \]
\[ X = \text{RTrim}(\text{Address1} + \text{Address2}) \]
Sec( )

The Sec( ) function returns the secant of an expression. The value that is passed to the Sec( ) function must be expressed in radians. Use the Rad( ) function to convert degrees to radians when needed.

Form:
Sec( Expression )

Samples:
X = Sec( Y )
X = Sec( Rad(30) )

SecH( )

The SecH( ) function returns the hyperbolic secant of an expression. The value that is passed to the SecH( ) function must be expressed in radians. Use the Rad( ) function to convert degrees to radians when needed.

Form:
SecH( Expression )

Samples:
X = SecH( Y )
X = SecH( Rad(30) )
\textbf{Sgn()}

The \texttt{Sgn()} function returns one of three possible values: 1 if the expression that is passed to it is positive, -1 if the expression that is passed to it is negative, and 0 if the expression that is passed to it is zero.

\textbf{Form:}
\texttt{Sgn( Expression )}

\textbf{Samples:}
\texttt{X = Sgn(YearEndProfitOrLoss)}
\texttt{X = Sgn(A * B)}

\textbf{See Also:}
\begin{itemize}
  \item \texttt{Abs()}
  \item Math Functions
\end{itemize}

\textbf{SigUnit()}

The \texttt{SigUnit()} returns a 1 in the place of an expression's significant digit.

\textbf{Form:}
\texttt{SigUnit( Expression, SignificantDigit )}

\textbf{Samples:}
\texttt{SigUnit(12.3456, 4) yields 0.01}

\textbf{See Also:}
\begin{itemize}
  \item \texttt{RoundSig()}
  \item Rounding Functions
\end{itemize}

\textbf{Sin()}

The \texttt{Sin()} function returns the sine of an expression. The value that is passed to the \texttt{Sin()} function must be expressed in radians. Use the \texttt{Rad()} function to convert degrees to radians when needed.

\textbf{Form:}
\texttt{Sin( Expression )}
**Samples:**

\[ X = \sin(Y) \]
\[ X = \sin(\text{Rad}(30)) \]

**See Also:**

- `Cos()`
- `Rad()`
- [Trigonometric Functions](#)

---

**SinH( )**

The `SinH( )` function returns the hyperbolic sine of an expression. The value that is passed to the `SinH( )` function must be expressed in radians. Use the `Rad( )` function to convert degrees to radians when needed.

**Form:**

\[ \text{SinH( Expression )} \]

**Samples:**

\[ X = \text{SinH}(Y) \]
\[ X = \text{SinH}(\text{Rad}(30)) \]

**See Also:**

- `CosH()`
- `Rad()`
- [Trigonometric Functions](#)

---

**Sqr( )**

The `Sqr( )` function returns the square root of an expression. This function can also be expressed as `Sqrt()`.

**Form:**

\[ \text{Sqr( Expression )} \]

**Samples:**

\[ X = \text{Sqr}(\text{Acreage}) \]
\[ X = \text{Sqr}(\text{Hypotenuse}) \]

**See Also:**

- [Math Functions](#)
**Str()**

The Str() function converts numbers into string expressions. For example, the return value of Str(-150) is "-150". The Str() function has additional features that can be used to control the appearance of the resulting string. The Str() function is the inverse of the Val() function.

**Form:**
Str( Number )

**Sample:**
StreetAddress = Str( HouseNumber ) & " Main Street"

**See Also:**
Val()  
Conversion Functions  
String Functions

**Additional Features:**
In addition to the numeric expression that is passed to the Str() function, the function can accept optional expressions that control the appearance of the resulting string.

**Expanded Form:**
Str( Number, Decimals, Scientific Notation, Commas, Leading Zero )

**Samples:**

<table>
<thead>
<tr>
<th>Expression</th>
<th>Yields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Str( 0.150, 2)</td>
<td>0.15</td>
</tr>
<tr>
<td>Str( 15000, 0, 1)</td>
<td>1.5e4</td>
</tr>
<tr>
<td>Str( 15000, 2, 0, 1)</td>
<td>15,000.00</td>
</tr>
<tr>
<td>Str( 0.1500, 4, 0, 0, 1)</td>
<td>.1500</td>
</tr>
</tbody>
</table>

- The first expression is the number to be converted into a string.
- The second expression controls the number of decimal places to incorporate—as few as zero, as many as twenty. (If the second expression has a value of -1, Diploma 6 decides whether or not to use decimal places based upon whether the numeric expression is an integer.)
- The third expression controls whether or not the string is rendered in scientific notation. A value of 1 means use scientific notation. A value of 0 means don't use scientific notation.
- The forth expression controls whether commas are included in the resulting string to demarcate thousands. A value of 1 means use commas. A value of 0 means don't use commas.
- The fifth expression controls whether a leading zero appears in front of the decimal place when the numeric expression has a value between negative one and positive one. A value of 1 means remove the leading zero. A value of 0 means leave the leading zero in place.
**StudentSt()**

The StudentSt() function computes the cumulative distribution of the Student t-distribution on n degrees of freedom.

**Form:**

StudentSt( n, x )

**Samples:**

X = StudentSt(3, 2)
X = StudentSt(2, 1.5)

**See Also:**

erf()

Statistical Functions

---

**Sum()**

The Sum() function calculates an expression multiple times along an interval, returning the sum of the expressions.

**Form:**

Sum( i, m, n, expr )

**Samples:**

X = Sum(i, 4, 20, i^2)

**See Also:**

Statistical Functions

---

**Tan()**

The Tan() function returns the tangent of an expression. The value that is passed to the Tan() function must be expressed in radians. Use the Rad() function to convert degrees to radians when needed.

**Form:**

Tan( Expression )

**Samples:**

X = Tan( Y )
X = Tan( Rad(30) )
**TanH( )**

The TanH( ) function returns the hyperbolic tangent of an expression. The value that is passed to the TanH( ) function must be expressed in radians. Use the Rad() function to convert degrees to radians when needed.

**Form:**
TanH( Expression )

**Samples:**
X = TanH( Y )
X = TanH( Rad(30) )

See Also:
CscH()
Rad()
Trigonometric Functions

---

**Trim( )**

The Trim( ) function removes the leading and trailing space characters from a string. For example, the return value of Trim("  50 Main Street  ") is "50 Main Street".

**Form:**
Trim( String )

**Samples:**
X = Trim( BookTitle )
X = Trim( Address1 + Address2 )

See Also:
LTrim()
RTrim()
String Functions
**UCase( )**

The UCase( ) function returns an upper-case version of a string. For example, the return value of UCase("50 Main Street") is "50 MAIN STREET".

**Form:**
UCase( String )

**Samples:**
X = UCase( StudentName )
X = UCase( Address1 + Address2 )

**See Also:**
LCase()  
Conversion Functions
String Functions

**Unique( )**

The Unique( ) function is used to determine if each member in a series of expressions is unique. The expressions can be numbers or strings. If the each expression is unique, the function returns a value of positive one (1). If two or more of the expressions are identical, the function returns a value of zero (0).

**Form:**
Unique( Expression1, Expression2, ... ExpressionN )

**Samples:**
X = Unique( A, B, C, D, E )
X = Unique( Choice1, Choice2, Choice3 )

**See Also:**
Conversion Functions

**Val( )**

The Val( ) function returns the numeric value of a string expression. For example, the return value of Val("75 degrees") is 75. The Val( ) function is the inverse of the Str( ) function.

**Form:**
Val( Expression )
Samples:
X = Val( StreetAddress )
X = Val( "500 Main Street" )

See Also:
Str()
Conversion Functions
String Functions

Operators

Arithmetic Operators

Arithmetic operators perform mathematical operations on numeric expressions. There are seven levels of precedence for arithmetic operators:

<table>
<thead>
<tr>
<th>Precedence</th>
<th>Operation</th>
<th>Operator</th>
<th>Expression</th>
<th>Yields</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Factorial</td>
<td>!</td>
<td>7!</td>
<td>5040</td>
</tr>
<tr>
<td>2</td>
<td>Exponentiation</td>
<td>^</td>
<td>7^2</td>
<td>49</td>
</tr>
<tr>
<td>3</td>
<td>Negation</td>
<td>-</td>
<td>-7</td>
<td>-7</td>
</tr>
<tr>
<td>4</td>
<td>Multiplication</td>
<td>*</td>
<td>7*2</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Division</td>
<td>/</td>
<td>7/2</td>
<td>3.5</td>
</tr>
<tr>
<td>5</td>
<td>Integer Division</td>
<td>\</td>
<td>7\2</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>Modulo Arithmetic</td>
<td>MOD</td>
<td>7 MOD 5</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Addition</td>
<td>+</td>
<td>7+2</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Subtraction</td>
<td>-</td>
<td>7-2</td>
<td>5</td>
</tr>
</tbody>
</table>

Note: The percent sign ("%") can be instead of "MOD" in modulo arithmetic operations.

Relational Operators

Relational operators test expressions by comparing them. If the test is true, the value returned by the comparison is a positive one (1). If the test is false, the value returned is zero (0).

There are two levels of precedence for relational operators. All relational operators have a lower precedent than arithmetic operators.

<table>
<thead>
<tr>
<th>Precedence</th>
<th>Relational Test</th>
<th>Operator</th>
<th>Expression</th>
<th>Yields</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Less Than</td>
<td>&lt;</td>
<td>7 &lt; 2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Greater Than</td>
<td>&gt;</td>
<td>7 &gt; 2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Less Than or Equal To</td>
<td>&lt;=</td>
<td>7 &lt;= 2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Greater Than or Equal To</td>
<td>&gt;=</td>
<td>7 &gt;= 2</td>
<td>1</td>
</tr>
</tbody>
</table>
2 | Equality | = | 7 = 2 | 0
   | Inequality | <> | 7 <> 2 | 1

**Note:** Relational operators can also be used to compare string expressions (text) with other strings. In such cases, "A" is less than "B", "B" is less than "C", and so forth. Also, all upper-case letters are less than all lower-case letters: "Z" is less than "a", "a" is less than "b", etc.

**Boolean Logic Operators**

To support logical operations, Boolean operators -- those that resolve to a value of either 1 or 0 -- are featured in Diploma 6's algorithm-solving engine. The Boolean operators are evaluated in the following order of precedence:

<table>
<thead>
<tr>
<th>Precedence</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NOT</td>
</tr>
<tr>
<td>2</td>
<td>AND</td>
</tr>
<tr>
<td>3</td>
<td>OR</td>
</tr>
<tr>
<td>4</td>
<td>XOR</td>
</tr>
<tr>
<td>5</td>
<td>EQV</td>
</tr>
<tr>
<td>6</td>
<td>IMP</td>
</tr>
</tbody>
</table>

**Note:** Because the NOT operator works with only one value at a time (like many functions), it has the same level of precedence as the factorial operator which is evaluated immediately after functions.

**Reference**

**AND Operator**

The following table outlines the values that is returned when the AND operator is used. (In the table, "N" represents any value other than zero.)

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X AND Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example:**

X = Bonus * (AnnualSales > Target AND DaysAbsent < 5)
Concatenation Operator

The concatenation operator (the "&" symbol) is used to join strings together.

Example:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>yield</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Impor&quot; &amp; &quot;tant&quot;</td>
<td></td>
<td></td>
<td>&quot;Important&quot;</td>
</tr>
<tr>
<td>&quot;Mary&quot; &amp; &quot; &quot; &amp; &quot;Hatch&quot;</td>
<td></td>
<td></td>
<td>&quot;Mary Hatch&quot;</td>
</tr>
</tbody>
</table>

EQV Operator

The following table outlines the values that are returned when the EQV operator is used. (In the table, "N" represents any value other than zero.)

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X EQV Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Example:

\[ X = \text{TravelBonus} \times (\text{AirTravelReserved} > 0 \ \text{EQV} \ \text{AirMilesTraveled} > 0) \]

IMP Operator

The following table outlines the values that are returned when the IMP operator is used. (In the table, "N" represents any value other than zero.)

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X IMP Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Example:

\[ X = \text{Bonus} \times (\text{PromisesMade} \ \text{IMP} \ \text{PromisesKept}) \]
NOT Operator

The following table outlines the values that are returned when the NOT operator is used. (In the table, "N" represents any value other than zero.)

<table>
<thead>
<tr>
<th>X</th>
<th>NOT X</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Example:

X = Bonus * (NOT NewEmployee)

OR Operator

The following table outlines the values that are returned when the OR operator is used. (In the table, "N" represents any value other than zero.)

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X AND Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Example:

X = Bonus * (AnnualSales > Target OR DaysAbsent = 0)

XOR Operator

The following table outlines the values that are returned when the XOR operator is used. (In the table, "N" represents any value other than zero.)

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X XOR Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Example:

X = SmallBonus * (Region = TopSalesRegion XOR Line = TopProductLine)
Additional Resources

TheTestingCenter.Com

TheTestingCenter.Com is a site where tests generated by Diploma 6 can be administered online.

Many question banks come with a license for TheTestingCenter.Com. You may also contact Wimba to obtain an end-user Wimba Diploma 6 license that includes access to TheTestingCenter.Com.

For more information on using TheTestingCenter, consult the Support Page for Diploma 6 at TheTestingCenter.Com
Software Updates

To update Diploma 6, simply select **Check for Updates...** from the Help menu. This command contacts Wimba’s web site and checks for a more recent version of Diploma.

Alternatively, you can go to the [download page](http://www.wimba.com/diplomasupport/) to download the latest updates, bug patches, and add-ons for Diploma 6.

Technical Support

If you need further assistance with Wimba Diploma 6 select **Tech Support** from the Help menu to access the Tech Support dialog. The dialog contains the most recent technical support contact information.

When contacting tech support, the information displayed in the Tech Support window may be requested. Click the **Copy to Clipboard** button, and then paste the information into an email. This information helps Wimba to troubleshoot your problem.

Alternately, you can contact Wimba Technical Support for assistance using the information below:

- diplomasupport@wimba.com
- 866.847.8771 (US & Canada)
- 703.956.3813 (Other)
Product Feedback

Wimba welcomes your suggestions for improving our products. If you have an idea for a new feature or enhancement, please click the link below:

Wimba Product Feedback Form

You may also email your suggestions to feedback@wimba.com

Your suggestion will be sent directly to Wimba’s Product Management Team.
Disclaimer

This Guide, as well as the software described in it, furnished under license may only be used or copied in accordance with the terms of such license. The information in this Guide is furnished for information purposes only, is subject to change without notice, and should not be construed as a commitment by Wimba, Inc. Wimba, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this Guide. Except as permitted by such license, no part of this Guide publication may be reproduced, stored in a retrieval system or transmitted in any form by any means, electronic, mechanical, recording or otherwise without the prior written permission of Wimba, Inc. Any references to company names in examples or illustrations are for demonstration purposes only and are not intended to refer to any actual organizations.

Diploma 6 and the Wimba logos and trademarks are property of Wimba, Inc., 10 East 40th St, Floor 11, New York, NY 10016.

(c) 2009 Wimba, Inc. All Rights Reserved.