With Class - Help Index

Commands

How To - User Manual

How To - Script Manual

There are four primary ways to use the With Class Help System:
- Select F1,
- Select Shift F1,
- Select Help from Menu and Index,
- Select Help from Menu and Search.

The simplest way to access help is the select a menu item or toolbar icon and then press F1. The second way to access help is press Shift F1. With Class then enters the Help mode. The cursor turns into a large question mark. Then select any menu item or toolbar icon to show the appropriate help topic. The third way to access help is to select the Help menu item from the menu bar. Then select index to find the appropriate topic. The fourth way to access help is to select the Help menu item from the Toolbar and then select Search. Find the appropriate help topic in alphabetical order.

File Menu Commands

The File menu offers the following commands:

New	Creates a new document.
Open	Opens an existing document.
Close	Closes an opened document.
Save	Saves an opened document using the same file name.
Save As	Saves an opened document to a specified file name.
Print	Prints a document.
Print Preview	Displays the document on the screen as it would appear printed.
Edit File	Opens the text editor to edit a text file.
Exit	Exits With Class.

Edit Menu Commands

The Edit menu offers the following commands:

Copy — Copies data from the document to the clipboard as a bitmap.

Copy OLE — Copies data from the document to the clipboard as an OLE object.

Insert New Object — Inserts and embeds an object, such as a chart or an equation in a document.

Draw Menu Commands

The View menu offers the following commands:

Class Draws a Class Shape on the screen. Brings up a cross hair. Click where you wish to place the class

Relation Brings up relation pencil. Connect between two classes by holding mouse button down on one class and dragging to the connecting class.

Association Brings up a relation pencil for association relationship.

Aggregation Brings up a relation pencil for aggregation (assembly-part) relationship.

Inheritance Brings up a relation pencil for inheritance (generalization) relationship.

Link Association Brings up a relation pencil for link association relationship.

One to One Makes the branch selected a one to one relationship.

One to Many Makes the selected a one to many relationship.

Many to Many Makes the selected branch a many to many relationship.

One to Zero Branch selected becomes one to zero or one relationship.

Delete Shape Deletes the shape that is selected with the X cursor.

Draw Text Places text on the diagram.

View Menu Commands

The View menu offers the following commands:

Toolbar	Shows or hides the toolbar.
Status Bar	Shows or hides the status bar.
Show Connections	Shows or hides the small x connection symbols on each class.
Show Inheritance	Hides and shows the inheritance connection.
Show Aggregation	Hides and shows the aggregation relationship.
Show Association	Hides and shows the association relationship.
Show Class Only	This command shows or hides attributes and operations on the class diagram.
Line Thickness	This command sets the line thickness on class symbols and relationship symbols.
Show Print Border	This command shows a print border for printing.

Methodology Menu Commands

The Methodology menu offers the command to select the object-oriented methodology:

Rumbaugh	Selects the Rumbaugh Object Management Technique (OMT) methodology.
Coad-Yourdon	Selects the Coad-Yourdon OOA/OOD methodology.
Booch Command	Selects the Booch OOD methodology. .

Help Menu Commands

The Help menu offers the following commands, which provide you assistance with this application:

Index Offers you an index to topics on which you can get help.
Using Help Provides general instructions on using help.
About Displays the version number of this application.

Zoom Menu Commands

The Zoom menu offers the following commands to display the diagram in various view scales.

Zoom-25 Zooms to 25% to view the largest diagram area. Classes and text are very small.

Zoom-50 Zooms to 50% of normal view. Classes and text are smaller than normal.

Zoom-75 Zooms to 75% of normal view. Classes and text are smaller than normal.

Zoom-100 Zooms to 100% of normal view. This is the default.

Zoom-150 Zooms to 150% of normal view. Classes and text are larger than normal.

Zoom-200 Zooms to 200% of normal view. Classes and text are very large.

Generate Menu Commands

The Generate menu offers the command to generate source code files and text reports by running scripts:

GenerateCode	Runs a script to generate source code.
GenerateReport	Runs a script to generate text reports.

New Command (File Menu)

Use this command to create a new document in With Class.

You can open an existing document with the Open command.

Shortcuts

Toolbar: 🗋
Keys: CTRL+N

Open Command (File Menu)

Use this command to open an existing document in a new window. You can open multiple documents at once. Use the Window menu to switch among the multiple open documents. See Window 1, 2, ... command.

You can create new documents with the New command.

Shortcuts

Toolbar: 📂
Keys: CTRL+O

File Open dialog box

The following options allow you to specify which file to open:

File Name
Type or select the filename you want to open.   This box lists files with the extension you select in the List Files of Type box.

List Files of Type
Select the type of file you want to open, e.g. .omt files.

Drives
Select the drive in which With Class stores the file that you want to open.

Directories
Select the directory in which With Class stores the file that you want to open.

Network...
Choose this button to connect to a network location, assigning it a new drive letter.

Close Command (File Menu)

Use this command to close all windows containing the active document. With Class suggests that you save changes to your document before you close it. If you close a document without saving, you lose all changes made since the last time you saved it. Before closing an untitled document, With Class displays the Save As dialog box and suggests that you name and save the document.

Shortcuts

Toolbar: File
Keys: CTRL+C

Save Command (File Menu)

Use this command to save the active document to its current name and directory. When you save a document for the first time, With Class displays the Save As dialog box so you can name your document. If you want to change the name and directory of an existing document before you save it, choose the Save As command.

Shortcuts

Toolbar: 💾
Keys: CTRL+S

Save As Command (File Menu)

Use this command to save and name the active document.   With Class displays the Save As dialog box so you can name your document.

To save a document with its existing name and directory, use the Save command.

Shortcuts

Keys: CTRL+A

Edit File Command (File Menu)

Use this command to open, edit, and save a text file.

Shortcuts

Keys: CTRL+E

File Save As Dialog Box

The following options allow you to specify the name and location of the file you're about to save:

File Name

Type a new filename to save a document with a different name.   A filename can contain up to eight characters and an extension of up to three characters.   With Class adds the extension you specify in the Save File As Type box.

Drives

Select the drive in which you want to store the document.

Directories

Select the directory in which you want to store the document.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

1, 2, 3, 4 Command (File Menu)

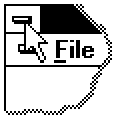
Use the numbers and filenames listed at the bottom of the File menu to open the last four documents you closed.   Choose the number that corresponds with the document you want to open.

Exit Command (File Menu)

Use this command to end your With Class session.   You can also use the Close command on the application Control menu.   With Class prompts you to save documents with unsaved changes.

Shortcuts

Mouse: Double-click the application's Control menu button.



Keys: CTRL+X

Undo/Can't Undo Command (Edit Menu)

Use this command to reverse the last editing action, if possible. The name of the command changes, depending on what the last action was. The Undo command changes to Can't Undo on the menu if you cannot reverse your last action.

Shortcuts

Toolbar:
Keys: CTRL+Z or
ALT-BACKSPACE

Copy Command (Edit Menu)

Use this command to copy selected data onto the clipboard.   This command is unavailable if there is no data currently selected.

Copying data to the clipboard replaces the contents previously stored there.

Shortcuts

Toolbar: ⎘
Keys: CTRL+INS

Copy OLE Command (Edit Menu)

Use this command to copy selected data onto the clipboard as an OLE object.   This command is unavailable if there is no data currently selected.

Copying data to the clipboard replaces the contents previously stored there.

Paste Command (Edit Menu)

Use this command to insert a copy of the clipboard contents at the insertion point. This command is unavailable if the clipboard is empty.

Shortcuts

Toolbar: 📋
Keys: CTRL+V

Insert Object Command (Edit Menu)

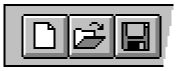
This command inserts an OLE object into the diagram such as a drawing or equation. To place an OLE object on a diagram, first select a class on the diagram. Then, select Edit - Insert Object, select Create New or Create from File and select the Object Type, e.g. Microsoft Paintbrush. The object will be placed to the right of the selected class on the class diagram.

Toolbar Command (View Menu)

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in With Class, such as File Open.   A check mark appears next to the menu item when the Toolbar is displayed.

See Toolbar for help on using the toolbar.
See Drawing Toolbar for help on using the drawing toolbar.

Toolbar

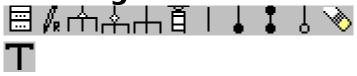


The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in With Class,

To hide or display the Toolbar, choose Toolbar from the View menu (ALT, V, T).

Click	To
🗋	Open a new document.
📂	Open an existing document. With Class displays the Open dialog box, in which you can locate and open the desired file.
💾	Save the active document or template with its current name. If you have not named the document, With Class displays the Save As dialog box.
🖨	Print the active document.
✂	Remove selected data from the document and stores it on the clipboard.
⎘	Copy the selection to the clipboard.
📋	Insert the contents of the clipboard at the insertion point.
↶	Reverse the last editing. Note: You cannot undo some actions.

Drawing Toolbar



The toolbar is displayed across the bottom of the application window. The toolbar provides quick mouse access to the drawing symbol used in With Class,

Click **To**

Draws a Class Shape on the screen. Brings up a cross hair. Click where you wish to place the class.

Brings up relation pencil. Connect between two classes by holding mouse button down on one class and dragging to the connecting class.

Brings up a relation pencil for inheritance (generalization) relationship.

Brings up a relation pencil for aggregation (assembly-whole) relationship.

Brings up a relation pencil for association relationship.

Brings up a relation pencil for link association relationship.

Makes the branch selected a one to one relationship.

Branch selected becomes one to many relationship.

Branch selected becomes many to many relationship.

Branch selected becomes one to zero or one relationship.

Deletes the shape that is selected with the X cursor.

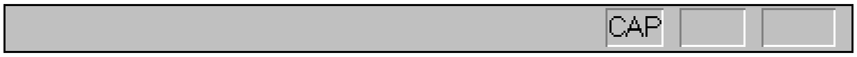
T Places text on the diagram.

Status Bar Command (Draw Menu)

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

See Status Bar for help on using the status bar.

Status Bar



The status bar is displayed at the bottom of the With Class window.   To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus.   This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them.   If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar indicate which of the following keys are latched down:

Indicator	**Description**
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.

New Command (Window Menu)

Use this command to open a new window with the same contents as the active window. You can open multiple document windows to display different parts or views of a document at the same time. If you change the contents in one window, all other windows containing the same document reflect those changes. When you open a new window, it becomes the active window and is displayed on top of all other open windows.

Cascade Command (Window Menu)

Use this command to arrange multiple opened windows in an overlapped fashion.

Tile Command (Window Menu)

Use this command to arrange multiple opened windows in a non-overlapped fashion.

Tile Horizontal Command (Window Menu)

Use this command to vertically arrange multiple opened windows in a non-overlapped fashion.

Tile Vertical Command (Window Menu)

Use this command to arrange multiple opened windows side by side.

Window Arrange Icons Command

Use this command to arrange the icons for minimized windows at the bottom of the main window.   If there is an open document window at the bottom of the main window, then some or all of the icons may not be visible because they will be underneath this document window.

1, 2, ... Command (Window Menu)

With Class displays a list of currently open document windows at the bottom of the Window menu. A check mark appears in front of the document name of the active window. Choose a document from this list to make its window active.

Index Command (Help Menu)

Use this command to display the opening screen of Help.   From the opening screen, you can jump to step-by-step instructions for using With Class and various types of reference information.

Once you open Help, you can click the Contents button whenever you want to return to the opening screen.

Using Help Command (Help Menu)

Use this command for instructions about using Help.

About Command (Help Menu)

Use this command to display the copyright notice and version number of your copy of With Class.

Context Help Command

Use the Context Help command to obtain help on some portion of With Class. When you choose the Toolbar's Context Help button, the mouse pointer will change to an arrow and question mark. Then click somewhere in the With Class window, such as another Toolbar button. The Help topic will be shown for the item you clicked.

Shortcut

Keys: SHIFT+F1

Title Bar

With Class Title Bar

The title bar is located along the top of a window.   It contains the name of the application and document.

To move the window, drag the title bar.   Note: You can also move dialog boxes by dragging their title bars.

A title bar may contain the following elements:

- Application Control-menu button
- Document Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Name of the document
- Restore button

Scroll Bars

Displayed at the right and bottom edges of the document window.   The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the document.   You can use the mouse to scroll to other parts of the document.

Size Command (System Menu)

Use this command to display a four-headed arrow so you can size the active window with the arrow keys.

✥

After the pointer changes to the four-headed arrow:

1. Press one of the DIRECTION keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a DIRECTION key to move the border.
3. Press ENTER when the window is the size you want.

Note: This command is unavailable if you maximize the window.

Shortcut

Mouse: Drag the size bars at the corners or edges of the window.

Move Command (Control Menu)

Use this command to display a four-headed arrow so you can move the active window or dialog box with the arrow keys.
✥

Note: This command is unavailable if you maximize the window.

Shortcut
Keys: CTRL+F7

Minimize Command (Application Control Menu)

Use this command to reduce the With Class window to an icon.

Shortcut

Mouse: Click the minimize icon ▼ on the title bar.
Keys: ALT+F9

Maximize Command (System Menu)

Use this command to enlarge the active window to fill the available space.

Shortcut

Mouse: Click the maximize icon ▲ on the title bar; or double-click the title bar.
Keys: CTRL+F10 enlarges a document window.

Next Window Command (Document Control Menu)

Use this command to switch to the next open document window.   With Class determines which window is next according to the order in which you opened the windows.

Shortcut

Keys:	CTRL+F6

Previous Window Command (Document Control Menu)

Use this command to switch to the previous open document window.   With Class determines which window is previous according to the order in which you opened the windows.

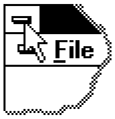
Shortcut

Keys: SHIFT+CTRL+F6

Close Command (Control Menus)

Use this command to close the active window or dialog box.

Double-clicking a Control-menu box is the same as choosing the Close command.



Note: If you have multiple windows open for a single document, the Close command on the document Control menu closes only one window at a time. You can close all windows at once with the Close command on the File menu.

Shortcuts

Keys: CTRL+F4 closes a document window
ALT+F4 closes the With Class window or dialog box

Restore Command (Control Menu)

Use this command to return the active window to its size and position before you chose the Maximize or Minimize command.

Switch to Command (Application Control Menu)

Use this command to display a list of all open applications.   Use this "Task List" to switch to or close an application on the list.

Shortcut

Keys: CTRL+ESC

Dialog Box Options

When you choose the Switch To command, you will be presented with a dialog box with the following options:

Task List

Select the application you want to switch to or close.

Switch To

Makes the selected application active.

End Task

Closes the selected application.

Cancel

Closes the Task List box.

Cascade

Arranges open applications so they overlap and you can see each title bar.   This option does not affect applications reduced to icons.

Tile

Arranges open applications into windows that do not overlap.   This option does not affect applications reduced to icons.

Arrange Icons

Arranges the icons of all minimized applications across the bottom of the screen.

Next Pane

Displays the next pane.

Prev Pane

Displays the previous pane.

Modifying the Document

Select any menu item or toolbar icon to update the current diagram.

No Help Available

No help is available for this area of the window.

No Help Available

No help is available for this message box.

Print command (File menu)

Use this command to print a document. This command presents a Print dialog box, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

Shortcuts

Toolbar: 🖨
Keys: CTRL+P

Print dialog box

The following options allow you to specify how the document should be printed:

Printer

This is the active printer and printer connection. Choose the Setup option to change the printer and printer connection.

Setup

Displays a Print Setup dialog box, so you can select a printer and printer connection.

Print Range

Specify the pages you want to print:

All	Prints the entire document.
Selection	Prints the currently selected text.
Pages	Prints the range of pages you specify in the From and To boxes.

Copies

Specify the number of copies you want to print for the above page range.

Collate Copies

Prints copies in page number order, instead of separated multiple copies of each page.

Print Quality

Select the quality of the printing. Generally, lower quality printing takes less time to produce.

Print Progress Dialog

The Printing dialog box is shown during the time that With Class is sending output to the printer.   The page number indicates the progress of the printing.

To abort printing, choose Cancel.

Print Preview command (File menu)

Use this command to display the active document as it would appear when printed. When you choose this command, the main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job.

Print Preview toolbar

The print preview toolbar offers you the following options:

Print
Bring up the print dialog box, to start a print job.

Next Page
Preview the next printed page.

Prev Page
Preview the previous printed page.

One Page / Two Page
Preview one or two printed pages at a time.

Zoom In
Take a closer look at the printed page.

Zoom Out
Take a larger look at the printed page.

Close
Return from print preview to the editing window.

Print Setup command (File menu)

Use this command to select a printer and a printer connection.   This command presents a Print Setup dialog box, where you specify the printer and its connection.

Print Setup dialog box

The following options allow you to select the destination printer and its connection.

Printer

Select the printer you want to use.   Choose the Default Printer; or choose the Specific Printer option and select one of the current installed printers shown in the box.   You install printers and configure ports using the Windows Control Panel.

Orientation

Choose Portrait or Landscape.

Paper Size

Select the size of paper that the document is to be printed on.

Paper Source

Some printers offer multiple trays for different paper sources.   Specify the tray here.

Options

Displays a dialog box where you can make additional choices about printing, specific to the type of printer you have selected.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

Page Setup command (File menu)

<< Write application-specific help here. >>

Class Command (Draw Menu)

This command draws a class.   Selecting this menu command is the same as selecting the Class Icon (Box Symbol with Three Sections) in the Drawing Toolbar.   A cursor cross-hair with a C appears.   Place and click the cursor cross-hair on the diagram where you want the class to appear.

Once the class has been placed on the diagram, enter the class name, attribute names, and operation names.   Double clicking the class symbol on the diagram brings up the Class Dialog Box.   Clicking the Spec button, brings up the Class Specification Dialog Box.   Clicking the Include button brings up the Include File Dialog Box.   Clicking an attribute, brings up Attribute Specification Dialog Box.   Clicking an operation, brings up the Operation Specification Dialog Box.

Shortcuts

Toolbar: ▤
Keys: CTRL+I

Relation Command (Draw Menu)

This menu command invokes the pencil cursor to draw a relation between classes. Selecting this menu command is the same as selecting the Relation Icon (Pencil Symbol) in the Drawing Toolbar. A cursor as a pencil appears. Click the cursor (pencil) inside the source class and then click or drag the cursor (pencil) inside the destination class. You may create a recursive relationship, i.e. a loop indicating a class has a relationship with itself. For example a part class may have an aggregation relationship with itself indicating that a part may have a part. To create a recursive relationship first click inside the class near a connection point. Then click inside the class near another connection point. Press ESC to return to the normal mode.

Shortcuts

Toolbar: T
Keys: CTRL+R

Association Command (Draw Menu)

This menu command invokes the pencil cursor to draw an association relationship between classes.   Selecting this menu command is the same as selecting the Association Icon (Fork Symbol with Lines) in the Drawing Toolbar.   After you select the appropriate cardinality, e.g. One to One, One to Many, etc. select this command.   Click or drag the cursor (pencil) from inside one class to inside another class to connect classes with an association relationship. Click inside the second class.   For example, the class car has a one to many association relationship with the class passenger.

Shortcut

Toolbar: T

Aggregation Command (Draw Menu)

This menu command invokes the pencil cursor to draw an aggregation relationship between classes.   Selecting this menu command is the same as selecting the Aggregation Icon (Fork Symbol with a Diamond) in the Drawing Toolbar.   After you select the appropriate cardinality, e.g. One to One, One to Many, etc. select this command.   Click or drag the cursor (pencil) from inside one class to inside another class to connect classes with an aggregation relationship.   Click inside the second class.   For example, class car has an aggregation relationship with the class door, class radio, class engine and class tire.

Shortcut

Toolbar: T

Inheritance Command (Draw Menu)

This menu command invokes the pencil cursor to draw an inheritance relationship between classes.   Selecting this menu command is the same as selecting the Inheritance Icon (Fork Symbol with Triangle) in the Drawing Toolbar.   For example, there is an inheritance relationship between the class vehicle and class car.   The class vehicle is the superclass and the class car is the subclass.   This is often referred to as generalization specialization.   Click or drag inside the superclass near a connection point to inside the subclass near a connection point.   Click inside the subclass.

Shortcut

Toolbar: T

Link Association Command (Draw Menu)

This menu command invokes the link association symbol to show a link association between classes. A link association holds one or more link attributes to describe the association. Selecting this menu command is the same as selecting the Link Association Icon (Box Symbol with Three Sections on a Line) in the Drawing Toolbar. After you select the appropriate cardinality, e.g. One to One, One to Many, etc. select this command. Click or drag the cursor (pencil) from inside one class to inside another class to connect classes with a link association relationship. Click inside the second class. For example, a link association between the person and company classes is "works-for". The link attributes are "salary" and "job title".

Shortcut

Toolbar: T

One to One Command (Draw Menu)

This menu command invokes the one to one cardinality. Selecting this menu command is the same as selecting the One to One Icon (Line) in the Drawing Toolbar. For example, the car class has a one to one relationship with the motor class.

Shortcut

Toolbar: T

One to Many Command (Draw Menu)

This menu command invokes the one to many cardinality. Selecting this menu command is the same as selecting the One to Many Icon (Line with Black Circle) in the Drawing Toolbar. For example, the car class has a one to many relationship with the passenger class.

Shortcut

Toolbar: T

Many to Many Command (Draw Menu)

This menu command invokes the many to many cardinality. Selecting this menu command is the same as selecting the Many to Many Icon (Line with Two Black Circles) in the Drawing Toolbar. For example, the car class has a many to many relationship with the owner class.

Shortcut

Toolbar: T

One to Zero or One Command (Draw Menu)

This menu command invokes the one to zero or one cardinality. Selecting this menu command is the same as selecting the One to Zero or One Icon (Line with Open Circle) in the Drawing Toolbar. For example, the car class has a one to zero or one relationship with the cellular phone class.

Shortcut

Toolbar: T

Delete Shape Command (Draw Menu)

This menu command invokes the X delete symbol.   The delete tool can be turned back off by toggling this menu option.   It can also be turned on and off from the X in the toolbar (palette).

Shortcuts

Toolbar: T
Keys: CTRL+D

Draw Text Command (View Menu)

This menu command permits the insertion of text on the diagram.

Shortcut

Toolbar: T

Show Connections Command (View Menu)

This command hides and shows the connection points (xs) on the class.   The default is checked to show connection points (xs).

Show Inheritance Command (View Menu)

This command hides and shows the inheritance connection between classes on the class diagram.   The default is checked to show the inheritance connection.

Show Aggregation Command (View Menu)

This command hides and shows the aggregation relationship between classes on the class diagram.   The default is checked to show the aggregation relationship.

Show Association Command (View Menu)

This command hides and shows the association relationship between classes on the class diagram.   The default is checked to show the association relationship.

Show Class Only Command (View Menu)

This command shows or hides attributes and operations on the class diagram.   If you check this command, then only the class name is shown on the diagram, in print preview, and on a printed page.   The default is that this command is not checked so that all attributes and operations are shown on the class diagram.

Line Thickness Command (View Menu)

This command sets the line thickness on class symbols and relationship symbols. Enter a value from 1 to 4. 1 is very thin and 4 is very thick. The selected line thickness is both displayed and printed. The default setting is 2.

Show Print Border Command (View Menu)

This command shows a print border for printing. Show the print border before you print a class diagram. Then select File Print Preview to see how the diagram will appear on the printed page.

Rumbaugh Command (Methodology Menu)

This menu option selects the O-O graphic notation as described in "Object-oriented Modeling and Design" by Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen.   The class symbol and relationship symbols are based upon this methodology.

Shortcut: CTRL+b

Coad-Yourdon Command (Methodology Menu)

This menu option selects the Coad-Yourdon O-O graphic notation as described in "Object-Oriented Analysis and Object-Oriented Design by Coad and Yourdon.   The class symbol and relationship symbols are based upon this methodology.

Shortcut: CTRL+Y

Booch Command (Methodology Menu)

This menu option selects the O-O graphic notation as described in "Object-Oriented Design with Applications" by Grady Booch.   The class symbol and relationship symbols are based upon this methodology.

Zoom_25 Command (Zoom Menu)

The Zoom_25 menu item zooms the diagram to 25% of normal view.   This view displays the largest diagram area.   Classes and text are very small.

Shortcut: CTRL+1

Zoom_50 Command (Zoom Menu)

The Zoom_50 menu item zooms the diagram to 50% of normal view.   Classes and text are smaller than normal.

Shortcut: CTRL+2

Zoom_75 Command (Zoom Menu)

The Zoom_75 menu item zooms the diagram to 75% of normal view.   Classes and text are smaller than normal.

Shortcut: CTRL+3

Zoom_100 Command (Zoom Menu)

The Zoom_100 menu item zooms the diagram to 100% of normal view.   This is the default. Classes and text are the normal size.

Shortcut: CTRL+4

Zoom_150 Command (Zoom Menu)

The Zoom_150 menu item zooms the diagram to 150% of normal view.   Classes and text are the larger than normal size.

Shortcut: CTRL+5

Zoom_200 Command (Zoom Menu)

The Zoom_200 menu item zooms the diagram to 200% of normal view.   Classes and text are the larger than normal size.   This zoom shows the most detail.

Shortcut: CTRL+6

Generate Code Command (Generate Menu)

The Generate Code menu item brings up the script options dialog box. Select the appropriate file extension, e.g. H. Then select the appropriate script file to execute to generate source code, e.g. CPPHEAD.SCT. A source code file for each class is generated.

Shortcut: CTRL+G

Generate Report Command (Generate Menu)

The Generate Report menu item brings up the script report dialog box. Enter the report name, e.g. REPORT.TXT. Then select the appropriate script file to execute to generate a report, e.g. RPTALL.SCT. A single report file for all classes is generated.

Class Specification Dialog Box

By clicking on the "Spec" button on the class dialog box, the class specification dialog box appears with the following specification items to describe the class:

Description: general class information, purpose, roles, essential behavior, and responsibilities.

Access: indicates the visibility or export control of a class, e.g. exported, private, or imported relative to the enclosing system or subsystem.

Cardinality: indicates how many objects (instances) of the class are permitted, e.g. 0, 1, or N (a number).

Concurrency: indicates if objects of the class are sequential or concurrent.   Alternatives are sequential, blocking, concurrent, or active.

Persistence: indicates if objects of the class are transitory or persistent.   Transitory objects lose their values when the program ends.   Persistent objects retain their values when the program ends.

File Name: indicates the file name for the class.   This is useful to reduce a long class name to eight characters as required by the MS/DOS file system.

Authors Initials: indicates the authors initials.

Library Base Class: indicates the name of the Library Base Class that is the superclass of the current car.   For example, in Visual C++ it is often necessary to make the class a derived class (subclass) of the MFC class CObject. To do this type : public CObject.   For collections this feature allows the user to derive their class for compatible collections like COblist.

Invariant: indicates the invariant of the class.   An invariant is a rule or constraint that specifies the correctness of the class.   Invariants are a part of the Eiffel Language.

Class Dialog Box

By clicking on a class on the diagram, the class dialog box appears.   This dialog box displays the following information:

- Class Name,
- Attribute Names,
- Operation Names,
- Base Classes (Superclasses),
- Aggregates - Part Classes,
- Association Classes.

The cardinality, e.g. 1:0/1,1:1, 1:M, M:M, is shown next to each aggregate - part class and association class.

This dialog box has the following buttons to invoke other dialog boxes:
- Spec Button to invoke the Class Specification Dialog Box
- Include Button to invoke the Include Dialog Box
- A Button to invoke the Attribute Specification Dialog Box
- O Button to invoke the Operation Specification Dialog Box

Attribute Specification Dialog Box

The attribute dialog is invoked by double clicking any attribute. It provides the user with the ability to enter information about each attribute. A form is provided to enter the following:

- **Attribute name** provides the name of the attribute.

- **Attribute type** provides the attribute type or class.

- **Description** provides other information about the attribute.

- **Access** (visibility) states the visibility of the attribute, e.g. public, protected, private, or implementation.

- **Initial value** provides the initial value of the attribute. This is important for code generation to set the initial value of the attribute.

- **Minimum value** provides the minimum value of the attribute.

- **Maximum value** provides the maximum value of the attribute.

- **Constraints** (limits or restrictions) provides rules, limits, or restrictions of the attribute.

A constraint is a rule or limitation on a attribute value, e.g. worker_salary must be less than the boss_salary. For code generation ensure the initial value is filled-in.

Note: After entering an attribute name, enter semicolon (;) to enter another attribute or press Add Button

External Include File Dialog Box

By clicking on the "Include Button" in the Class Dialog Box, the user can insert the name of an external include file.   For example, if an attribute is a String, e.g. String color then the appropriate include file might be strng.h.

Note: After entering an include file type comma or press Add button.

Line Thickness Dialog Box

Selecting the Draw Line Thickness menu item invokes the Line Thickness Dialog Box.   Enter a value of 1 to 4 to set the line thickness for the class diagram and relationship lines.   1 is thinnest and 4 is thickest.   2 is the default.

Operation Specification Dialog Box

The operation dialog is invoked by double clicking any operation. It provides the user with the ability to enter information about each operation. A form is provided to enter the following:

- **Operation name** states the name of the operation.

- **Return type or class** states the return type or class, e.g. integer, float, string, Passenger, etc.

- Parameters (arguments) state input parameters, e.g. integer anInteger, float aFloat, string aString. An operation may have several input parameters. Note that you should enter the Parameter Class or Type first followed by the Parameter Name. There are several script variables to print out the parameters in various formats, e.g. CPP_OPERATION_PARAMETERS and PASCAL_OPERATION_PARAMETERS,

Note: After entering a parameter, type a comma or press the Add button.

- **Description** states a description of the operation.

- **Access** states the visibility of the operation, e.g. public, protected, private, or implementation

- Abstract/Normal states whether an operation is normal or abstract. A normal operation, e.g. not abstract, has an implementation. An abstract operation is declared but not implemented in an abstract class, e.g. a C++ pure virtual function.

- **Classification** states the category of the operation e.g. modifier, selector, iterator, constructor, destructor. A **modifier** is an operation that alters the value or state of an object. A **selector** is an operation that accesses the value or state of an object but does not alter the value or state of an object. An **iterator** is an operation that permits the parts of an object to be visited. A **constructor** is an operation that creates an object and/or initializes an object. A **destructor** is an operation that deinitializes an object and/or destroys the object itself.

- **Precondition** states a condition that must be satisfied for the correct transformation.

- **Postcondition** states a condition that ensures that the correct transformation has occurred.

- **Exception Type** states the type or class of the exception that is raised (thrown) by the operation.

- **Exception Name** states the name of the exception that is raised (thrown) by the operation.

- **Concurrency** states whether the operation is a sequential or concurrent operation. A sequential operation is a member of a sequential class. A concurrent operation is a member of a concurrent class. See Object-Oriented Design with Applications by Grady Booch.

- **Transformation** states the formula or expression of the operation.

A user may enter C++ source code by hitting the Code Button. This button opens up a simple editor in which the user only need enter the lines of code contained inside the brackets of the function.   Hitting exit will save the code to the program.

Check boxes are provided to designate an operation as virtual, static, const, or pure virtual.

Note: After entering an operation name, enter a semicolon (;) to enter another operation name or press the Add button.

Overview of With Class CASE Tool

With Class is an object-oriented CASE (Computer Aided Software Engineering) tool. It is a tool to assist a software developer or programmer to graphically model a system using an object-oriented methodology. With this tool you can take a problem such as process control system or information registration system and model it in terms of classes, attributes, and operations. You can create a class diagram to graphically display the classes, attributes, and operations in the system. Then you can fill-in forms to specify in detail each class, attribute, and operation. You can print out this information in reports. You can generate C++ source code files.

With Class supports the following major S/W development functions:
- creation of class diagrams,
- collection and storage of class, attribute, and operation information,
- creation of C++ and other source code files with scripts,
- creation of reports with scripts,
- creation of new scripts to generate source code and text report files.

With Class has a built-in script language to create custom reports and generate source code files for C++, Eiffel, and other languages. The script language consists of the following:
- Literals, e.g. Report Number 1,
- Variables, e.g. CLASS, ATTRIBUTE_NAME, OPERATION_NAME, etc,
- Operators, e.g. repeat operation [] and concatenation operation $.

As a user, you are able to run predefined scripts provided in With Class. Additionally, you are able to create your own custom scripts. See the With Class Script Manual for details.

Installing With Class

The following are the minimum system requirements for With Class:
- 80386 processor or higher (80486 or higher recommended)
- MS-Windows 3.1 or higher
- Mouse that supports MS-Windows
- VGA monitor - 1024x768 or higher resolution recommended

To initially install and run With Class, follow these steps:

>> Copy all With Class files into a directory, e.g. wc2

>> Ensure CTL3D.DLL in your path, e.g. C:\WINDOWS\SYSTEM is in your autoexec.bat path statement

>> Ensure the following files are in the C:\WINDOWS\SYSTEM directory: COMPOBJ.DLL, STORAGE.DLL, OLE2.DLL, OLE2DISP.DLL, OLE2NLS.DLL.

>> Copy MFCOLEUI.DLL into C:\WINDOWS\SYSTEM

>> In Windows Program Manager, create the With Class icon by selecting "File New", "Program Icon", and entering WC2.EXE

>> In Windows, run With Class from Windows, e.g. "File Run WC2.EXE" or double click the With Class icon

>> Select "File - Set Directory" to create a new directory, e.g. c:\car

>> Select "File - Printer Setup" to select your printer

To install With Class from a disk provided from MicroGold: Type from windows program manager FILE | RUN a:install.exe. Then from DOS prompt type **copy c:\wc2\*.dll c:\windows\system**
Note: No not overwrite any installed DLL unless you have problems installing With Class or using another Windows application. The DLLs on the installation disk are the most recent DLLs available from Microsoft.

Steps to Create a Class Diagram

The following are the steps to create a class diagram with a single class:

>> Create a directory for Car Products, e.g., c:\md car

>> Run With Class   from Windows

>> Select "File - New"

>> Select "File - SaveAs" e.g., c:\car\car.omt

>> Select Class Icon from the Toolbar or Draw menu and place it on the diagram

>> Click in the upper box of the class shape and enter the class name, e.g., Car

>> Click in the middle box of the class shape (or press TAB) and enter an attribute, e.g. speed

>> Click in the lower box of the class shape (or press TAB) and enter an operation, e.g. start

>> Select "File - Save" to save the diagram

>> Select "Print" to print the diagram

Press ESC to return to the normal mode.

Press Tab to shift between class name, attributes, and operations.   Press Tab after entering the class name to enter attributes.   Press Tab after entering attributes to enter operations.

After selecting the class or relationship icon, hold the Shift key down to repeat placing classes or relationships.

To add attributes or operations in the Attribute or Operation Dialog Box, enter a semicolon (;) after each entry or click add button.

To enter a relationship name, click on the green/gray circle on the relationship line.

Steps to Show Relationships Between Classes

The following are the steps to show relations between classes:

>> In With Class create several classes.

>> To associate two classes, first select the appropriate cardinality symbol (1 to 1, 1 to many, etc) from the toolbar or Draw Menu, second select the association icon (fork symbol with straight lines) from the toolbar or Draw Menu, third click the pencil inside one class and then click inside the second class near a connection point. Alternatively drag the pencil from inside one class to inside the second class.

>> To name the association relationship, click on the green/gray dot on the relationship line and enter a name, e.g. hasA.

>> To associate three or more classes, first associate two classes, second click on the association icon to display the pencil, third click the pencil on the green/gray dot and then click inside the third class. To name the association, click on the green/gray dot and press TAB to get to the appropriate text block. Alternatively drag the pencil from the green/gray dot to inside the third class.

>> To show aggregation between two classes, first select the appropriate cardinality symbol (1 to 1, 1 to many, etc) from the toolbar or Draw Menu, second select the aggregation icon (fork symbol with diamond) from the toolbar or Draw Menu, third click the pencil inside the assembly class near a connection point, then click inside the part class near a connection point. Alternatively drag the pencil from inside one class to inside the second class.

>> To name the aggregation relationship, click on the green/gray dot on the relationship line and enter a name, e.g. partOf

>> To show aggregation between three or more classes, first connect two classes, second click on the aggregation icon to display the pencil, third click the pencil on the green/gray dot and then click inside the third part class. To name the aggregation, click on the green/gray dot and press TAB to get to the appropriate text block. Alternatively drag the pencil from the green/gray dot to inside the third part class.

>> To show inheritance between two classes, first select the inheritance icon (fork with triangle) from the toolbar or Draw Menu, second click the pencil inside the superclass near a connection point and then click inside the subclass near a connection point. Alternatively drag the pencil from inside the superclass to inside the subclass.

>> To show inheritance between a superclass and two or more subclasses, first connect the superclass with a subclass, second click on the green/gray dot and then click inside the other subclass. Alternatively drag the pencil from the green/gray dot to inside the other subclass.

Note 1 - Press ESC to enter the normal mode. With a class selected, press TAB to move the cursor between the top section (class name), middle section (attributes), and bottom section (operations). The relation text field is located at the end of the relation line. Use this text field to enter a variable name that can be used in code generation, e.g. currentMotor.

Note 2 - Click on the green/gray button on the relationship line to enter the name of the relationship. The relation text field is located at the end of the relation line. Use this text field to enter a variable name that can be used in code generation, e.g. currentMotor.

Note 3 - To create a recursive relationship, click once inside the class near a connection point. Then click the pencil inside the class near another connection point. A recursive relationship appears as a loop showing a class that has a relationship with itself, e.g. a part may have another part.

Note 4 - After selecting the class or relationship icon, hold the Shift key down to repeat placing classes or relationships

Generating Source Code and Reports

With Class has a sophisticated script language to permit users to generate source code and report text files from With Class diagrams and specifications.   This section states how to run a script.   If you desire to create a script yourself, read the With Class Script Manual.

Steps to Generate C++ or Other Code Using With Class

In With Class a code generation script creates a file for each class.   The following are the steps run a C++ generation script.

>> Run With Class   from Windows

>> Select "File - Open", e.g., c:\car\car.omt

>> Select "Generate - Generate Code" and select the appropriate script, e.g. cpphead.sct to generate the source code text file

>> Select "File - Edit File" or run a Windows editor, e.g. Notepad to view and print the text file.

Note - After you have generated code, switch to your C++ environment to create a project, to create a main function, and to compile, link, and run.

Steps to Enter Class Information and Create Text Reports

In With Class a report script creates a single report test file for all classes.   The following are the steps to run a report script, e.g. rptall.sct.

>> In With Class create a class with several attributes and operations

>> Double click on the class shape to bring up the class dialog box to enter class specification information, include files, attribute information, and operation information

>> In the class dialog box, double click on an attribute and then enter attribute information, e.g. type, description, access (visibility), etc.

>> In the class dialog box, double click on an operation and then enter operation information, e.g. return type, parameters, access (visibility), etc.

>> Select "Generate - Generate Report" and select an appropriate script, e.g. rptall.sct to generate the report text file

>> Select "File - Edit File" or run a Windows editor, e.g. Notepad to view and print the report text file.

Overview of Script Language

1. Introduction

With Class has a built-in script language to create custom reports and generate source code. The script language consists of the following:
- Literals, e.g. This is a report
- Variables, e.g. CLASS, ATTRIBUTE_NAME, OPERATION_NAME, etc,
- Operators, e.g. repeat operation [] ,concatenation $, DELETE_LAST_SYMBOL, TRUNCATE_EIGHT and NO_REPEAT.

The following are the steps to use With Class to create OMT class diagrams, create reports, and generate source code:
- Create the class diagram,
- Save the diagram and the internal With Class database.
- Identify a provided With Class script or create a new script that accesses the internal database to create reports and C++ code.
- Generate a text file for a report or C++ code listing by running the script.

2. Overview of With Class Script Language

The script language provides a means to create custom reports and C++ code files. Previous versions of With Class had an internal database that was accessible only from the internal C++ program. It was difficult to make even the most minor changes to reports or generated C++. Just to provide a new C++ comment or change the indention of C++ statements required a new With Class program. Previous versions of With Class were not extendible. This version is a highly extendible, modifiable CASE tool. With the new script language, any user can create their own scripts to create their own customized reports and language files. Scripts can be written to generate source code for any language and library. The following is a very short script (script1.sct) to generate a C++ header file:

```
------------Sample With Class Script -SCRIPT1.SCT--------------
// Class: CLASS

class CLASS_NAME : public [BASE_CLASS]
{
  [ATTRIBUTE_TYPE ATTRIBUTE_NAME;]
  public:
  [OPERATION_RETURN_TYPE OPERATION_NAME (CPP_OPERATION_PARAMETERS);]
  [ATTRIBUTE_TYPE Get$ATTRIBUTE_NAMES$() {return ATTRIBUTE_NAME;
} ]
}
```

```
-----------Sample With Class Diagram Information---------------
Base Class - Vehicle
Class - Car
Attributes: speed, gasQuantity
Operations: start, stop
```

```
---------Sample Script Output File car.h-----------------------

// Class: Car
class Car : public Vehicle
{
    int speed;
    int gasQuantity;
public:
    void start();
    void stop();
      Getspeed(){return speed;}
      GetgasQuantity(){return gasQuantity;}
}
```

3.   Major Components of the With Class Script Language

There are several components of the With Class script language:

3.1 **Class diagram and internal database** - these hold system and class information e.g. classes, attributes, operations, etc.   The class diagram and the internal database is the information that the script uses to create script output files.

3.2 **Script of script sentences** - a script is a set of sentences which consist of variables, literals, repeat operator [], and concatenation operator $.   For example, script1.sct above is a sample script.

3.3 **Script Editor** - any text editor, e.g. Windows Notepad.   One uses the text editor to create scripts.

3.4 **Script Interpreter** - With Class "Generate" menu item executes (interprets) a script. The script interpreter creates the script output file, e.g. REPORT.TXT, CAR.H, etc.

3.5 **Script Output File** - When a script is executed, then an output file, e.g. a report, a C++ header file, a C++ source file, or any text file is created.

3.6 **Script Variables** - Variables are special terms that represent information in the internal With Class database.   Sample variables are CLASS, BASE_CLASS, ATTRIBUTE_NAME, and OPERATION_NAME.   All variables are in upper case.   When a script is executed, then actual model information is substituted for the variable in the output file.   For example, the actual model information, e.g. Student is substituted for the variable CLASS.

3.7 **Literals** - a literal is any character or number that a user enters into a script.   When a script is executed, then the literal is placed into the output file.   For example, a literal in the above program is // Class: .

3.8 **Concatenation Operator $** - the symbol $ means to connect a literal and a variable. For example, Get$ATTRIBUTE_NAME connects the literal Get and the variable ATTRIBUTE_NAME.   When executed, the result is Getspeed.

3.9 **Repeat Operator []** - the symbol [] causes the variable or literal to be repeated.   For example, [ATTRIBUTE_NAME] causes a list of attributes to be placed into the output file. Anything placed in between the brackets will be repeated for every variable.

3.10 **DELETE_LAST_SYMBOL** operator removes the trailing punctuation marks.   This is useful in code generation.   (See example for NO_REPEAT)

3.11 **TRUNCATE_EIGHT** operator truncates a class name to eight characters.
(e.g. [#include "TRUNCATE_EIGHT$BASE_CLASS.h "]   )

3.12 **NO_REPEAT** operator prints a literal or punctuation symbol once without repeating it.
 (e.g.   class CLASS_NAME   [NO_REPEAT: NO_REPEAT public BASE_CLASS,

DELETE_LAST_SYMBOL])

3.13 **LITERAL_SYMBOL** operator treats a bracket '[' as a literal
(e.g. LITERAL_SYMBOL[10 LITERAL_SYMBOL])

3.14 **NO_RETURN** operator prints literals and script variables on the same line.
(e.g. class CLASS_NAME [NO_RETURN : public BASE_CLASS, DELETE_LAST_SYMBOL]
)

4. Steps to Create a Script

4.1 Manually create the sample output file to be generated from the script, e.g.

```
Class Attribute Report
Class: Student
Attributes: age, gradePointAverage
```

4.2 Create the class diagram to be used with the script. For example, create a class with the class name "Student" with the attributes "age" and "gradePointAverage".

4.3. Analyze the class diagram to identify variables. For example, the class diagram has the variable CLASS and the variable ATTRIBUTE_NAME. When the script is executed "Student" is substituted for the variable CLASS and "age" is substituted for the variable ATTRIBUTE_NAME.

4.4 Analyzed the sample output file to identify literals, variables, operation symbol [], and concatenation symbol $. For example, see the comments below:

```
Class Attribute Report             -- Literal
Class: Student                     --Class: Literal; Student is Variable CLASS
Attributes: age, gradePointAverage --Attributes: Literal; --age is Variable
ATTRIBUTE_NAME              --comma is literal;
                            --gradePointAverage is Variable
ATTRIBUTE_NAME
                            --age, gradePointAverage needs repeat symbol []
```

4.5 Create the script as follows:

- First, place literals in the script as they would appear in the output file, e.g.

```
Class Attribute Report
Class:
Attributes:                 ,
```

- Second, place variables in the script where actual information from the class diagram would appear in the output file, e.g.

```
Class Attribute Report
Class: CLASS
Attributes: ATTRIBUTE_NAME,
```

- Third, place operators in the script, e.g. repeat symbol [], concatenation symbol $, NO_REPEAT, DELETE_LAST_SYMBOL, and TRUNCATE_EIGHT. Use the repeat symbol [] whenever a literal or variable must be repeated. For example, since a class has many attributes then attribute name should be repeated, e.g. [ATTRIBUTE_NAME]. Use the concatenation symbol $ to put together a literal and/or variables without a space. For example, Get$ATTRIBUTE_NAME will result in Getage or GetgradePointAverage in the output file.

```
Class Attribute Report
Class: CLASS
Attributes: [ATTRIBUTE_NAME,]
```

4.6 Double check the script and the class diagram to ensure the following:
- All literals are spelled correctly,
- All variables are spelled correctly, (e.g. ATTTRRIBUTE_NNNAME)
- All repeat operators have matching brackets, e.g. [ATTRIBUTE_NAME],
- There are not repeat operators within repeat operators, e.g. [[ATTRIBUTE_NAME]],
- Relationship, Attribute, and Operation script variables appear within a repeat operator, e.g. [ASSOCIATION_ONE_NAME],
- Class diagram and text specifications are up to date, e.g. Attribute Initial Value and Name on each association/aggregation relationship line is filled in. Operation return type is filled in.

4.7 Execute your script in With Class by selecting "Generate - Generate Code" and then selecting the script file name, e.g. script1.sct. Ensure that you closed the script file in the text editor. Check to ensure that your generated output file is correct. If not, then update your script and/or your class diagram to generate the correct output.

Certain scripts may not work properly. If you have questions on scripting,
Please send the script and the class diagram to MicroGold Software
71543.1172@compuserve.com for analysis.

5. Steps to Insert Script Generated Reports into Another Windows Application

Frequently, it is desirable to insert model information into another Windows application. For example, it is desirable to insert a data dictionary report into a Windows word processor (Microsoft Word) for formatting, sorting, and printing. It is desirable to insert an information report into a Windows spreadsheet or database application (Microsoft Excel or Access) for special analysis, sorting, and reports. The following are the steps to insert a generated data dictionary report into Microsoft Word 6.0.

1. In With Class, create a class diagram and fill-in text specifications. Ensure that the class, attribute, operation, and relationship names are entered. Missing names may appear as a blank in the data dictionary.

2. In With Class, run a report to create the data dictionary, e.g. worddic1.sct. This script has a TAB field delimiter and a CARRIAGE RETURN record delimiter. This script creates the file REPORT.TXT. Check REPORT.TXT to ensure that there are three fields of information in each line separated by a TAB. Each line should end with a CARRIAGE RETURN. You may need to remove any extra CARRIAGE RETURNs and to insert a -- in any blank fields. These actions are required to ensure the information is correctly imported into the Word document.

3. In Microsoft Word, create a document, e.g. datadic1.doc. You may use the Word template datadic1.dot which provides a date, title, and column heading titles. Enter the system name.

4. In the Microsoft Word document, select Insert Database. Then select Get Data and enter REPORT.TXT. Enter the TAB field delimiter and the CARRIAGE RETURN record delimiter. Select Insert Data. A table containing the model information is created.

5. In the Microsoft Word document, enter the column headings, e.g. Name (Entity), Type of Entity, Class if required. You may need to remove any extra CARRIAGE RETURNs and to insert a -- in any blank fields.

6. In the Microsoft Word document, select the first column (Name) and sort the names. As desired sort and format the table.

Script Variables

Script variables represent information in the class diagram and internal database. Script variables are upper case. Upon execution, model information is substituted for each variable. Generally, there is a script variable for each piece of information on a class diagram or specification text box. The following are the script variables.

1 Class variables access class information. Since there is only one piece of information for each class, these variables are not used within the [] (Repeat Operator).

CLASS_NAME or CLASS	Prints class name
CLASS_CARDINALITY	Prints the class cardinality
CLASS_PERSISTENCE	Prints the class persistence
CLASS_CONCURRENCY	Prints the class concurrency
CLASS_ACCESS	Prints the class access
CLASS_DESCRIPTION	Prints the class description
CLASS_FILENAME	Prints the class file name if assigned
CLASS_AUTHOR_INITIALS	Prints the author initials if assigned
CLASS_INVARIANT	Prints the invariant (rule)
CLASS_LIBRARY_BASE_CLASS	Prints name of library base class

2 Relationship variables access relationship information. Since there are multiple relationships for each class, these variables are used within the [] (Repeat Operator).

BASE_CLASS	Prints superclass name for a class
ASSOCIATION_CLASS	Prints associated class name
ASSOCIATION_NAME	Prints attribute name for association
ASSOCIATION_ONE_CLASS	Prints class name for a 1 to 1 association
ASSOCIATION_ONE_NAME	Prints attribute name for a 1 to 1 association
ASSOCIATION_MANY_CLASS	Prints class name for a 1 to M association
ASSOCIATION_MANY_NAME	Prints attribute name for a 1 to M association
AGGREGATION_CLASS	Prints aggregation (part) class
AGGREGATION_NAME	Prints attribute name for aggregation (part) class
AGGREGATION_ONE_CLASS	Prints class name for a 1 to 1 aggregation
AGGREGATION_ONE_NAME	Prints attribute name for a 1 to 1 aggregation
AGGREGATION_MANY_CLASS	Prints class name for a 1 to M aggregation
AGGREGATION_MANY_NAME	Prints attribute name for a 1 to M aggregation
INCLUDE_FILE	Prints name of include file

The following special C++ variables do require the Repeat Operator:

CPP_BASE_CLASSES	Prints all superclasses for a class
CPP_ASSOCIATIONS	Prints class name and attribute name for all associated classes for a class in C++ format
CPP_AGGREGATIONS	Prints class name and attribute name for all aggregation (part) classes for a class in C++ format

3 Attribute variables access attribute information. Since there are multiple attributes for each class, these variables are used within the [] (Repeat Operator).

ATTRIBUTE_NAME	Prints attribute name
ATTRIBUTE_TYPE	Prints attribute type
ATTRIBUTE_INITIAL_VALUE	Prints attribute initial value

ATTRIBUTE_ACCESS	Prints attribute visibility
ATTRIBUTE_MAX_VALUE	Prints attribute maximum value
ATTRIBUTE_MIN_VALUE	Prints attribute minimum value
ATTRIBUTE_CONSTRAINT	Prints attribute constraint or rule
ATTRIBUTE_INFO	Prints attribute information (remarks)

The following special C++ variable requires the Repeat Operator:

CPP_ATTRIBUTES	Prints all attribute type and name for a class in C++ format

4 Operation variables access operation information. Since there are multiple operations for each class, these variables are used within the [] (Repeat Operator). However the parameter variables, e.g. CPP_OPERATION_PARAMETERS and PASCAL_OPERATION_PARAMETERS do not require a separate [] (Repeat Operator). Do not place parameter variables within repeat operators.

OPERATION_NAME	Prints operation name
OPERATION_RETURN_TYPE	Prints operation return type
OPERATION_ACCESS	Prints operation visibility
OPERATION_TRANSFORMATION	Prints transformation (formula)
OPERATION_PRECONDITION	Prints precondition
OPERATION_POSTCONDITION	Prints postcondition
OPERATION_EXCEPTION_TYPE	Prints exception type or class
OPERATION_EXCEPTION_NAME	Prints exception name
OPERATION_CLASSIFICATION	Prints classification
OPERATION_CONCURRENCY	Prints concurrency
OPERATION_CODE	Prints code
OPERATION_COMMENT1	Prints comment 1
OPERATION_COMMENT2	Prints comment 2
OPERATION_COMMENT3	Prints comment 3

The following special C++ variables do require the Repeat Operator.

CPP_OPERATIONS	Prints all operations for a class in C++ format
CPP_OPERATION_VIRTUAL	Prints the virtual keyword if checked
CPP_OPERATION_CONSTANT	Prints the keyword const if checked
CPP_OPERATION_STATIC	Prints the keyword static if checked
CPP_OPERATION_PURE_VIRTUAL	Prints = 0 if checked

The following special C++/Pascal variables does not require the Repeat Operator.

CPP_OPERATION_PARAMETERS	Prints operation parameters in C++ format, e.g. Class/Type Name Parameter Name - int aNumber
PASCAL_OPERATION_PARAMETERS	Prints operation parameters in Pascal format, e.g. Parameter Name : Class/Type Name - aNumber: Integer
PASCAL_OPERATION_PARAMETERS_PAREN	Prints operation parameters in Pascal format with parentheses, e.g. (aNumber : Integer)
PASCAL_OPERATION_PARAMETERS_WITH_VAR	Prints operation parameters in Pascal format with parenthesis and Var parameter, e.g. (aNumber : Integer; Var aReturn : Integer)
PASCAL_OPERATION_RETURN_TYPE	Prints : return type in Pascal format, e.g.

: Integer

PASCAL_VIRTUAL Print virtual ; if the operation is virtual

ADA_OPERATION_PARAMETERS Prints parameters in Ada format for use in a package, e.g.
(The_Car : in out Car;
 Number1 : in Integer;
 A_Integer : out Integer)

SMALLTALK_OPERATION_PARAMETERS Prints parameters in Smalltalk format, e.g.
parameter1 with: parameter2 with: parameter3

ACTOR_OPERATION_PARAMETERS Prints parameters in Actor 4.0 format, e.g.
, parameter1, parameter2, parameter3

CLASS_PERSISTENT Prints the term persistent if the class is persistent

Glossary

Action - any response to an event. Actions may be updating an attribute, sending a message, or similar action.

Aggregation (Strong Association) - "Part of" or "bill of material" connection between an assembly and parts that has special semantics for "part of" (transitivity, antisymmetric, and propagation) and for the creation, copy, and deletion of an assembly and parts.

Assertion - a rule or expression for correctness, such as a data value must always be greater than zero.

Association - A link, connection, or mapping between two or more objects, such as "has a", "knows about", "part of", or "bill of materials".

Atomic Class - An atomic class is a primitive data type whose objects are not logically decomposable, such as character, Boolean, integer, and floating point types.

Attribute - A characteristic or property of an object. An attribute has a name (ID), class or type, and a value.

Class - A definition for one or more objects that have common attributes, common behavior, common relationships, and common semantics. In S/W a module that encapsulates attributes, operations, exceptions, and relationships.

Cohesion - the degree of internal relatedness of elements within a larger, more complex entity.

Collection Class - A collection class defines an object that holds other objects called elements. Elements may be added or removed from the collection object. Elements are stored in the collection object with an index. Sample collection classes are array, sets, bags, lists, stacks, queues, rings, trees, etc.

Composite Class - A composite class defines an object that has attributes and associated/part objects.

Condition - a guard or Boolean that may affect the stimulus response logic. A condition is a guard or Boolean expression signifying OK or NOTOK that are used in IF Condition = True THEN DoSomeAction. Examples of conditions in a Temperature Class might be "temperature high" and "temperature OK".

Control Class - defines time oriented or state based stimulus response behavior including rules and logic to respond to events.

Coupling - the degree of interconnectivity, interdependence, joining and linking of entities.

Dynamic Binding - Late binding (association) of an object name with an object and its class at run time. The object name may later be associated with a different object and its class. Also, the run time look-up of the correct version of a polymorphic operation.

Entity Class - manages and computes application information. An entity class is independent of protocol details of other interacting systems.

Event - an occurrence or physical phenomenon in the external environment that occurs at a point in time such as a user pressing a button to which one or more systems must respond.

For finite state machines, an event is any stimulus to an object that results in some action and that may result in a transition to a new state.

Exception - An abnormal, unusual error condition that may result in an operation performing incorrectly. An exception check, such as a precondition or postcondition check, is an expression that detects the presence of an exception and invokes an exception handler. An exception handler takes some action, such as attempts correct the abnormal condition or notify a user.

Finite State Machine - an entity that has state based stimulus response behavior in which there may be different actions from the same event depending upon the state of the entity.

Generalization Specialization - An "is a" or "type of" connection between superclasses and subclasses.

Generic Parameterized Class (Template) - a class that can be modified with parameters to contain or operate on objects of the parameter class, such as a parameterized Stack Class.

Inheritance - The capability of a subclass receive for use attributes and operations defined in a superclass.

Initializer Class - a top level class that initializes the system. It becomes the "main" in a C++ program.

Interface Class - defines communications with another interacting system. An interface class handles events and/or responses. An interface class isolates protocol details to communicate with other interacting systems.

Invariant - a general rule or expression that must be satisfied at all times by all applicable operations.

Message - A call to an object of a class to invoke one of its operations. A one way message is in one direction only from a requester to a server. A two way message is a peer to peer message objects both send and receive messages from each other, i.e. each can initiate a message to each other.

Object Oriented Modeling - A term referring to the modeling phases of object oriented S/W development including analysis, design, and prototyping. It does not include implementation, productization, testing, etc..

Object Oriented Design - A software development methodology (set of steps) to build systems consisting of classes and objects.

Object - A thing; an instance of a class. An entity that has state (retained information), has behavior (responds to messages), sends and/or receives messages from other objects, and has relationships with other objects. In S/W a variable defined by a class.

Object Oriented Programming - A method to develop software using inheritance, dynamic binding, and polymorphism with object oriented languages such as Smalltalk, Actor, C++, Eiffel, Object Pascal, etc.

Operation precondition - a rule or expression that must be satisfied before the execution of an operation for correct results.

Operation postcondition - a rule or expression that is satisfied upon the correct execution of an operation.

Operation - An action, service, procedure, function that performs some action in response to a message.

Pattern - Two or more entities with a well defined purpose, behavior, connections, and structure, such as a tree pattern.

Polymorphism (one name many forms) - An object name may refer to objects of different classes. An operation name may refer to different implementations.

Problem Domain Object - an object that exists in the system environment that is passed in a message to or from the system.

Relationship - A link or connection between classes or objects e.g. association "has a", aggregation "part of", and generalization specialization "is a" or "type of".

Response - a message from the system to other H/W and S/W systems. A response implements some action requested in an event.

State represents a mode of behavior that has a unique combination of events, conditions, actions, and next state. A state is static, i.e. waiting for an event to arrive. While in a state, a defined set of rules, laws, and policies apply. A state is like a manager or coordinator that knows how to respond to each event according to his rules, laws, and procedures.

Static Binding - The association of an object name (ID) with an object and its class at compile time. The object name (ID) is permanently bound to the object and its class for the life of the program.

Subclass - A refined, more specific class of a superclass. It defines more specific, attributes, operations, and exceptions.

Subsystem - a component of a larger system environment. A subsystem has components, such as smaller subsystems or classes that are connected together.

Subtype class - defines a specialized class. An object of the subtype may be substituted for an object of the supertype. Objects of supertype and subtype respond to the same messages.

Superclass - A general class that defines the most general attributes, operations, and exceptions which may be inherited by subclasses.

Supertype class - defines a general class that has the same operations (same protocol) as the specialized subtype classes.

System Environment - a complex system that has other systems (subsystems) as components.

System - a general term for a complex entity that can be treated as a unit and that has simpler components that work together to perform a function. The system is the S/W system to be developed. The system which is the center of focus and which becomes a single program (.EXE file). The system generally consists of 10 to 100 classes as a very rough order of magnitude.

Transformation - a description of how a data value may be correctly changed in a formula, expression, table, etc.

Transition - a unique pattern of an event, conditions, actions, and a destination state. For each state identify applicable events. Then for each event identify the applicable conditions, actions, and the destination state.

Weak Association - "Has a" or "knows about" connection between associated objects that does not have aggregation semantics.