# Sample Macros

**DISCLAIMER OF WARRANTY**

**The samples described in this document are undocumented sample code. These samples are provided "AS IS" WITHOUT WARRANTY OF ANY KIND. MICROSOFT FURTHER DISCLAIMS ALL IMPLIED WARRANTIES INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE SAMPLES REMAINS WITH YOU.**

**IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SAMPLES, EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.**

This document contains information about the sample Microsoft Excel, Microsoft Visual Basic, and Microsoft Word macros included with Microsoft Project. The purpose of these macros is to help you understand how to create your own macros using dynamic data exchange (DDE) to communicate between Microsoft Project and other applications. Although these macros can be used to perform the functions described in this document, they are provided to you as samples only.

# Microsoft Excel Macros

Four Microsoft Excel macros are included with Microsoft Project:

- INFLATE.XLA lets you enter and calculate inflation rates, and pass the increased costs back to Microsoft Project as fixed task costs.
- PROJPERT.XLM helps you estimate durations for tasks using the PERT method of averaging the best, worst, and expected durations.
- PROJXTAB.XLM creates a "crosstab" report listing either each task followed by its resources, or each resource followed by its tasks.
- TRACKING.XLM is similar to PROJXTAB.XLM but focuses on tracking actual resource work on a task-by-task basis. You can update actual work in Microsoft Excel and send it back to Microsoft Project.

These macros are located in the EXCEL subdirectory of the SAMPLES.DDE subdirectory in your working Microsoft Project directory.

To run these macros, you must have a copy of Microsoft Excel, version 3.0a or later. You also need Microsoft Excel to view or modify the macros. The comments in the macros will help you understand how Microsoft Excel and Microsoft Project can use

DDE to work together.

## INFLATE.XLA

### Variable Resource Rate Calculations for Microsoft Project using Microsoft Excel

INFLATE.XLA can be used to calculate more accurately the cost of your project through time. Using the macro, you enter multiple "standard rates" for each of your resources, producing accurate cost information on a task-by-task basis. Because the inflated rates are stored only as task information in Microsoft Project, the changes in resource rates are not reflected numerically in your project's resource information.

INFLATE.XLA creates an inflation worksheet for each project or resource pool for which you want to use variable resource rates. On the worksheet is an inflation table for each resource in the project. Each column in this inflation table represents a change in one or more of the rates associated with the resource. For example, to change the standard rate of a resource to $75.00/h as of September 20, 1992, you would add a new column. Then type "9/20/92" in the Effective Date row, "75" in the Standard Rate row and copy the information from the other two rows into the new column.

To run this set of macros, start Microsoft Excel and open INFLATE.XLA. This is an Add-In file, so no new windows are created when you open it. However, commands are added to the File and Data menus. These commands are described below.

The macro sheet for INFLATE.XLA is called INFLATE.XLM. You can look at the macro sheet to understand how the macro works with Microsoft Project.

All resource rates should be entered in hours, and the default work unit (set in the Preferences dialog box, available from the Options menu in Microsoft Project) should be set to hours. You must also make sure the project that corresponds to the currently active inflation worksheet is active in Microsoft Project before choosing any of the commands (except the New Inflation Worksheet command on the File menu). Although you could change this, INFLATE.XLA does not make any calculations for Overtime Rate or Cost Per Unit.

**To use INFLATE.XLA:**

1. In Microsoft Project, open the projects for which you want to add inflation factors for the resources.
2. In Microsoft Excel, choose Open from the File menu, and open INFLATE.XLA.
3. From the File menu, choose New Inflation Worksheet. This creates an inflation factors worksheet from the active project, and saves it with the same name and in the same directory as your Microsoft Project file, but with the worksheet extension .XLS.
4. Fill in the rate changes and dates for each resource.

**To calculate the inflated resource rates:**

- From the Data menu, choose Calculate Rates. This calculates the inflated resource rates. For example, you can use this command to reflect biannual raises by setting the Project Start date to the date of the first raise period and then having the macro inflate the standard rate by a given percentage every 2 quarters (or 6 months). You can calculate rates for the selected resource only. The active cell can be anywhere within the inflation table for the resource you are interested in.

**To attach a rate table to a resource:**

- From the Data menu, choose Attach Rate Tables. This copies the inflation tables from the worksheet and links them to the object box for each resource in Microsoft Project. To print this information, edit the Resource report to show objects (by selecting the Objects check box in the Resource Report Definition dialog box), and then print the Resource report.

**To calculate inflated costs:**

- From the Data menu, choose Calculate Inflated Costs: This takes the inflated rate information from your worksheet and adds it to the values in the Fixed Cost fields in Microsoft Project. The actual resource costs and rates in Microsoft Project are never changed; rather, the additional cost is added to the fixed cost for each task.

**To restore the original costs:**

- From the Data menu, choose Restore Original Costs. Don't choose this command before choosing the Calculate Inflated Costs command or all data in the Fixed Cost fields will be lost. Do use this command before choosing Calculate Inflated Costs a second or later time to avoid having your fixed costs added to each time.

## How the INFLATE tools work and what you can change

Most of the calculation is done by the Microsoft Excel macro CalcInflFactors, which corresponds to the Calculate Inflated Costs command on the Data menu. It looks at each resource, and, within the resource, at each rate that you have defined for the resource. Then it looks at each task that has work being done on it while that rate is in effect and adjusts the fixed cost of each of those tasks to reflect the inflated cost of using the resource during that period.

For example, suppose you had a resource whose initial rate was $10/h and whose inflated rate was $12/h. If this resource is assigned to a 10-hour task after the inflated rate takes effect, the Fixed Cost field for the task would be increased by $20 to reflect the inflated rate.

The fixed cost of each task is saved in the Number5 field whenever you choose Calculate Inflated Costs from the Data menu. Before calculating inflated costs a second time, choose Restore Original Costs from the Data menu to restore the original fixed costs. In this way, you can calculate your project with inflated costs multiple times, without increasing your fixed costs each time. The Fixed Cost and Number5 columns are included in the Inflated Cost table. This table is part of INFLATE.MPV, which is merged with your views when you choose Attach Rate Tables or Calculate Inflated Costs from the Data menu.

No calculations are made on the Overtime Rate or Cost Per Use information you enter in your inflation worksheet. If you are interested in this information, you can modify the macro to include calculations for these fields. You can also include reports that import the inflated rate information, or you can create a tool bar button to start the macro automatically. For a sample of how to write a Microsoft Project macro or create a tool bar button to run INFLATE.XLA from Microsoft Project, see PROJXTAB.XLM later in this document. You can also change the macro to accept rates in a work unit other than hours.

# PROJPERT.XLM

## PERT Analysis for Microsoft Project using Microsoft Excel

PROJPERT.XLM shows how you can use Microsoft Excel to calculate durations using the PERT analysis method. The view file, PROJPERT.MPV, shows how macros written for other applications can be run directly from Microsoft Project.

Note: For the unmodified macro to work properly from within Microsoft Project, the directory containing Microsoft Excel must be on your path.

You may need to modify the PERT Calculate Now macro in Microsoft Project to give the full path of the PROJPERT.XLM file. If you installed Microsoft Project to a directory named WINPROJ and did not move the PROJPERT.XLM file, you will not need to modify the macro. To modify the PERT Calculate Now macro:

1. From the View menu, choose Define Views. Choose the Open button. Select PROJPERT.MPV. This file should be in the SAMPLES.DDE subdirectory of your Microsoft Project directory. Choose the OK button. In the Open View File dialog box, choose the Replace option button and choose the OK button.
2. From the Analysis menu, choose Define Macros.
3. In the Macros box, select PERT Calculate Now.
4. Choose the Edit button.
5. Find the comment line beginning with: //Ensure opening of PROJPERT.XLM macro
6. Find the line beginning with: .Command="[OPEN(""C:\WINPROJ\SAMPLES.DDE\EXCEL\PROJPERT.XLM"")]"
7. If the path to PROJPERT.XLM is not correct, change it. Leave all quotes and brackets unchanged.
8. Choose the OK button, and then choose the Close button.

### About the PERT Analysis Method

Many project managers are familiar with the PERT or network chart used to represent tasks and relationships in a project. You can view any project you create in Microsoft Project as a PERT Chart by selecting PERT Chart from the View menu.

The PERT method (which stands for Program Evaluation Review Technique) is also a scheduling method. PERT was developed by the United States Navy to help estimate task durations by combining three estimates -- best, worst, and expected task durations -- instead of one. These estimates, sometimes termed optimistic, pessimistic, and expected estimates, are combined using a weighted average into a single estimate for the task known as the PERT duration.

### To use PROJPERT.MPV:

1. Open the view file PROJPERT.MPV, and open any projects you want to analyze using the PERT tools. This view file changes the View menu, tool bar, and Macro menu (now called the Analysis menu).
2. Enter best, worst, and expected duration estimates for each task that is not a summary task. Choose the Gantt Chart or the PERT Entry Sheet from the View menu and enter estimates into the table provided.
   You can also enter estimates using the custom PERT Analysis Entry Form. To use the form, select the tasks that have the same duration estimates, and then click the form button at the right side of the tool bar. The values you type in the form

will be entered for all selected tasks.
When entering PERT durations, you do not need to enter a duration unit such as days, only the value.  PROJPERT.XLM assumes that the PERT durations are in the default duration units.
If you know the duration for some of your tasks and don't need to use the PERT method to estimate their duration, enter the same duration for each of the three estimates. You could also modify the Microsoft Excel macro to automatically enter a default duration or the original duration for tasks with no estimates.

3. To begin the PERT analysis calculation, click the PERT button on the tool bar or choose the PERT Calculate Now command from the Analysis menu.
   The PERT Analysis tool is written in the Microsoft Excel macro language. When you click the PERT button or choose PERT Calculate Now, a Microsoft Project macro starts Microsoft Excel and the PERT Analysis macro, which displays the PERT Calculations dialog box so you can select the appropriate options.
4. To calculate the PERT durations once, select the Once (this time only) option button.
   To set up a dynamic calculation so that every time you change a best, worst, or expected value in Microsoft Project, the duration estimate is recalculated, select the Dynamically (always update) option button.
5. If you also want to see schedules based on each of the best, worst, and expected durations, select the Calculate Best, Worst, Expected Schedules In Start/Finish Fields check box. The start and finish dates are placed in the Start1/Finish1, Start2/Finish2, and Start3/Finish3 fields in Microsoft Project. This calculation is not dynamic, so any time you change a value in a Best, Worst, or Expected field, you would need to click the PERT button or choose PERT Calculate Now to recalculate these schedules.
6. To change the options for the calculations, choose the Options button. In the PERT Calculation Options dialog box, you can change the weight you want assigned to best, worst, and expected estimates, and you can change the number of decimal places displayed for the duration. After changing the options, choose the OK button.
7. Choose the OK button to begin the calculations in Microsoft Excel. When the calculations are finished, the macro returns to Microsoft Project. The original durations have been replaced by the calculated PERT durations. If you wanted to save the original durations, you could modify the Microsoft Excel macro to save them in one of the unused Start/Finish fields.
8. To look at the best, worst, and expected schedules, display the Best Case Gantt, Expected Case Gantt, or Worst Case Gantt. To display these views, either click one of the "face" buttons on the tool bar or choose the view from the View menu. The smiling face compares the PERT schedule with the Best Case schedule; the frowning face displays the Worst Case schedule; the "neutral" face displays the Expected Case schedule.

### How PROJPERT tools works and what you can change

The Microsoft Excel macro creates a worksheet to hold PERT calculations each time you choose PERT Calculate Now. When you quit Microsoft Project and Microsoft Excel, you do not need to save these worksheets. The next time you run Microsoft Project, you can set up the PERT calculations again by opening PROJPERT.MPV and choosing the PERT Calculate Now command.

If you find you use PERT analysis often, you can merge the views, commands, form, and tool bar into your everyday Microsoft Project environment, using the VIEW.MPV file that is opened when you start Microsoft Project. To add these items to your VIEW.MPV file, open the PROJPERT.MPV file, and then select the Merge From (*viewname*) option button and the items you want to add to VIEW.MPV.

The PROJPERT tools calculate duration based on your best, worst, and expected estimates for tasks. In Microsoft Excel, you can add other calculations or features. For example, you could calculate:

- Standard deviations of the duration estimates. The standard deviation provides a range within which the task estimate is considered 65% probable. The chance of the duration being within two standard deviations is 93%. Microsoft Excel could also be used to chart this curve.

- Near-critical paths of the project. By looking at the slack values and their standard deviations, you can flag tasks that are not on the critical path but are close to it. Combined with the standard deviation calculations, this could be a powerful analysis of your project.

Any other analyses you choose could be done by Microsoft Excel in the same way. Import the Microsoft Project data using the Clipboard or DDE, make calculations or charts in Microsoft Excel, and then return the new data to Microsoft Project. Look at the macros in Microsoft Excel and Microsoft Project to get ideas about how to create your own.

## PROJXTAB.XLM
## Resources by Task or Tasks by Resource Report

When you run PROJXTAB.XLM, you create a "crosstab" report listing either each task followed by its resources, or each resource followed by its tasks. You can specify the work units, the timescale units, the period the report should cover, and what type of information to include, such as costs, work, availability, and so on. These are the commands available on the Format menu when the Resource Usage view is active. To isolate the appropriate information in Microsoft Project, the Resource Usage view is used, with the Selected Tasks Only command on the Format menu active.

PROJXTAB.XLM sets up an array of either all the tasks or all the resources, and then add all the assignments. It cycles through the arrays, using the data to copy information from the Resource Usage view.

For this macro to work correctly, you need to have a Task Sheet and the default Resource Usage view, with just the ID and Name columns in the table. Step 1 below opens the BACKUP.MPV view file to make sure you have these views. If you have not changed your views, you can skip step 1.

**To use PROJXTAB.XLM:**

1. From the View menu, choose Define Views. Choose the Open button. In the directory containing Microsoft Project, select the file BACKUP.MPV. In the Open View File dialog box, choose the Replace option button. Choose the OK button.
2. Open the project on which you want a crosstab report.
3. In Microsoft Excel, open PROJXTAB.XLM.
4. From the Macro menu, choose Run

5. Select the macro called MacroRun.
6. After you finish with the information and close the macro sheet, choose the No button so changes are not saved.

**To run PROJXTAB.XLM from within Microsoft Project:**

1. Create a one-line macro or button on the tool bar that looks like this:
**AppExecute .Command=[*path*\EXCEL *path*\PROJXTAB.XLM]**
Include the paths for both Microsoft Excel and the macro. Initially, the macro is installed in the SAMPLES.DDE subdirectory in your Microsoft Project directory.
2. In Microsoft Excel, change MacroRun to an Auto_Open macro. For information about how to do this, see your Microsoft Excel documentation.
3. When you want to run PROJXTAB, just run the macro or click the tool bar button created in step 1.

The formatting for the report is at the beginning and end of the macro. You can change the formatting to whatever you want. For example, you could format the numbers in the usage information to include a percent sign or add the work unit to the numbers. After creating the report, you could also add outlining if appropriate.

If your work week starts on Monday instead of Sunday, you'll need to change one cell in the Microsoft Excel macro. In cell B79, type **WEEKDAY(ReportStart)-2** to replace WEEKDAY(ReportStart)-1.

## TRACKING.XLM
## Tracking Resource Work on each Task

TRACKING.XLM is similar to PROJXTAB.XLM, except that it is focused on tracking resource work on a task-by-task basis. When you run TRACKING.XLM, you can view information for all resources or just one resource. To isolate the appropriate information in Microsoft Project, the Resource Usage view is used, with the Selected Tasks Only command on the Format menu active. After the data is in Microsoft Excel, the macro cycles through the Resource Usage information and blanks out any work that has not been done. In effect, you are creating a Resource Usage view that shows only actual work.

There are two parts to TRACKING.XLM: getting data from Microsoft Project and sending information back to Microsoft Project. In the first part, TRACKING.XLM gives you the option of getting the information for one resource or for all resources, in addition to the work units and timescale you want for the report.

TRACKING.XLM then sets up an array of all the resources and then all the assignments. It cycles through the arrays, using the data to copy information from the Resource Usage view.

For the second part, TRACKING.XLM creates a button to click when you want to update information in Microsoft Project. To do this, you enter the actual work in the "resource usage" part of the spreadsheet. Microsoft Excel keeps a running total of actual work in the column at the left. You should also update remaining work so that the total is correct. After you enter the updated work information, choose the Update button. Microsoft Excel sends the work and actual work totals to Microsoft Project for each resource on each task. Since only the total is sent to Microsoft Project, the information on the Resource Usage view in Microsoft Project will not necessarily look like the spreadsheet in Microsoft Excel, but the totals will be accurate.

For this macro to work correctly, you need to have a Task Sheet and the default Resource Usage view, with just the ID and Name columns in the table. Step 1 below opens the BACKUP.MPV view file to make sure you have these views. If you have not changed your views, you can skip step 1.

**To use TRACKING.XLM:**

1. From the View menu, choose Define Views. Choose the Open button. In the directory containing Microsoft Project, select the file BACKUP.MPV. In the Open View File dialog box, choose the Replace option button. Choose the OK button.
2. Open the project on which you want a tracking report.
3. In Microsoft Excel, open TRACKING.XLM.
4. From the Macro menu, choose Run
5. Select the macro called MacroRun.
6. After you finish with the information and close the macro sheet, choose the No button so changes are not saved.

**To update information in Microsoft Project:**

1. In the timescale portion of the worksheet, enter the actual information.
2. In the Remaining Work column, enter the remaining work information.
3. Click the Update button. Work and remaining work are copied into Microsoft Project.

Note: If you are using a non-English version of Microsoft Excel 3.x, you need to do the following: Unprotect the worksheet by choosing Unprotect Document from the Options menu. Hold down CTRL and click the Update button. From the Macro menu, choose Assign To Object. In the Reference box, replace the current information with **tracking.xlm!B264**

**To run TRACKING.XLM from within Microsoft Project:**

1. Create a one-line macro or button on the tool bar that looks like this:
**AppExecute .Command=[*path*\EXCEL *path*\TRACKING.XLM]**
Include the paths for both Microsoft Excel and the macro. Initially, the macro is installed in the SAMPLES.DDE directory in your Microsoft Project directory.
2. In Microsoft Excel, change MacroRun to an Auto_Open macro. For information about how to do this, see your Microsoft Excel documentation.
3. When you want to run TRACKING.XLM, just run the macro or click the tool bar button created in step 1.

The formatting of the report is at the beginning and end of the macro. Because the macro protects the document it created at the end to protect the formulas used for calculating work and actual work, you can't change the formatting until you remove the protection. You can easily unprotect the document, or remove that section from the macro. You can then change the formatting to whatever you want. For example, you could format the numbers in the usage information to add the work unit to the numbers. After creating the report, you could also add outlining if appropriate. You may also want to unfreeze the panes if they are too big.

If your work week starts on Monday instead of Sunday, you'll need to change one cell in the Microsoft Excel macro. In cell B98, type **WEEKDAY(ProjStart)-2** to replace WEEKDAY(ProjStart)-1.

# Visual Basic Macros

Two Microsoft Visual Basic applications are included with Microsoft Project:

- ANALYZER.EXE finds and displays the task most behind schedule, the resource most over budget, and the resource most overworked in an open project.
- LATEDATE.EXE helps you determine the cause of the Microsoft Project message that a task cannot be completed in the time allocated.

These applications are located in the VB subdirectory of the SAMPLES.DDE subdirectory in your working Microsoft Project directory.

You do not need Visual Basic to run either of these applications. However, if you want to view or modify the code used to create the applications, you do need Visual Basic. The comments in the source code will help you understand the code so you can write your own Visual Basic applications to work with Microsoft Project.

## ANALYZER.EXE

### Highlighting Information from Microsoft Project through DDE

When you run ANALYZER.EXE, it looks at one of your open projects and finds the task most behind schedule, the resource most over budget, and the resource most overworked. It can also show all slipped tasks and over budget and overworked resources. ANALYZER.EXE works by retrieving information from Microsoft Project using DDE requests and DDE execute statements.

For this application to work correctly, you need to have several of the original views available. Step 1 below opens the BACKUP.MPV view file to make sure you have these views. If you have not changed your views, you can skip step 1.

**To use ANALYZER.EXE:**

1. From the View menu, choose Define Views. Choose the Open button. In the directory containing Microsoft Project, select the file BACKUP.MPV. In the Open View File dialog box, select the Replace option button. Choose the OK button.
2. Open the project files you want to analyze.
3. Switch to the Program Manager or the File Manager, and then choose Run from the File menu.
4. In the Command Line box, type ***path*\analyzer.exe**

For example, if ANALYZER.EXE is in the VB subdirectory of the SAMPLES.DDE subdirectory of the WINPROJ directory, type **c:\winproj\samples.dde\vb\analyzer.exe**

5. To select a project to analyze, choose the OK button in the first window.
6. Select the project you want to analyze, and then choose the OK button.
7. To see a complete list of tasks or resources listed in the information boxes, click the appropriate More button.
8. To select another project to analyze, choose Change Project from the File menu and return to step 5.
9. To get the most recent information from your project, choose Update Information from the File menu.

**To quit ANALYZER.EXE:**

- From the File menu, choose Exit.

If you run this application often, you can create an icon for it in the Program Manager. To create an icon, choose New from the Program Manager File menu, and type the complete path to the ANALYZER.EXE file.

Using ANALYZER.EXE as a starting point, there are many changes you could make to retrieve exactly the information you want. For example, you could change ANALYZER.EXE to retrieve information other than text, or change it to ask what type of information to retrieve each time you run the application.

### LATEDATE.EXE
#### Resolving Task Conflicts in Microsoft Project

When scheduling tasks in Microsoft Project, a message sometimes appears, telling you there is not enough time to complete a given task in the time allocated. This occurs whenever a successor of a task must start too soon to allow all work to be completed on a task. This might happen if a successor task has either an actual start date or a "must" constraint such as Must Start On. For more information about how this type of conflict might arise, choose the Help button available when the message appears, or see the *Microsoft Project User's Reference*.

LATEDATE.EXE will help you determine the tasks that create this sort of conflict. When the message appears, it gives you the ID of one of the conflicting tasks. When you run LATEDATE.EXE, you enter the ID number from the message to display a list of tasks that could be creating the conflict.

**To use LATEDATE.EXE:**

1. When the message warning of a task-scheduling conflict appears, note the ID number in the message, and choose the OK button.
2. Switch to the Program Manager or the File Manager, and then choose Run from the File menu.
3. In the Command Line box, type ***path*\latedate.exe**

For example, if LATEDATE.EXE is in the VB subdirectory of the SAMPLES.DDE subdirectory of the WINPROJ directory, type **c:\winproj\samples.dde\vb\latedate.exe**

4. Type the ID you noted in step 1.
5. In the Notepad window, scroll to the bottom of the document. Below the last line of asterisks with a date and time is a list of conflicts that could be causing the message to appear.

## Microsoft Word Macro

One Microsoft Word macro, called PRJTABLE.DOT, is included with Microsoft Project. To run this macro, you must have a copy of Microsoft Word, version 2.0 or later. You also need Microsoft Word to view or modify the macro. The comments in the macros will help you understand how Microsoft Word and Microsoft Project can use DDE to work together.

This template is located in the WORD subdirectory of the SAMPLES.DDE subdirectory in your working Microsoft Project directory.

## PRJTABLE.DOT

### Creating a Table in Word from Data in Microsoft Project

PRJTABLE.DOT is a Microsoft Word template. This template contains a macro called CreateTableFromProject. The macro creates a table in Microsoft Word filled with Microsoft Project data. In Microsoft Project, you select the data you want in the table, and then run the macro from Word. Each selected field in Microsoft Project is copied and then pasted and linked into a cell in the table in Word. Because the information is dynamically linked to Microsoft Project, the table in Word is updated when you change information in Microsoft Project.

**To use PRJTABLE.DOT:**

1. In Microsoft Project, select the fields you want in a table in a Word document.
2. Switch to Microsoft Word, and then open PRJTABLE.DOT.
3. From the Tools menu, choose Macro.
4. Select CreateTableFromProject.
5. Choose the Run button.

Once the table exists, you can change the formatting of the table so that it works in your document.