EndUser SharePoint.com

SharePoint 101

Tricks and Traps

EndUserSharePoint.com
No GeekSpeak on SharePoint 2007

Taming the Elusive Calculated Column

SharePoint 101: Tricks and Traps

Dessie Lunsford, EndUserSharePoint.com

SHAREPOINT 101: TRICKS AND TRAPS

Dessie Lunsford is writing a series of articles on EndUserSharePoint.com to show how to best utilize the Calculated Column. This book is the first set of articles in that series.

Mark Miller, August 2008
EndUserSharePoint.com

DESSIE LUNSFORD

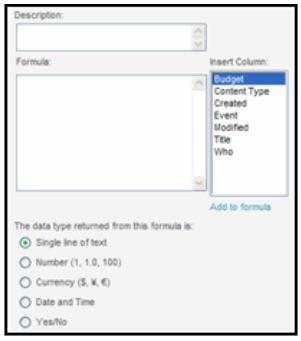
"I enjoy figuring out how to get my users to understand what they can do with SharePoint, and "how to do it" – this is probably the biggest hurdle my users have to face.

"Suggestions for future posts on calculated columns are always welcome, and if fact are encouraged.

Some of the best scenarios to illustrate are the "real-world" problems that we each face day to day, so if you have an example, an idea you want to explore, or a "**Can this be done with a Calculated Column?**" question that I can use as the topic of a future post, please submit it as a comment below and I'll see what I can do to work up a post covering it. "

Dessie Lunsform, August 2008
EndUserSharePoint.com

TAMING THE ELUSIVE "CALCULATED COLUMN"



JUNE 12, 2008

Filed Under Calculated Column, Dessie Lunsford, Tips and Tricks

View the Comments on this Article:
http://www.endusersharepoint.com/?p=362

When filling in a form, uploading a document, adding in an Announcement, or simply creating a list to house your favorite movies, you'll generally be working with multiple columns of information that may include required and/or optional fields, as well as several columns of "custom" information that you may want to collect.

The setup for this is pretty straight-forward (add new / modify existing columns, etc.)…but what if your requirement is to automatically add in a preset expiration date 2 weeks in the future, or make sure that a "phone" field displays numbers in a specific format, or perform a complex mathematical formula that displays a daily pick of Lottery numbers based on the user's birthday in relation to the autumnal equinox (still working on this one)…

ENTER THE "CALCULATED COLUMN"…

In this initial introduction of Calculated Columns, I want to mention some information available on the web as well as citing portions of it here for reference.

On the Microsoft Office site, they list out a reference to the types of Calculations possible in SharePoint:
(http://office.microsoft.com/en-us/sharepointtechnology/CH100650061033.aspx)

- Date and Time
- Financial
- Information
- Logical
- Lookup and Reference
- Math and Trigonometry

- Statistical
- Text and Data

Additionally, they list a reference on commonly used functions to get you familiar with examples of what you can do:

(http://office.microsoft.com/en-us/sharepointtechnology/HA011609471033.aspx)

- Conditional Formulas
- Date and Time Formulas
- Math Formulas
- Text Formulas
- Other Formulas

As we get into the depth of what can be accomplished through Calculations, you'll no doubt come across roadblocks and miscellaneous error messages stating something isn't supported, or that you can't perform the function on the data type, or other similar problems. To (hopefully) get around many of these, I'll be making heavy use of Microsoft Excel as the "test bed" for most of the formulas I'll be demonstrating.

The reasoning behind this is simple - Excel provides a straight-forward interface that works as a great tool for modeling formulas…with the added in bonus of built-in error checking (something SharePoint lacks), and generally (not always), if you can get it to work in Excel, you can get it to work in SharePoint.

$$s_{a+c} = k\left(\frac{\Delta t}{2}\right)^3 + 2v_0\frac{\Delta t}{2}$$

$$s_{a+c} = (v_t - v_0)\frac{\Delta t}{2} + 2v_0\frac{\Delta t}{2}$$

$$s_{a+c} = (v_t + v_0)\frac{\Delta t}{2}$$

Obviously, there are many formulas available in Excel that will not translate since they have nothing to do with the data types available in a SharePoint list, but again, as a "test bed" its one of the best tools available for working out the logic that complicated formulas can present - so once we get the formula laid out, we'll then go over how to "translate" it into a compatible "SharePoint" formula.

"*In future posts in this series I will be diving into the power of Calculations and just how to accomplish many different tasks that you may come across through sample exercises and formulas, as well as (hopefully) some real-world scenario's suggested by the readers. Till next time…"* – Dessie

TAMING THE ELUSIVE "CALCULATED COLUMN" - CUSTOMIZING A TASK LIST (PART 1)

JUNE 19, 2008

Filed Under Calculated Column, Dessie Lunsford, Libraries and Lists, Tips and Tricks

View the Comments on this Article:
http://www.endusersharepoint.com/?p=410

A good place to start with learning how to work with calculated columns is to start with an existing list and see what additional functionality you can add to give it more "value" (and make it worth using).

In this walkthrough, we'll be looking at the default OOTB ("Out of the Box" for the non-acronym friendly) "Tasks" list. I want to start with this particular list because I've always felt that it could use some help since in it's default state, it's almost a bit "too" basic to really be used effectively.

Yes, it does allow you to assign items to individuals and groups and give them, what I like to call "suggested", "Due Date" on its completion, but what if it takes them longer to complete the task? What if they have a large number of tasks assigned...how do you set up a decent view to display and prioritize them (overdue items, upcoming due dates, number of days left to complete, etc.)? How can you go back and audit their performance (Do they normally complete tasks on time?)?

We're going to add in 5 additional calculated columns that will allow us to setup more complex filters and views to make accountability and tracking of pending and completed tasks more useful and available.

The 5 columns we're going to create are the following:

1. "**Date Completed**" - This one is all about archiving and finding work history, or how much work users have been able to accomplish - useful for custom views and filtering.

2. "**How many days left to complete?**" - Another column that will aid in custom views and filtering to assist in time management of upcoming task "Due Dates".

3. "**Days Overdue**" - Advanced version of the "How many days left to complete?" column that takes in account for after the task has been completed (and doesn't use any negative numbers).

4. "**Completed Early?**" - Who doesn't like bonuses for work completed ahead of schedule? This one can be useful in auditing past performance or trends.

5. "**Overdue?**" - Visually used to create a custom view that at-a-glance will let users know if they have Tasks that are overdue.

I encourage you to read through the "**Introduction to data calculations**" article on the MS Office site. Although basic in its examples, it does contain some much needed information on some of the particulars in working with calculated columns.

As mentioned in my previous article, **Taming the Elusive Calculated Column**, we're going to be designing these in MS Excel to take advantage of error checking and design ease-ability, then convert them to a "SharePoint"-friendly formula.

As a pre-cursor, I want to mention that we'll be working with a few different data types that are all classed as a "Constant" type, data that does not vary:

1. **Strings** - Strings (text field) can contain a set of up to 255 characters and are enclosed in a set of quotes (""). In our examples we'll only be using simple strings, but in reality a string could contain anywhere from and empty value ("") and up (i.e. "John", "John Smith", "John Smith is a person", etc.).

2. **DateTime** - a set of characters that represents a specific date and/or time. In SharePoint, dates are stored as sequential numbers starting with the serial number 1 representing December 31, 1899. In the example of January 1, 2008, the serial number would be 39448 because it is 39,448 days after January 1, 1900 (this isn't too important to remember now, but will make sense later as we convert our formulas into compatibility for SharePoint and want the dates displayed in a meaningful manner).

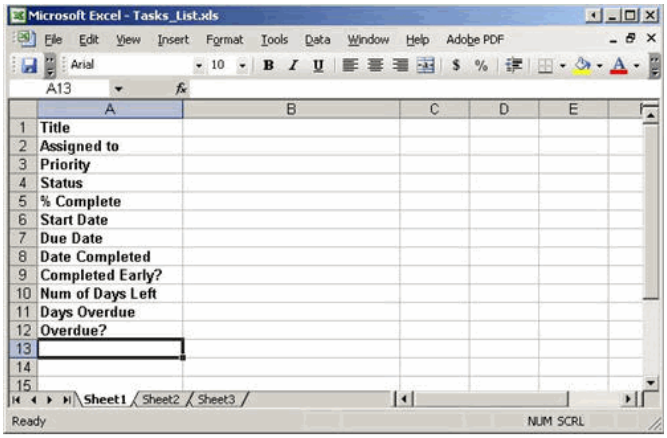
3. **Numbers** - any range of numbers (can be positive or negative and contain decimal places).

4. **Boolean** - an evaluation that returns either a "Yes" or "No" value (can be written to display a specific value if "Yes" or something completely different if "No").

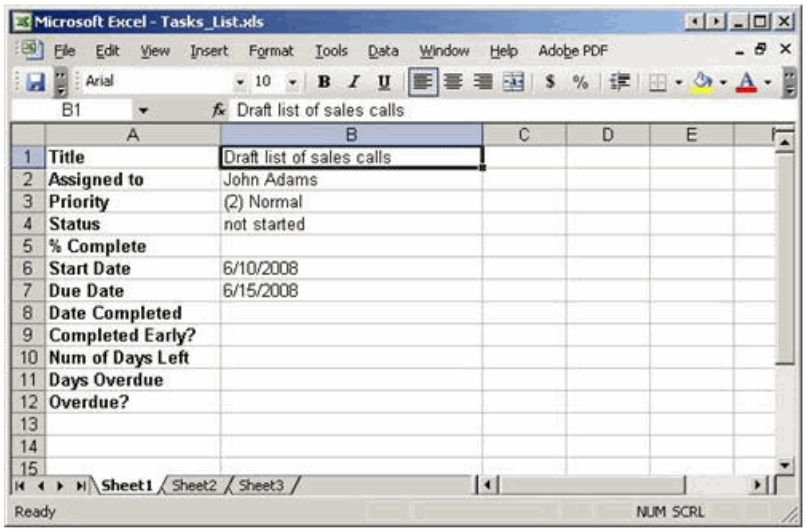
To begin, open Excel and enter the following in cells A1 through A12:

- Title
- Assigned to
- Priority
- Status
- % Complete
- Start Date
- Due Date
- Date Completed
- Completed Early?
- Num of Days Left
- Days Overdue
- Overdue?

This sets up a representation of the "Tasks" list with placeholders for our soon-to-be created columns.

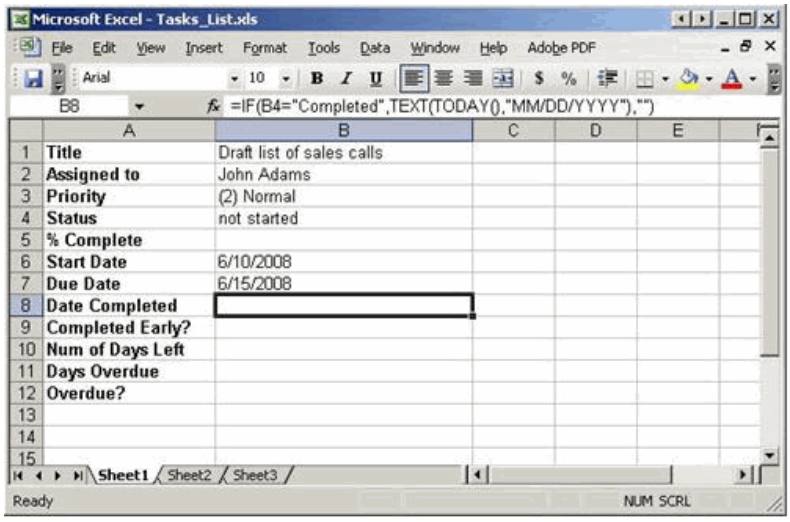


Enter in some sample information in the "Title" through "Due Date" fields (B1 through B7):



To create the formula for the first custom column ("Date Completed"), we need a way to first find out if the task is actually completed. If it is, we'll then grab the date and display it in the format we want. For this we'll use the "IF" function as follows:

Formula:
=IF(B4="Completed",TEXT(TODAY(),"MM/DD/YYYY"),"")



In the above formula, we're checking to see if the "Status" field (B4) equals "Completed". If the result of this is "True" (value of the cell is "Completed"; this is a "Boolean" check), we grab today's date, convert it from "DateTime" to "text", and then display it in the format of "MM/DD/YYYY" (06/16/2008). If the result is "False", we don't display anything.

Dissecting the formula, notice how it's structured and grouped with parentheses:
=IF(B4="Completed",TEXT(TODAY(),"MM/DD/YYYY"),"")

Starting from the left, we have an "equals" sign (=) to indicate that this calculated column will have a value that is based on the formula that follows it.

We then encounter our first function (the "IF"). This tells us that we're performing a check on some data that produces a result based on whether a specific condition is "True" or "False".

After the "IF", we have our opening parenthesis that signifies the start of our condition to evaluate, which is then followed by the actual condition. The condition we're evaluating is "Does the value in cell B4 equal the text "Completed"?

If the value in B4 does equal "Completed", the result is "True", and in this case, for the "True" result, we're grabbing today's date (represented by the function "TODAY()"), converting it from a "Date" type into a "Text" type (string) then finally forcing it appear in the specific "**DateTime**" format we want for it to display in. Additionally, you'll notice that there is a new set of parenthesis surrounding the "TODAY(),"MM/DD/YYYY"" part. This is because the "**TEXT**" function has two required parameters when it is used; a **value to convert** and a text format for it to be displayed in.

Finally, we have an empty set of quotes ("") that are used for the "False" result, followed by the closing parenthesis of the "IF" check.

Breaking this down even more, to really understand this type of formula, the "IF" function contains three parts.

1. The logical condition to check ("Does the value in cell B4 equal the text "Completed"?)

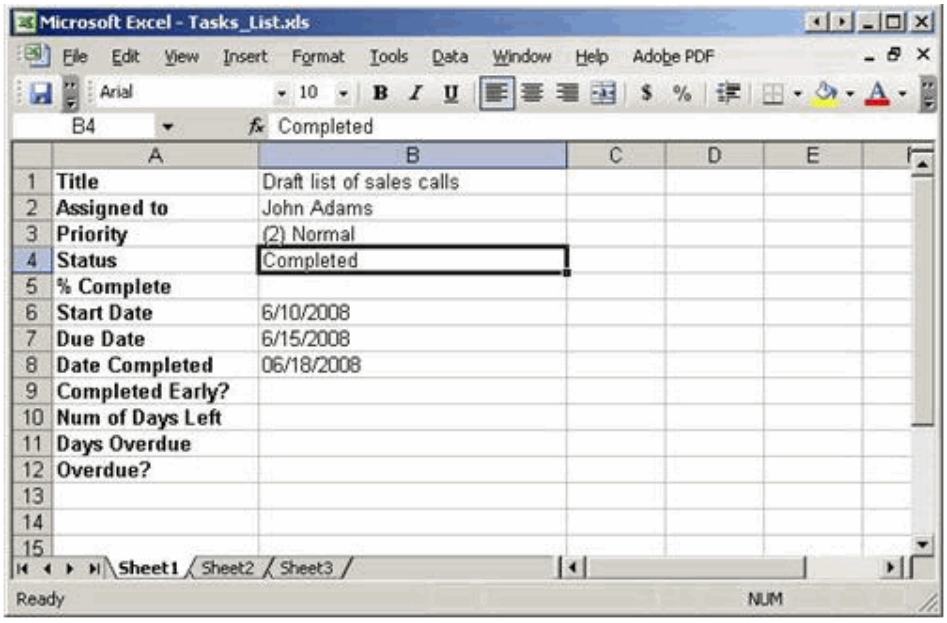
2. Value if "True" (Display the current date)

3. Value if "False" (Display nothing)

So, with our formula:

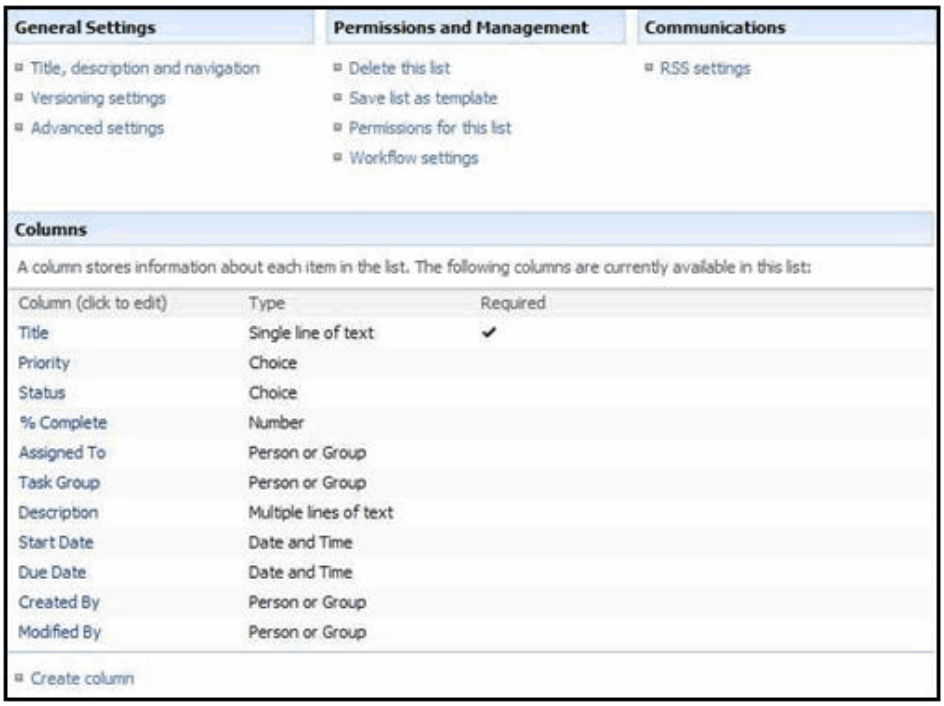
1. The logical condition to check: B4="Completed"
2. Value if "True": TEXT(TODAY(),"MM/DD/YYYY")
3. Value if "False": ""

To test this new column, enter "Completed" into cell B4 and you should see today's date appear in cell B8.



Now that we have the formula working in Excel, how do we now move it into the actual "Tasks" list in SharePoint? Actually, it's pretty simple, but there are a couple steps that we do need to take to get past some of the errors I mentioned last time.

In SharePoint, go in to the "Tasks" list and its settings page.



General Settings

- Title, description and navigation
- Versioning settings
- Advanced settings

Permissions and Management

- Delete this list
- Save list as template
- Permissions for this list
- Workflow settings

Communications

- RSS settings

Columns

A column stores information about each item in the list. The following columns are currently available in this list:

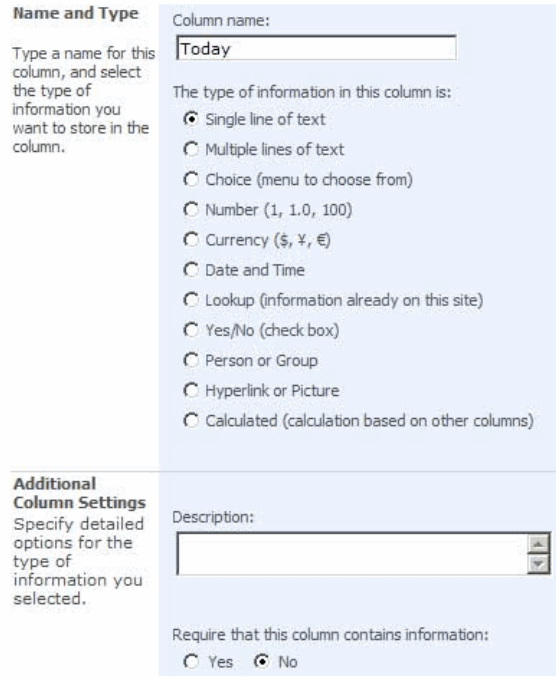
Column (click to edit)	Type	Required
Title	Single line of text	✔
Priority	Choice	
Status	Choice	
% Complete	Number	
Assigned To	Person or Group	
Task Group	Person or Group	
Description	Multiple lines of text	
Start Date	Date and Time	
Due Date	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

- Create column

We could simply just copy/paste the Excel formula into a new calculated column and modify it to use the appropriate column name instead of the cell we specified in Excel, but if left as is; it would produce an error message stating "Calculated columns cannot contain volatile functions like Today and Me". Although this is true (somewhat), we can get around this by instead using a common approach of "fooling" the system to allow us to use the "Today" reference.

To do this, we simply create a new column called "Today" leaving all of its options as default. We can then proceed with our formula and not get the error message (neat huh?).

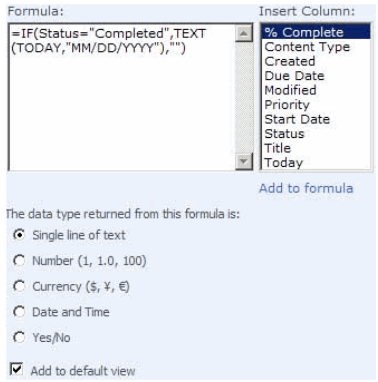
So, moving forward, create a column called "Today".



Next, create a column called "Date Completed". Make it a "Calculated" type and paste in our formula from Excel. Before saving, replace "B4" with "Status", remove the parenthesis after "TODAY" () and leave the rest as default (after saving, the system will automatically change the "case" of the "TODAY" field to "Today" - **column names are not case-sensitive in our formulas**).

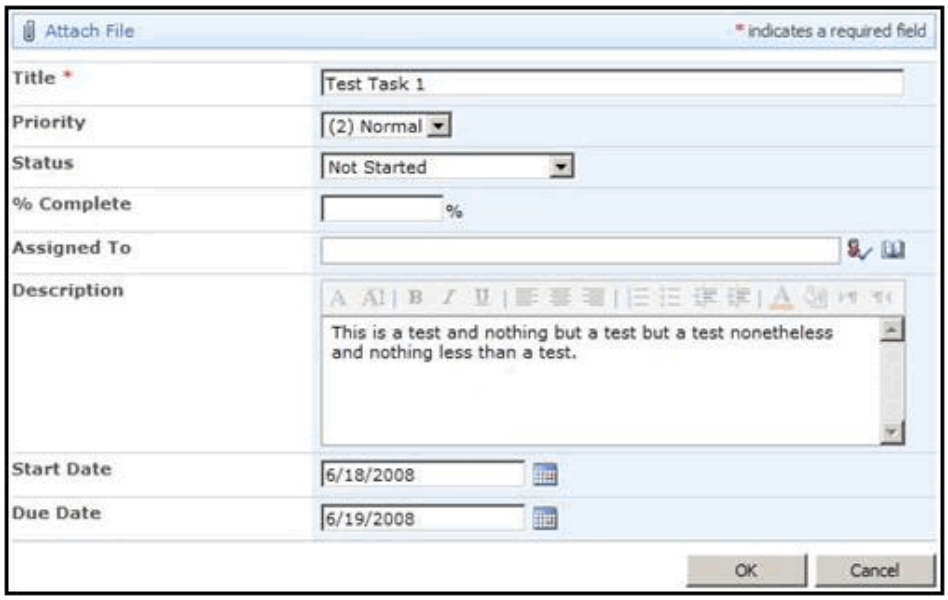
Formula in Excel:
=IF(B4="Completed",TEXT(TODAY(),"MM/DD/YYYY"),"")

Formula in SharePoint:
=IF(Status="Completed",TEXT(TODAY,"MM/DD/YYYY"),"")

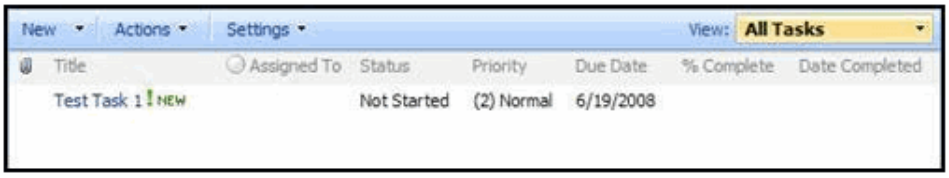


Go back into the "Today" column and delete it. This column was simply a placeholder that fools the system into allowing us to use the name for our formula, so deleting it will now match it up with the real "Today" data in the system, but without throwing an error (**read more on this here**).

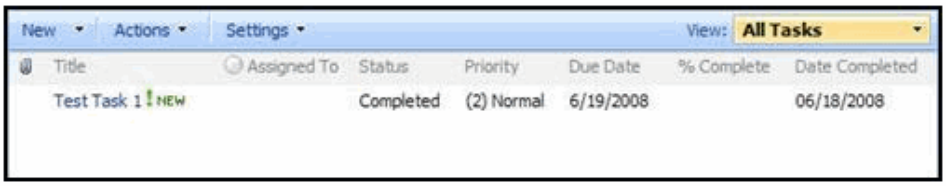
To test our new column to make sure all is working, simply create a new item in the list with a "Start Date" of the (default) current date, and a "Due Date" of tomorrow.



Once saved, we can see our new item:



To now test our new column, go back in and edit the item and change its Status to "Completed". If everything was done correctly and our formula was written right, we should now see a completed date populated with today's date.



So what we now have is a way to track when items were completed and use this information in a custom View that can be filtered and/or grouped, by our new column.

One caveat to consider in using this type of formula, is that because we're using a "Today" reference to set the value of the "Date Completed", we're subjecting ourselves to the possibility of inaccurate data if someone were to later (some date after the actual completed date) go in and edit any details of the task item. This happens because in this case "Today", literally means the current day's date, so if you were to make an update later, the value of the date is re-calculated to use that day.

Will the current column we just created work? Yes.

Is it the "best" approach to capturing the "Date Completed"? Probably not, but we'll remedy that later.

Was there value in running through the process of creating the column even though it's not the best approach to take? I think so (and I hope you did too).

None of the above however, means that our formula is a complete waste, if anything it was good practice for the other columns we're going to be creating (three of the remaining four all use the "Today" column in their formulas). As we move forward, we'll probably just replace this particular column with a standard "Date" field where a user can pick the completed date from the calendar popup.

“In the next post I'll be continuing on with the "Tasks" list and adding in the next two columns that will let us know how many days we have left to complete a task ("How many days left to complete" and "Days Overdue").”

Till next time...
- Dessie

ENDUSERSHAREPOINT.COM: TAMING THE ELUSIVE "CALCULATED COLUMN" - CUSTOMIZING A TASK LIST (PART 2)

JUNE 27, 2008

Filed Under Calculated Column, Content Type, Dessie Lunsford, Libraries and Lists, Tips and Tricks

View the Comments on this Article:
http://www.endusersharepoint.com/?p=446

If you haven't read through the previous article in this series and wish to follow along in the walkthrough, I encourage you to read it first, especially since we'll be working with the Excel spreadsheet we created last time.

In the previous article on "**Customizing a Task List**", we created a "Date Completed" column using a reference to "Today" in its formula. In order to accomplish this and not get one of the dreaded error messages that SharePoint likes to confuse it's users with, we created a temporary "Today" column which allowed us to build the formula and run it. Afterwards, we simply deleted the temporary column to "finalize" our calculation.

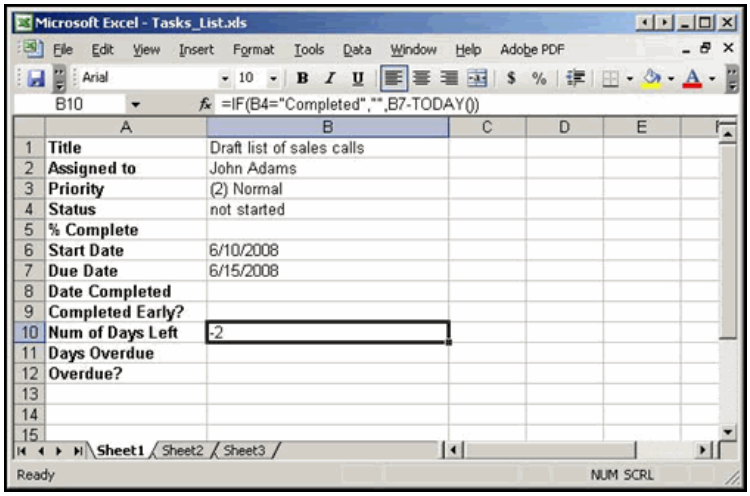
One thing to note when using the idea of this "temporary" column is that whenever there is a need to go back in and edit the formula that uses "Today", we will always have to re-create the temporary column first. This is needed for the exact same reason we created it in the first place - to fool the system into letting us use the name as a reference, and to bypass any of the errors that SharePoint will throw because of it.

Thanks to Adam Davidson for pointing this out in a comment from last time. I had intended to mention this in the previous article, but wound up leaving it out due to the length of the post.

Picking up where we left off, the next column we're going to create is our "How many days left to complete?" column. To shorten this a bit for the name of the column, we'll just be using "**Num of Days Left**". This column is designed to display a simple count of how many days we either have left, or are overdue - a positive number if "day's left", negative number if "days overdue".

Open up your Excel spreadsheet from last time and enter in the following formula for the new column:

Formula:
=IF(B4="Completed","",B7-TODAY())



This is a relatively simple formula in that our "IF" statement is simply looking to see if the "Status" field is equal to "Completed". If it is, nothing will be displayed. If it's not, then we take today's date and subtract it from the "Due Date" and display the difference (this is where we get the positive/negative number possibilities).

Also, and this is only for when we're working in Excel, format the result cell (B10) to be a "Number" with no Decimal places (right-click in the cell > choose "Format Cells").

This formula is very similar to our first calculated column formula ("Date Completed"), but adds in a date calculation to return the difference between two dates.

Looking at the details of this formula, we see the following parts:
=IF(B4="Completed","",B7-TODAY())

This literally means: "If the value of cell B4 equals the text "Completed", we're not going to do anything, but if it does, we're going to take the value of cell B7 ("Due Date") and subtract Today's date from it, then display the number of days (difference between the two) into our cell (B10)".

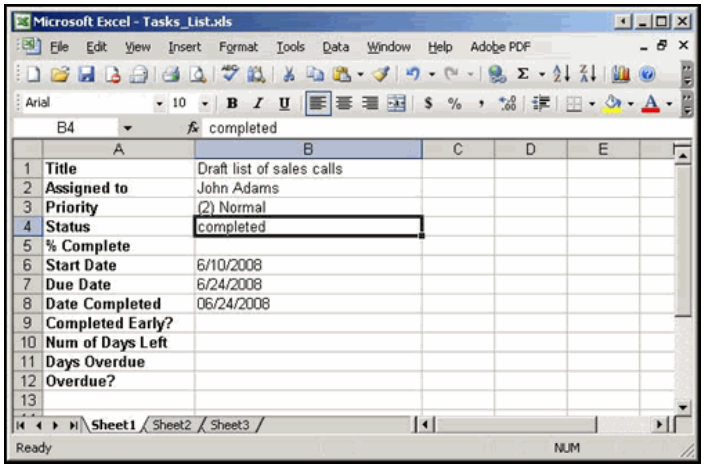
So, in this formula, the three parts of the "IF" function are as follows:

The logical condition to check (”Does the value in cell B4 equal the text “Completed”?)

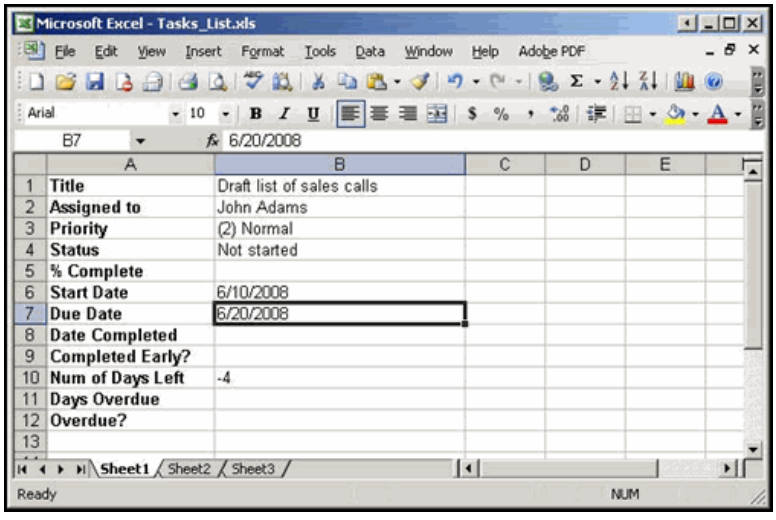
Value if “True” (Display nothing)

Value if “False” (Subtract Today’s date from the “Due Date” and display the number of days difference)

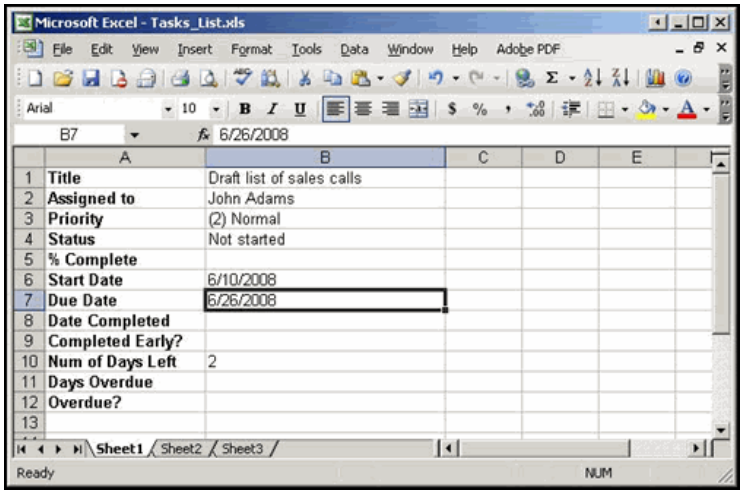
Test this by changing the value of cell B4 to “Completed” (nothing will be displayed in B10).



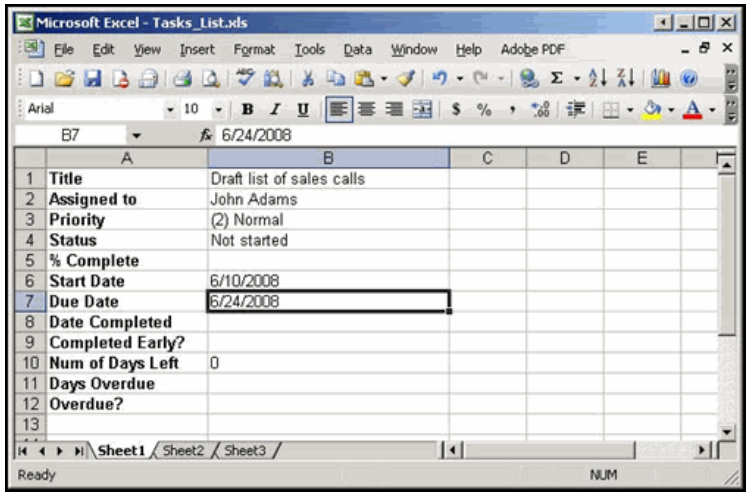
Test also with leaving B4 as “Not started” and modify the “Due Date” (B7) to be before today’s date (number displayed as negative letting us know how many days overdue).



Test again with the “Due Date” set past today’s date (number will be positive letting us know how many days we have left).

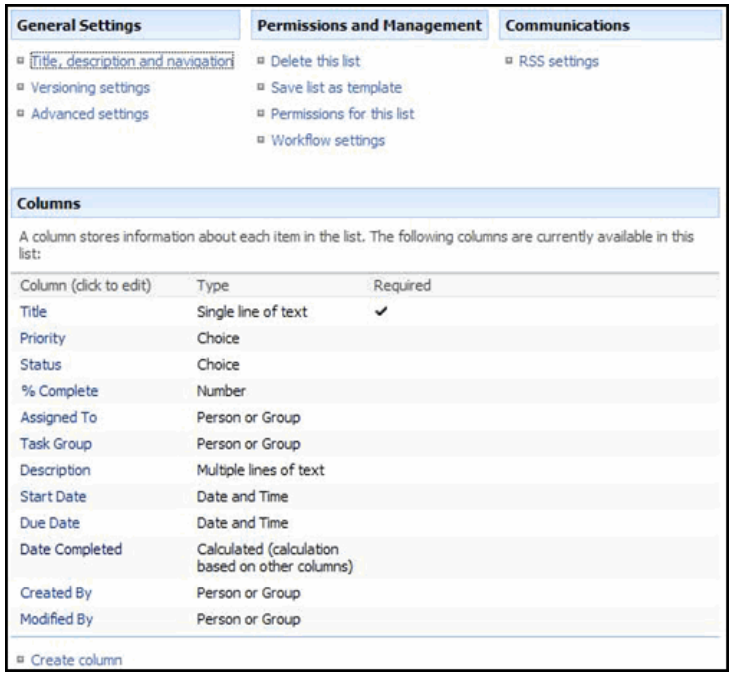


Test one final time with the “Due Date” set to today (a zero will be displayed).



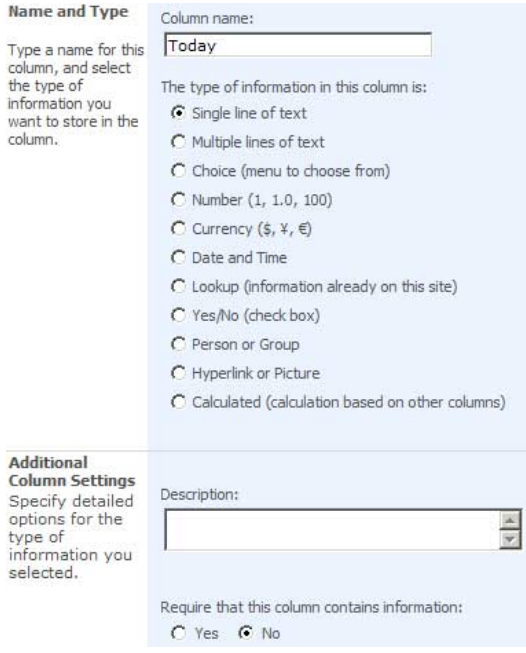
In order to get this into SharePoint, we run through the same steps as last time:

In SharePoint, go in to the “Tasks” list and its settings page.



Since we're again using a "Today" reference in our formula, we'll need to first create our temporary "Today" column to allow us to use it in our formula.

Create a new column called "Today", leaving the rest of its settings as default.

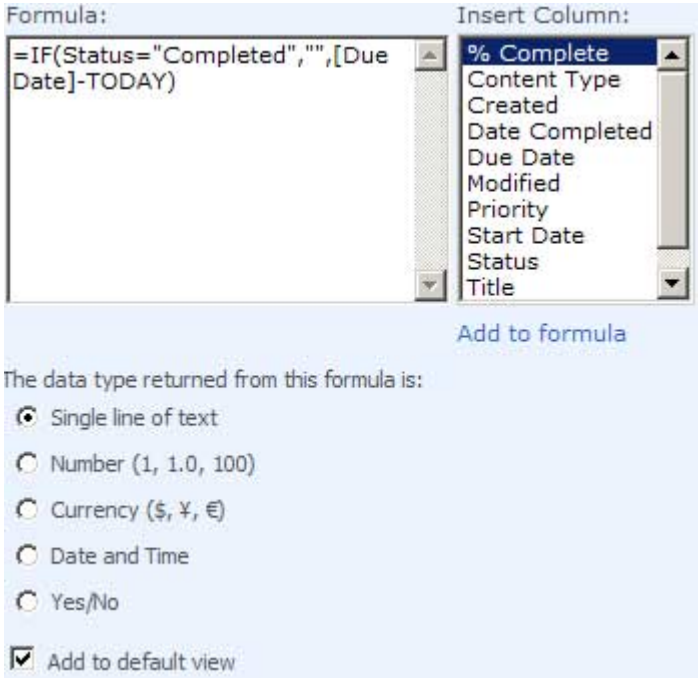


Create another new column called “Num of Days Left”, make it a “Calculated” type and paste in the formula from Excel. **Before saving**, replace “B4” with “Status”, “B7” with “[Due Date]“, and remove the parenthesis after “TODAY” ().

Also, **before saving**, notice the brackets [ ] around the “Due Date”. Whenever a space or special character is used in a column name, it has to be enclosed in brackets or it won’t be recognized as the correct name. The brackets tell the system that literally anything in between them is the name and to ignore characters that are normally used as part of an expression - e.g. “Pros & Cons” has to be written as “[Pros & Cons]” in the formula or it won’t be recognized as a column.

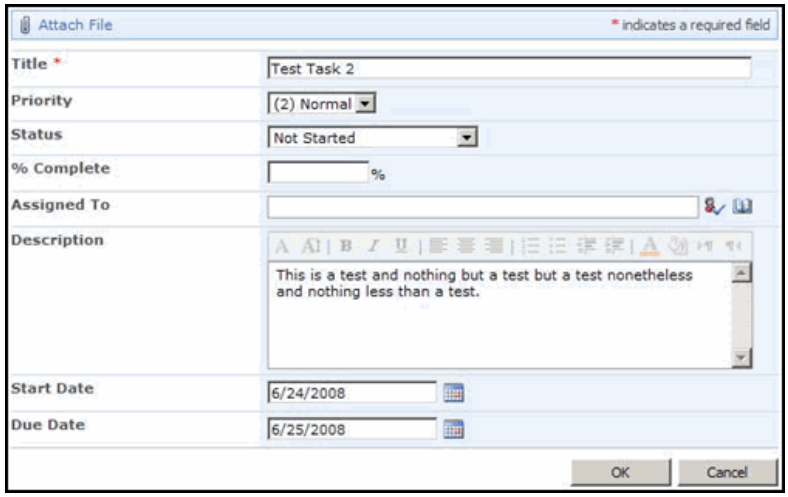
Formula in Excel:
=IF(B4=”Completed”,””,B7-TODAY())

Formula in SharePoint:
=IF(Status=”Completed”,””,[Due Date]-TODAY())

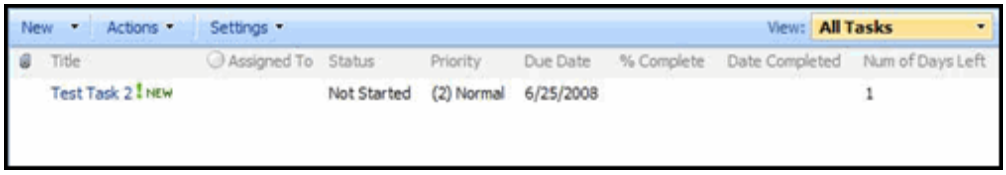


After you have the new calculated column, we need to go back and delete our temporary “Today” column. This column was simply a placeholder that fools the system into allowing us to use the name for our formula. Deleting it will now match it up with the real “Today” data in the system, but without throwing an error.

To test the new column, just as we did last time, create a new item in the list with a "Start Date" of the (default) current date, and a "Due Date" of tomorrow.

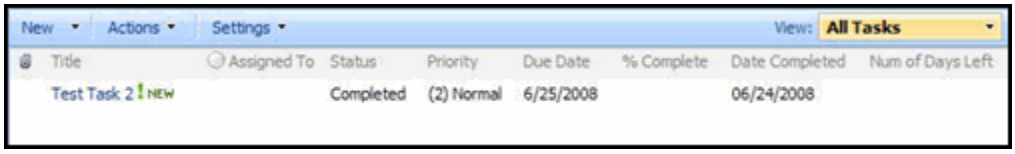


Once saved, we can see immediately that our new column has a number letting us know how many days left we have to complete the task:

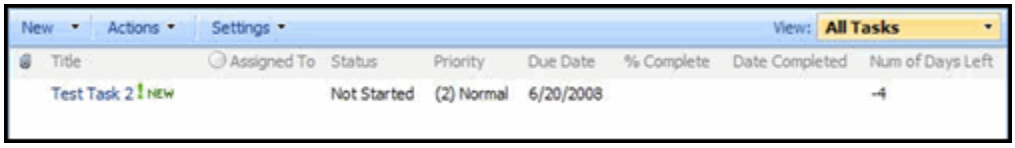


To test the various values that could be displayed, we'll just edit the task and see what the results are.

Set the task as "Completed" (note that nothing is displayed in the new column).



Adjust the "Start Date" and "Due Date" back several days (note how the value will display as a negative number since we're overdue).

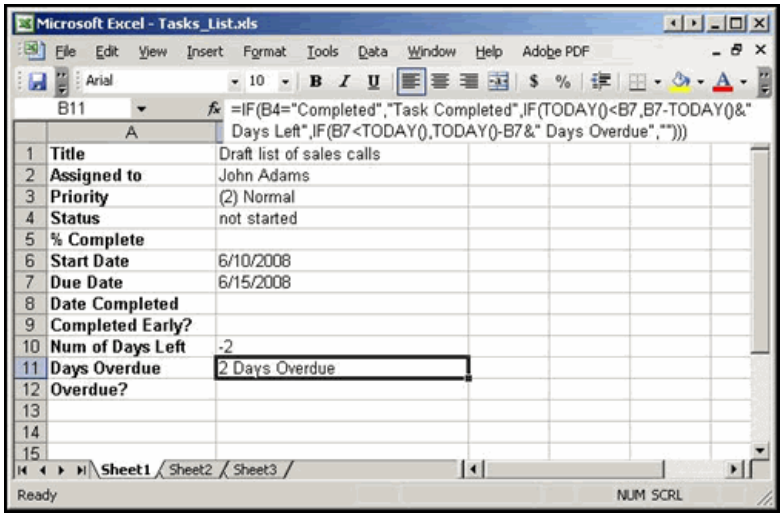


Ok, so now we have a way to see the date a task was completed, and if it hasn't yet been, then a way to show how many days we have left, or are overdue. This works, but probably isn't the most user-friendly way to display the numbers since once we're overdue the number displays as a negative.

This leads us to a modification, or "advanced" version of the "Num of Days Left" column we'll call the "Days Overdue" column. This column is an expansion of the "Num of Days Left" column that performs the same function, but includes additional logic to make the results more meaningful and displayed in a "user-friendly" format.

The formula begins the same with an initial check to see what the text is in the "Status" field, but immediately you'll notice it is very different (see the multiple "IF" functions?).

Formula:
=IF(B4="Completed","Task Completed",IF(TODAY()<B7,B7-TODAY()&" Days Left",IF(B7<TODAY(),TODAY()-B7&" Days Overdue","")))



In this formula, we're again running a check against the "Status" column to see what its value is.

If it's equal to "Completed", we display a result of "Task Completed".

If its not "Completed", we then run through two additional checks to see if we still have any days left to complete the task or if we're currently overdue.

If we have days left, we subtract "Today" from the "Due Date" and display the number of days left with the addition of some simple text letting us know that we have "x Days Left".

If we're currently overdue, we subtract the "Due Date" from "Today" (notice the swap in position of the two columns in the calculation?) and display the value as "x Days Overdue".

This form of calculation is known as an "**IF-ELSE**" chain, and can viewed as the following:

IF (condition to check)
{Do something if above is "true", if false, do nothing and move on}
Else IF (condition to check)
{Do something if above is "true", if false, do nothing and move on}
Else IF (condition to check)
{Do something if above is "true", if false, do nothing and move on}
Else
{Do Something - think of this as the "cleanup" or "catch-all" - "if none of the above, do this"}

In the processing of the formula, the first check that results in a value of "True" will terminate the remaining checks. So, in our example, if the value of the "Status" field is "Completed", no further checks are needed (neither of the remaining date comparisons happen). If the value is not "Completed", it moves to the next comparison and attempts to reach a value of "True". If "True", it then processes the result of "x Days Left", if "False", it moves on to the last comparison. If the final comparison is also "False", it will move on to the end and simply display nothing.

Since this is a more complicated formula, lets again examine each piece to get a more detailed view of how this type of formula is structured:

We're using "IF" statements again, so we know we have **three parts** to each "IF":

The logical condition to check

Value if "True"

Value if "False"

In order to achieve the chaining of "IF" statements, we will be substituting an "IF" in place of the "False" of the preceding "IF" statement in order to produce a result (stay with me).

So, to break this chain-of-checks down; let's look at it literally like this:

"If the value of the Status field equals "Completed", display the text "Task Completed" in the "Days Overdue" column. If it does not, is today's date less than the due date? If so, subtract today's date from the due date and display the number (difference between the two) in addition to the text "Days Left". If not, is the due date less than today? If so, subtract the due date from today and display the number (difference between the two) along with the text "Days Overdue". If not, don't display anything."

Since we're using a nesting-type of approach to this (it's not really "nesting" per say, but it does help to illustrate how the formula is structured), each of our "IF" statements three parts would look like this:

The logical condition to check (does the value of cell B4 equal "Completed"?)

Value if "True" (Display the text "Task Completed")

Value if "False" (start another "IF")

The logical condition to check (is today's date less than the "Due Date"?)

Value if "True" (Subtract today's date from the Due Date and display the number of days difference along with the text "Days Left")

Value if "False" (start another "IF")

The logical condition to check (is the Due Date less than today's date?)

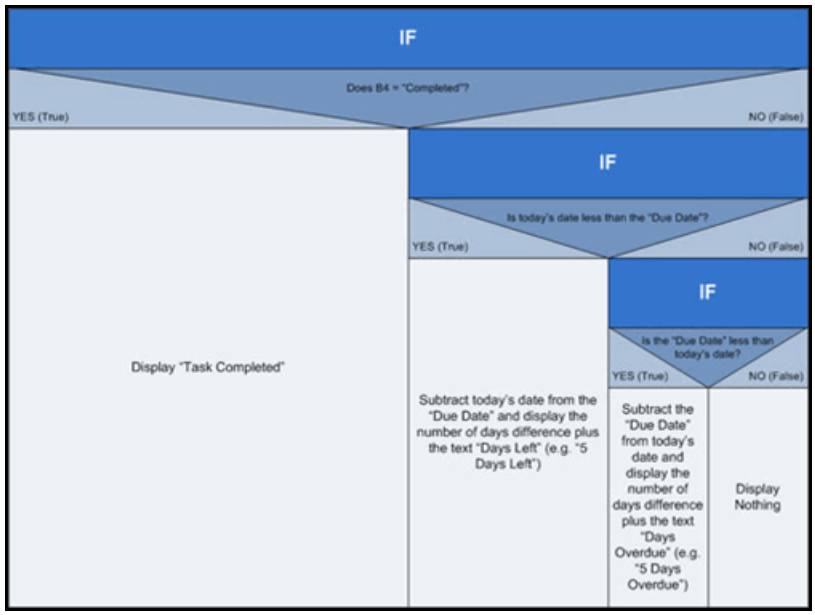
Value if "True" (Subtract the Due Date from today's date and display the number of days difference along with the text "Days Overdue")

Value if "False" (Don't display anything)

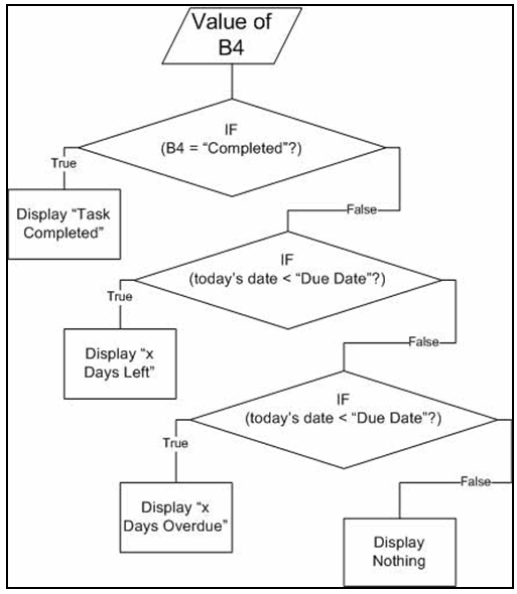
Each "False" result starts another "IF" statement. Each "True" result displays a value, and the final result (if not "true" in any of the others) will display nothing.

In this case, if our original "IF" statement's check returns a value of "False", it hands off the processing of the "False" result to another "IF" (the "ELSE-IF") which in turn does the same if it returns its own "False" for its check. Once each sub-process completes its task, it returns its result back to its parent until it reaches the "top" parent (original process) where it can display the final processed value for its "False" result.

Here are a couple of diagrams to illustrate this for those that like the "graphical" approach:



Standard "**Nassi-Shneiderman**" diagram used to "visually" walk through programming logic steps.



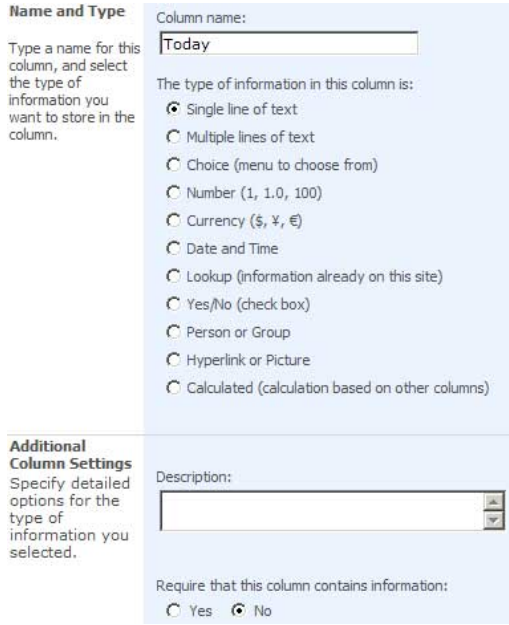
(Basic flowchart of "IF" function logical steps)

The end result of all of this is that if the original condition if “False”, it runs through two more checks until it reaches a final result, and then displays the value of that result.

To convert this into a compatible SharePoint formula we do the following in our “Tasks” list:

Once again, since we’re using a “Today” reference, we **first create** our temporary “Today” column to fool the system into letting us use the name.

Create a new column called “Today” leaving the rest of its settings as default.

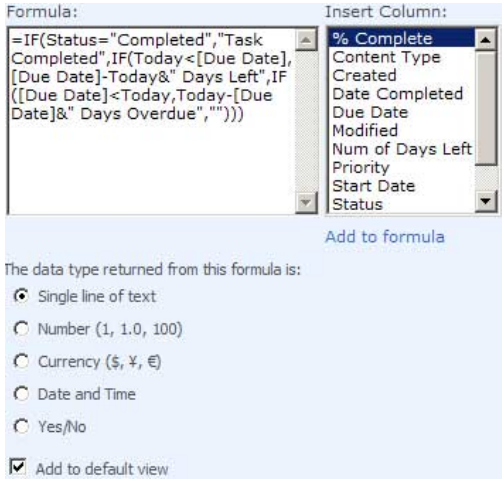


Create another new column called “Days Overdue”, make it a “Calculated” type and paste in our formula from Excel. **Before saving**, replace “B4” with “Status”, **each** “B7” with “[Due Date]” (noting the use of brackets), and remove the parenthesis after **each occurrence** of “TODAY” ().

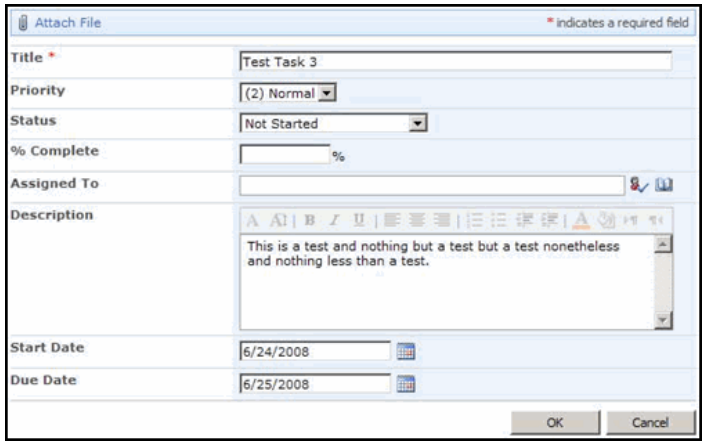
Excel Formula:
=IF(B4=”Completed”,”Task Completed”,IF(TODAY()<B7,B7-TODAY()&” Days Left”,IF(B7<TODAY(),TODAY()-B7&” Days Overdue”,””)))

SharePoint Formula:
=IF(Status=”Completed”,”Task Completed”,IF(Today<[Due Date],[Due Date]-

Today&" Days Left",IF([Due Date]<Today,Today-[Due Date]&" Days Overdue",""))))



To test this new column, create a new item on the task list with a (default) "Start Date" of today and a "Due Date" of tomorrow.

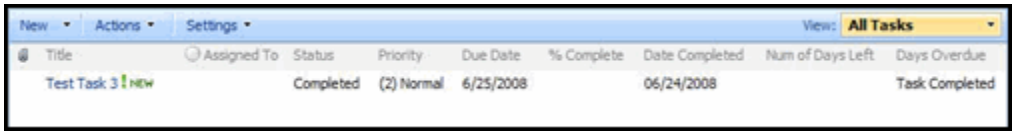


On saving, we'll now see:

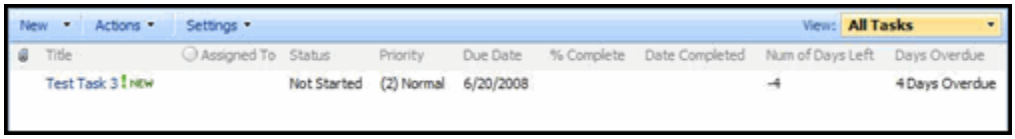
Notice how it displays the remaining days in a more meaningful manner ("1 Days Left")?

To test the various results that would display in each situation, we modify the task details.

Set the task as “Completed”. It will display “Task Completed” instead of a number.



Set the task back to “Not Started” and modify the “Start Date” and “Due Date” to several days back.



Notice how it now displays the day’s overdue amount without any negative numbers?

The main thing is that what we’re attempting to accomplish is to add in additional functionality to a default list that is pretty plain in its OOTB state. The first column we created in this article will do that exact thing, but doesn’t necessarily display information in the best format, so with the addition of a few extra lines in our formula that creates additional logic, we wind up with a second column (most likely to replace the first) that not only takes in account for each possible situation, but formats the results in a more user-friendly manner (which we’re all about - making it easier on the “**End User**“).

Hopefully what I’ve discussed here does make sense so far, but if not, don’t worry as we’ll be discussing these ideas further as we continue on in future articles.

Next time we’ll be adding in our two remaining columns (”Completed Early?” and “Overdue”) and introducing the concept of the “**AND**” into our “IF” which allows us to check for two conditions at the same time.

Till next time…
- Dessie

EndUserSharePoint.com: Taming the Elusive "Calculated Column" - Customizing a Task List (Part 3)

July 8, 2008

Filed Under Calculated Column, Dessie Lunsford, Tips and Tricks

View the Comments on this Article:
http://www.endusersharepoint.com/?p=458

If you haven't read through the previous two articles in this series (Part I, Part II), I would encourage you to read them first as we'll be continuing on with the use of an Excel spreadsheet developed in both of them.

Review and Next Steps

Last time ("Customizing a Task List Part II"), we created a formula to display how many days we had left to complete a task. Additionally, we created a more advanced version of that column that would show how many days left, days overdue, or completed, but in a more user-friendly manner.

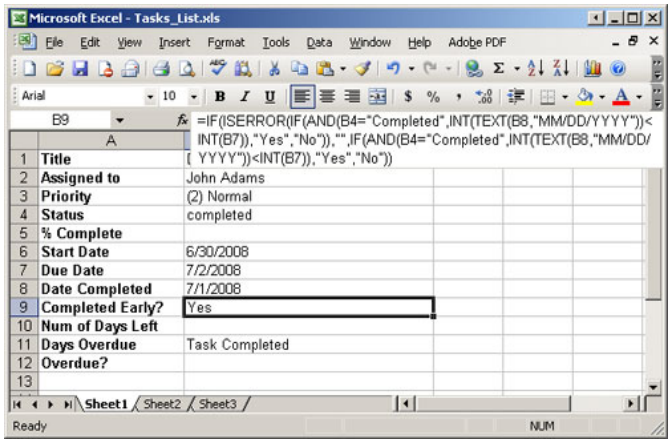
This time we'll be working with an advanced formula that uses multiple functions in a complex structure involving error checking, multiple check functions and data conversions. The field we're going to add is called "Completed Early?" and is a type of column that can be used in a custom filter to group tasks that were completed ahead of schedule (this would be an "After-the-fact" type of column that most likely would offer the most benefit to viewing tasks that have already been completed).

Working with Operators

The "**Completed Early**" column works very similar to the "Date Completed" example, but makes use of the logical operator "**AND**" that checks to see if multiple conditions exist at the same time for its comparison check. Additionally, we'll make use of multiple "IF" statements and a simple error check in performing our calculation (as well as a few conversion functions to get our data in the right format and type).

Excel Formula:

=IF(ISERROR(IF(AND(B4="Completed",INT(TEXT(B8,"MM/DD/YYYY"))<INT(B7)),"Yes","No")),"",IF(AND(B4="Completed",INT(TEXT(B8,"MM/DD/YYYY"))<INT(B7)),"Yes","No"))



Looking at this, the first thing we can see are the five distinct types of individual functions:

1. Four "INT" functions
2. Three "IF" functions
3. Two "TEXT " functions
4. Two "AND" functions
5. One "ISERROR" function

Dissecting this one will be a little bit harder than previous examples, but we can break it down into manageable chunks that we can then build back together to see how the formula as a whole, functions.

First, let's take a look at the individual pieces (the separate functions) and see what each does by itself.

The "**INT**" function is part of the "**Math and Trigonometry**" set of functions that when used, will round a number down to the nearest integer. Its only parameter is a number that is "passed" to it in between a set of parentheses.

A simple example would be the number "3.2". Using the "INT" function, we get the following:

Number = 3.2
Formula = INT(Number)
(Literal) formula = INT(3.2)
Result = 3

The "IF" function we're already familiar with since we've been using it for the last few formulas, but as a refresher, it has three parts:

The logical condition to check

Value if "True"

Value if "False"

A simple example of this would be:

IF("Sky" = "Blue", "Daytime", "Nighttime")

The logical condition to check (Is the sky blue?)

Value if "True" (Display the text "Daytime")

Value if "False" (Display the text "Nighttime")

The "TEXT" function converts a value to text and displays it in a specific numbering format we want, and it has two parts:

The value to convert

The format we want to use

An example of this would be (using a "Date"):

TEXT(Today(), "MM/DD/YYYY")
The result would be "07/02/2008".

The "AND" function is used to group together a set of "checks" that will only return a value of "TRUE" if all of the checks (arguments) are "TRUE". If one or more of the checks is "FALSE", the entire grouping is "FALSE" (this can be used by itself or in combination with the "IF" function).

This function can have as many as 30 arguments (checks) in its grouping:

- Logical check
- Logical check

- Logical check
- etc.

Examples of this would be:

AND("Sky" = "Blue", "Time" = "Daytime")
Result of above would be "TRUE" since both are true.

AND("Orange is a color", "Orange is a fruit", "Red is a color", "Red is a fruit")
Result of above would be "FALSE" since the last check is false.

IF(AND(2 + 2 = 4, "Blue is a color"), "This looks good", "Something is wrong")
The above example using the added "IF" would be broken down as:

The logical condition to check (Does 2+2=4? AND Is Blue a color?)

Value if "True" (Display the text "This looks good")

Value if "False" (Display the text "Something is wrong")

The "**ISERROR**" function is part of a collection known as the "**IS Functions**" and is a conditional-type of check that looks to see if a value is an error and returns "TRUE" or "FALSE" depending on the outcome.

Types of errors include:

- N/A
- #VALUE!
- #REF!
- #DIV/0!
- #NUM!
- #NAME?
- #NULL!

The "ISERROR" function takes in a single value and can be used in several ways to display the result.

Examples:

ISERROR(2 / 0)
The above would be “TRUE” because you can’t divide by zero, so it would be an error.

IF(ISERROR(2 + 2 = 4), “You cannot count”, “You can count”)
The above would be “FALSE” since the result of the calculation is not an error (2 + 2 does = 4), so the displayed result would be the “Result if False” of the “IF” which is “You can count”.

Using the above definitions (and examples) of the functions used in our formula, we can now begin to break down the formula and see what is occurring at each step.

Formula:
=IF(ISERROR(IF(AND(B4=”Completed”,INT(TEXT(B8,”MM/DD/YYYY”))<INT(B7)),”Yes”,”No”)),””,IF(AND(B4=”Completed”,INT(TEXT(B8,”MM/DD/YYYY”))<INT(B7)),”Yes”,”No”))

Starting from the leftmost point in the formula, we want to observe the standard “Order of Operations” that the formula will follow (hint - follow the same order you use in mathematic calculations).

Operator	Description
–	Negation (as in –1)
%	Percent
&	Concatenation (connects two strings of text)
* and /	Multiplication and division
^	Exponentiation
+ and –	Addition and subtraction
= < > <= >= <>	Comparison

To change the order (same as in algebra), we enclose calculations that need to happen first, in parentheses.

In our formula, we need to be aware of the opening and closing parentheses so we can see what has to occur in what order (which is another reason why I like to model my formulas first in Excel to take advantage of the parentheses highlighting feature).

To illustrate the flow of the formula, I’m going to list out the calculations in order to get a better feel for how it’s structured.

Try this yourself - use the "Evaluate Formula" option in Excel to follow the logic of the formula to see what occurs and in what order.

Steps (Note - *although the following example appears that we've started in the middle of the formula, we're actually starting from the far left and moving to the right until we get to an actual process which, once completed, will pass it's resulting value back to the left to the calling function - although this may sound confusing, it is accurate in how the functions operate - which is why I didn't list out the specific calling functions*):

Does B4="Completed"?
AND
Is the value of (take the date completed, convert it to text in the format of "MM/DD/YYYY" then round that down to the nearest integer)
Less than (take the value of the due date and round it down to the nearest integer)

If both of the above are "TRUE"

Display the text "Yes" (meaning, "Yes, the task was completed early")

If one or both of the above is "FALSE"

Display the text "No" (meaning, "No, the task was not completed early")

Does the above produces an error (e.g. the "Date Completed" field is blank [#VALUE! in this case])

If "TRUE" (Display nothing - this makes it so if the "Date Completed" field is blank, our calculation overrides the resulting display of the "IF" and will not display anything)

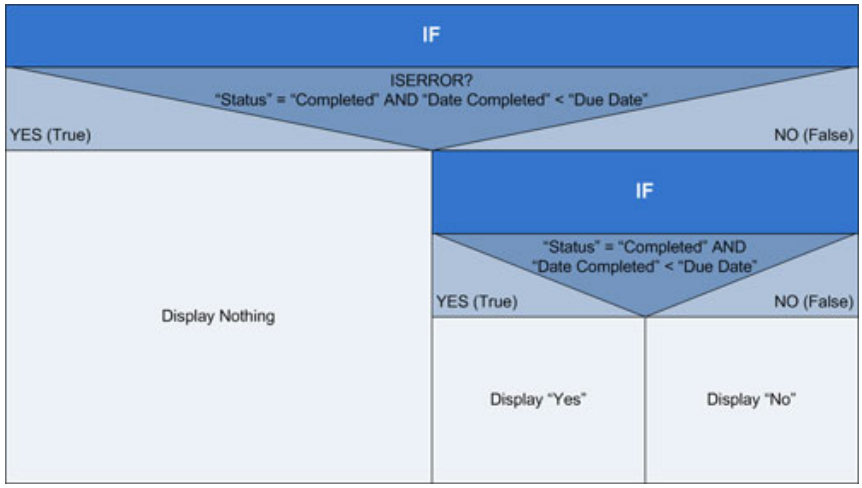
If "FALSE" (Do the calculation and display the results - this means that the "Date Completed" field is not blank, which indicates that the task is actually completed so we can go ahead and determine if it was indeed completed early and display the results which match up to that determination).

To simplify this some, think of the calculation as a speech that a writer must present to his boss before it can actually be spoke to the public. The writer passes the speech to his boss and the boss runs through the speech to make sure nothing wrong with it. If he finds any problems, he simply ignores it and throws it away (kind of a mean guy). If he doesn't find any problems, he then presents the speech to the public (all is good).

So really, when it comes down to it, all we're doing in the formula is looking to see if the "Status" field equals "Completed" and if the "Date Completed" is earlier than the "Due Date". If both of these questions are true, we display the text "Yes", and if false, we display the text "No".

Before we can do this though, we do need to make sure that no errors will occur (for the simple reason that we're relying on the "Date Completed" column which is also a calculated column, and since it is, must have a value we can work with). If we do come across an error (perhaps the "Today" calculation didn't occur for some reason - column was accidentally deleted, formula edited to produce wrong result, etc.) we don't do anything at all and display nothing. If no errors are detected, go ahead and to the calculations in the formula to give us our "Yes" or "No" answer.

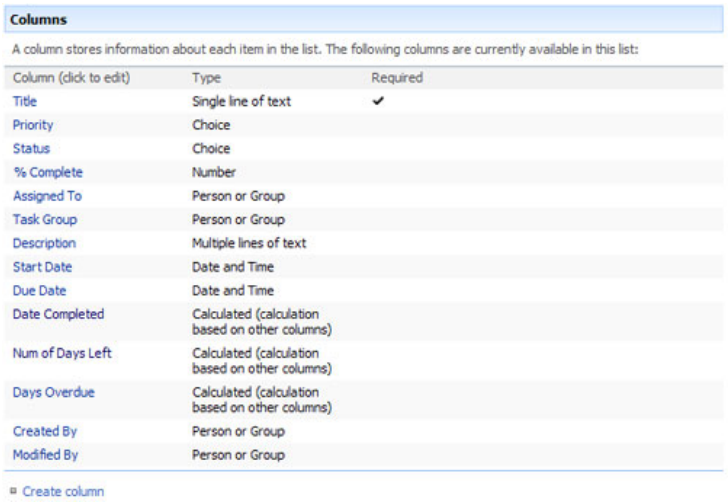
Another way to view this is through a "Nassi-Shneiderman" diagram depicting the logic process the formula follows:



(Simplified diagram depicting what actually gets processed)

So, now that we have it working in Excel, to get it into SharePoint we do the following:

In SharePoint, go into the "Tasks" list and its settings page.



Columns

A column stores information about each item in the list. The following columns are currently available in this list:

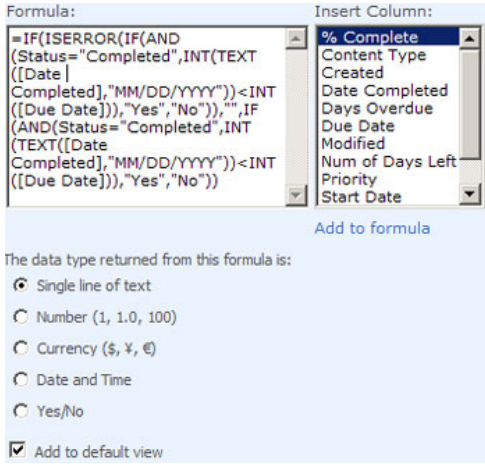
Column (click to edit)	Type	Required
Title	Single line of text	✔
Priority	Choice	
Status	Choice	
% Complete	Number	
Assigned To	Person or Group	
Task Group	Person or Group	
Description	Multiple lines of text	
Start Date	Date and Time	
Due Date	Date and Time	
Date Completed	Calculated (calculation based on other columns)	
Num of Days Left	Calculated (calculation based on other columns)	
Days Overdue	Calculated (calculation based on other columns)	
Created By	Person or Group	
Modified By	Person or Group	

Create column

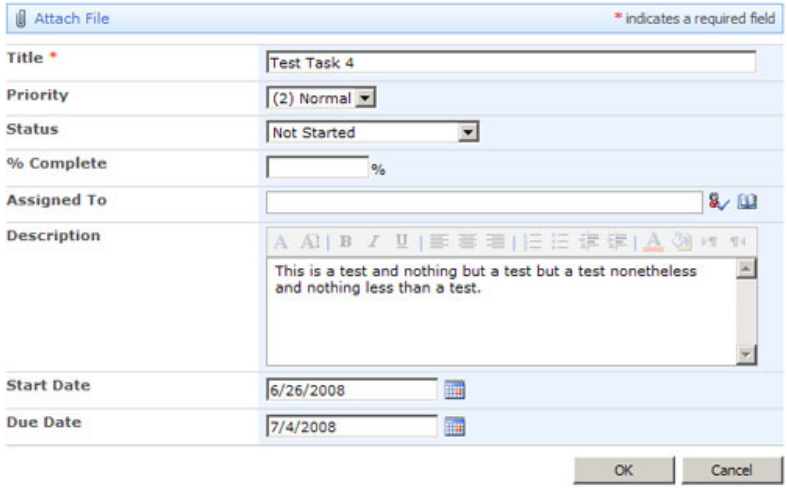
Create a new column called “Completed Early?”, make it a “Calculated” type and paste in the formula from Excel. **Before saving**, replace each occurrence of “B4” with “Status”, each occurrence of “B8” with “[Date Completed]” and each occurrence of “B7” with “[Due Date]” (making sure to include the brackets [] around the “Date Completed” and “Due Date” column titles so the system will recognize them as Columns).

Formula in Excel:
=IF(ISERROR(IF(AND(B4=”Completed”,INT(TEXT(B8,”MM/DD/YYYY”))<INT(B7)),”Yes”,”No”)),””,IF(AND(B4=”Completed”,INT(TEXT(B8,”MM/DD/YYYY”))<INT(B7)),”Yes”,”No”))

Formula in SharePoint:
=IF(ISERROR(IF(AND(Status=”Completed”,INT(TEXT([Date Completed],”MM/DD/YYYY”))<INT([Due Date])),”Yes”,”No”)),””,IF(AND(Status=”Completed”,INT(TEXT([Date Completed],”MM/DD/YYYY”))<INT([Due Date])),”Yes”,”No”))

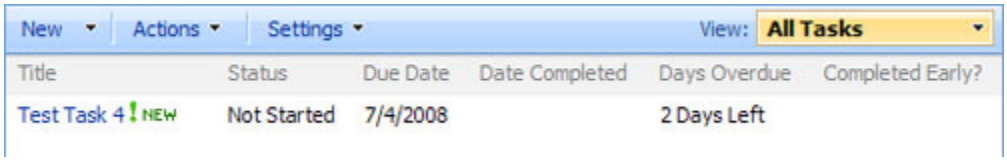


To test our new column, create a new item in the list with a "Start Date" in the past and a "Due Date" several days in the future.



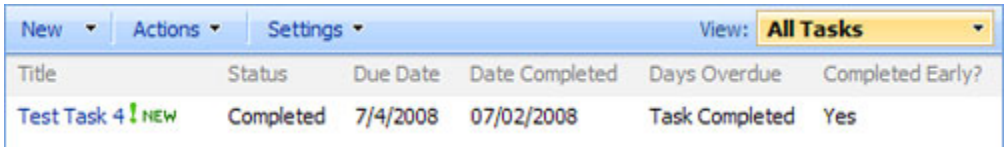
Once saved we can see that (based on our previously created columns) we do have time left to complete the task and that it is currently not completed.

Note - I've removed a few of the columns from the view to make it easier to see in the following screenshots.

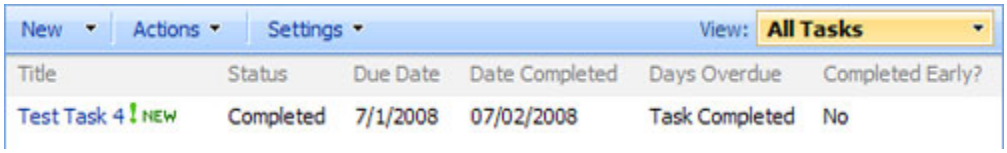


To test the functionality of the column, we'll edit the task to see what results will be displayed.

Set the task as "Completed" (note the "Completed Early?" column displays a "Yes" indicating that the task was completed before the "Due Date"):



Set the task "Due Date" to yesterday (note the "Completed Early?" column displays a "No" indicating that the task was completed after the "Due Date"):



SUMMARY

We now have is way to view which tasks have been completed ahead of schedule and which ones were completed late. Depending on your environment, you could use this in a custom view with filters and grouping to organize tasks based on upcoming due dates, completed early and completed late (which may assist in organizing day-to-day tasks for employees).

Next time we'll be finishing off this series on the "Tasks" list by adding our final column (had to cut it from this one since this post is already quite lengthy) and looking back at each of the columns we've created to see how they will all fit together as a more "Fully-Functional" list.

Till next time…
- Dessie

ENDUSERSHAREPOINT.COM: TAMING THE ELUSIVE "CALCULATED COLUMN" - CUSTOMIZING A TASK LIST (PART IV)

JULY 24, 2008

Filed Under Calculated Column, Dessie Lunsford, Libraries and Lists, Tips and Tricks

Follow Comments on this Article:
http://www.endusersharepoint.com/?p=591

If you haven't read through the previous three articles in this series (Part I, Part II, Part III), I would encourage you to read them first as we'll be continuing on with the use of an Excel spreadsheet developed in each of them.

REVIEW AND NEXT STEPS

In the **last article**, we created a calculated column which gave us the ability to see if a task had been completed early. This type of information could be beneficial to managers performing an audit of a particular users (or groups) performance in completing items in a timely manner.

This time, we'll be completing our customization of the "Tasks" list by adding in our final column called "Overdue?".

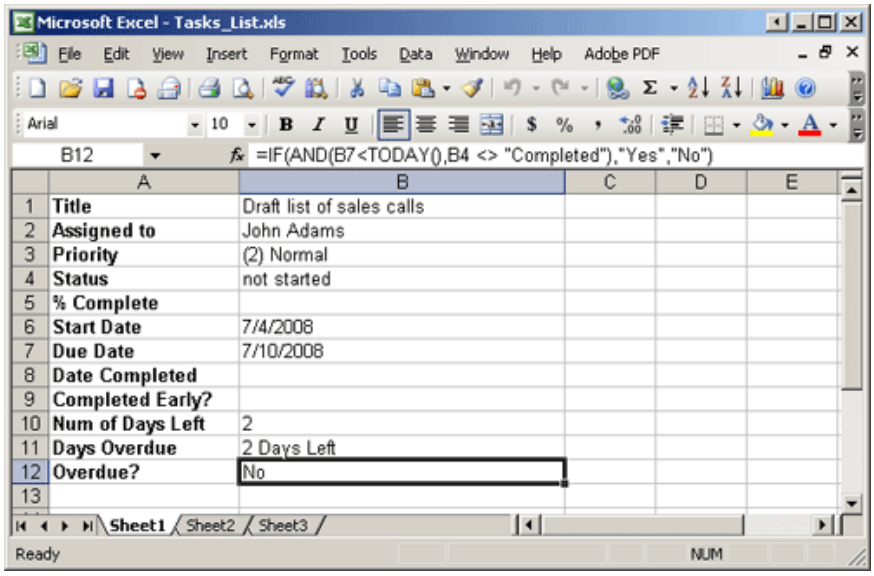
THE FORMULA

The "**Overdue?**" column is used to see "At-a-glance" which tasks are currently overdue and display a simple "Yes" or "No" depending on its status (which could assist in the day-to-day management and processing of Tasks).

Compared to previous examples, this column is rather straight-forward in its formula and what it's looking for, but at the same time it can be somewhat confusing unless you understand the structure and logic of the formula and how it works to produce a specific result.

(See the previous articles in this series to create and setup your Excel spreadsheet.)

Formula in Excel:
=IF(AND(B7<TODAY(),B4 <> "Completed"),"Yes","No")



Once again we see our old friend, the "IF" statement, as well as the "AND" function we discussed last time.

Dissecting this formula, we begin with the "IF" and its three parts:

1. The logical condition to check (is the "Due Date" less than today's date? **AND** is the value of the "Status" column not equal to "Completed"?)

2. Value if "True"(display the text "Yes")

3. Value if "False" (display the text "No")

In the formula's conditions, our first check is looking to see if the "Due Date" is less than "Today's" date. If it is, the result of the first check is "True". This is "technically" ok by itself, but it doesn't deal with tasks that haven't been completed yet, so we need to combine this with a second check to see what the "Status" is of the task.

For the second check, instead of looking to see if a specific value exists, we're looking to see if it doesn't. In this case, if the "Status" field **does not** equal "Completed" then the result is "True" (anything else will yield a result of "False"). Again, by itself this doesn't really help us because not every task that hasn't been completed is overdue (it wont be overdue until we have moved past the actual due date).

Combining multiple checks together (up to 30 of them) is a form of what's known as "**Boolean Logic**". To accomplish our end result of either "True" or "False" for the

combination of these checks, we're going to use an "**AND**" statement to merge our two checks into one statement that must be "True" (as a whole) in order to produce a result of "Yes" being displayed (note - the "AND" by itself will only produce a "TRUE / FALSE" result, so by adding it to an "IF", we can then control the displayed output of "TRUE" being displayed as "Yes" and "FALSE" being displayed as "No").

If we use another type of "**Logic Diagram**" tool called a "**Truth Table**", we can check each combination of values to see just what results are possible.

Input1 (1st check)	**Input2 (2nd check)**	**Result of each**	**End Result**
Due Date is less than today	Status not equal to "Completed"	**TT**	TRUE
Due Date is not less than today	Status not equal to "Completed"	**FT**	FALSE
Due Date is less than today	Status is equal to "Completed"	**TF**	FALSE
Due Date is not less than today	Status is equal to "Completed"	**FF**	FALSE

*The "*Truth Table*" although depicted rather simply here, can be used in conjunction with advanced calculations as well. In future posts, I'll dive deeper into the use of this tool to aid in checking your formula logic.*

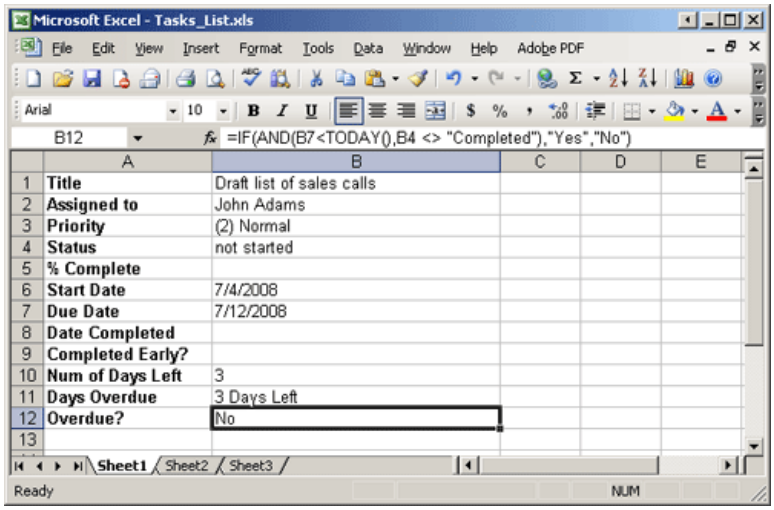
In our formula (and following the examples in the Truth Table):

1. If the "Due Date" is less than "Today" (meaning we're currently past the "Due Date") and the "Status" field does not equal "Completed", the entire comparison is "True" which results in a "Yes" being displayed (**True: True**).
2. If the "Due Date" is not less than "Today" (meaning we still have time left) and the "Status" field does not equal "Completed", the entire comparison is "False" which results in a "No" being displayed (**False: True**).
3. If the "Due Date" is less than "Today" (again, meaning we're currently past the "Due Date") and the "Status" field is equal to "Completed", the entire comparison is "False" which results in a "No" being displayed (**True: False**).
4. If the "Due Date" is not less than "Today" (again, meaning we still have time left) and the "Status" field is equal to "Completed", the entire comparison is "False" which results in a "No" being displayed (**False: False**).

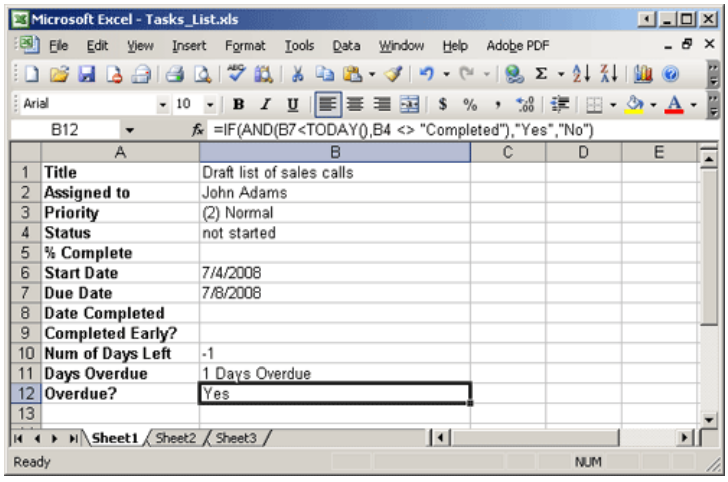
In the above four possibilities, the only result that will display a “Yes” is if we’re past the “Due Date” and the “Status” is anything but “Completed (each of the other three will either mean the task has been completed or we still have time left).

Testing this in Excel is as follows.

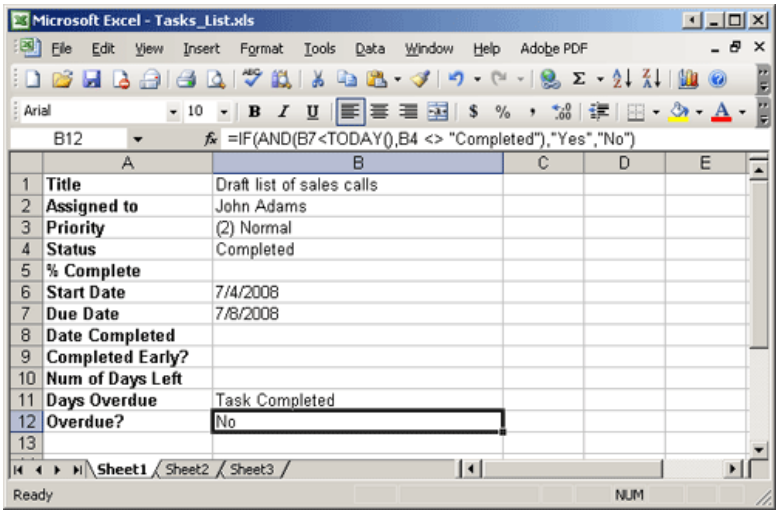
Enter a start date in the past and a due date in the future (notice the “Overdue?” column displays “No” indicating the task is currently not overdue):



Change the due date to be yesterday (notice the “Overdue?” column displays “Yes” indicating that we’re currently past the due date and that the task is overdue):



Modify the task again and set the status as “Completed” (notice the “Overdue?” column now displays “No” since the task is completed).



One extra thing to point out in the logic of this column is that it really only applies to tasks that are open. Once a task is completed we’re not concerned with it being overdue anymore, so after its completion it will always display a “No” (W*e’re using the “Completed Early?” column to inform us of tasks that were completed ahead of schedule, so adding in logic for that information on this column would be redundant*).

To get this into SharePoint, we follow the same approach as before:

In SharePoint, go into the “Tasks” list and its settings Page.

Columns

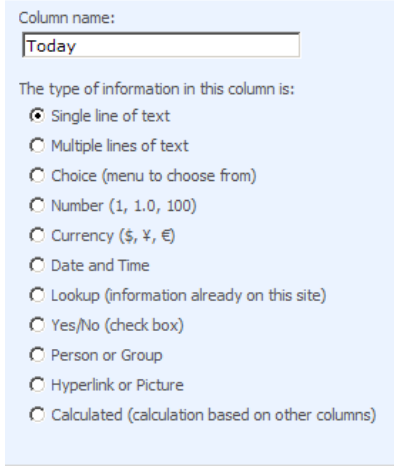
A column stores information about each item in the list. The following columns are currently available in this list:

Column (click to edit)	Type	Required
Title	Single line of text	✔
Priority	Choice	
Status	Choice	
% Complete	Number	
Assigned To	Person or Group	
Task Group	Person or Group	
Description	Multiple lines of text	
Start Date	Date and Time	
Due Date	Date and Time	
Date Completed	Calculated (calculation based on other columns)	
Num of Days Left	Calculated (calculation based on other columns)	
Days Overdue	Calculated (calculation based on other columns)	
Completed Early?	Calculated (calculation based on other columns)	
Created By	Person or Group	
Modified By	Person or Group	

Create column

Since we're once again using a "Today" reference in our formula, we'll need to create our temporary "Today" column that will allow us to use the reference without displaying any errors.

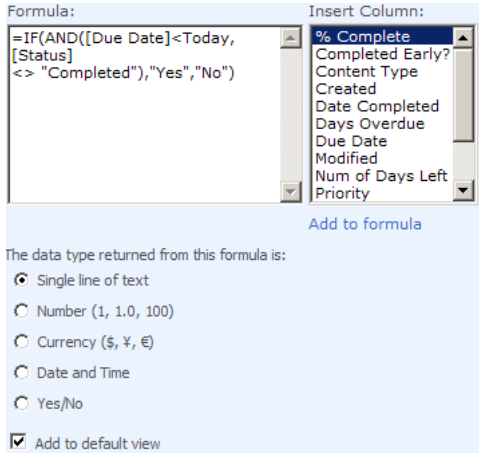
Create a new column called "Today" leaving the rest of its settings as default.



Next, create another new column called "Overdue?", make it a "Calculated" type and paste in the formula from Excel. **Before saving**, replace "B7" with "[Due Date]" and "B4" with "Status" (making sure to include the brackets [] around "Due Date" so the system will recognize it as a column).

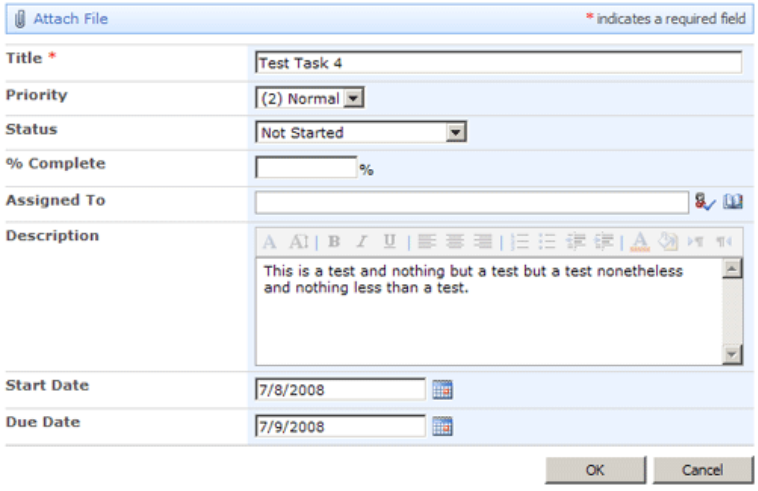
Formula in Excel:
=IF(AND(B7<TODAY(),B4 <> "Completed"),"Yes","No")

Formula in SharePoint:
=IF(AND([Due Date]<Today,[Status] <> "Completed"),"Yes","No")

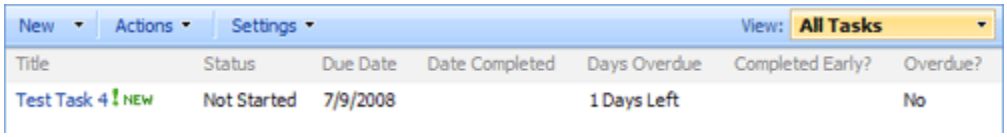


After you have the new calculated column, we need to go back and delete our temporary "Today" column (as discussed previously, this column was simply a placeholder that fools the system into allowing us to use the name for our formula, so deleting it will now match it up with the real "Today" data in the system, but without throwing an error).

To test our new column, create a new item on the list with a "Start Date" of the (default) current date and a "Due Date" of tomorrow.



Once saved, we can see that our new column displays a value of "No" indicating that the task is not completed and that we still have time to complete it before it comes due.

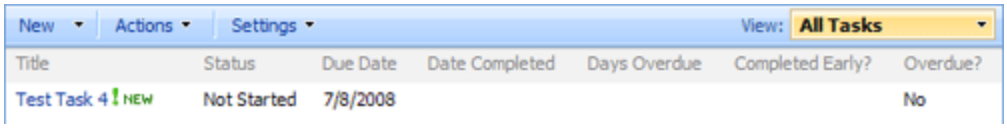


Our first check is to see if the "Due Date" is less than the current date. In this case, since the "Due Date" is tomorrow (greater than today, not less than), the result is "False". The second check is looking to see if the value of the "Status" column is "less than or greater than" (equivalent to the statement "Not equal to") the text "Completed", which is "True" since the task is in fact not completed. Since both checks are not "True" (first is "False", second is "True"), the entire check is "False" which results in a display of the text "No".

*Looking back at our "*Truth Table*", we see that any combination of the two checks that results in anything other than the two of them being "True" will result in the entire check being "False", which in this case, is our final result and displays the text "No".*

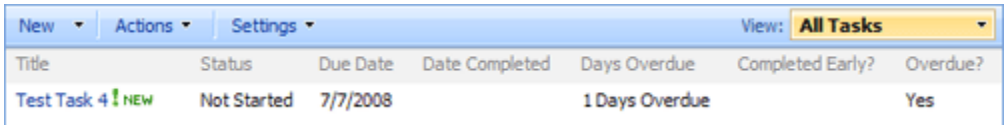
To test what other values could be displayed, we simply modify the details of the task to see what the results are.

Set the "Start Date" back a couple of days and the "Due Date" to today's date (notice how the value remains "No"?).



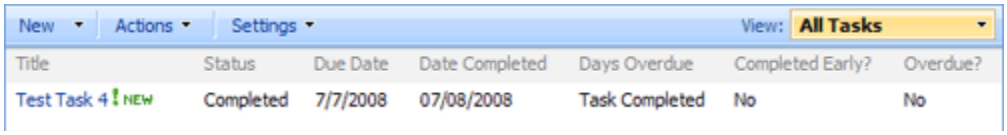
Since the "Due Date" is not less than today's date (it's equal), the first check is "False". The second check however, is "True" since the value of the "Status" field does not equal "Completed", so since both checks do not result in "True", the entire check is "False" resulting in "No" being displayed.

Modify the item again, but this time set the "Due Date" to yesterday (notice how the value now display's a "Yes"?).



In this example, the first check is “True” since the “Due Date” is less than the current (”Today”) date. The second check is also “True” since the value of the “Status” field does not equal “Completed”. As a result of both checks being “True”, the entire condition is “True” which gives us a displayed result of “Yes” indicating that the task has not been completed and is currently overdue.

Modify the item once again and mark the task as completed (notice how the value now display’s “No”?).



For this one, without even looking at the first check we know immediately that this will display a value of “No”. Why? Because in the second check we’re looking at the value of the “Status” field, and testing to see if it “Does Not” equal the text “Completed”. Literally, if it equals anything but the text “Completed” it will be “True”. Since we’ve set the task as completed, it will now automatically become “False”. Because of this, no matter what possible value could be in the first check, the entire (combined) check will always be “False” for the simple reason that the second check is “False”. So, once the task is completed it is no longer overdue (logically, this should make sense).

SUMMARY

Putting all of this together, let’s take a quick look at what we’ve accomplished throughout this series and how to make it an effective solution for making the “Tasks” list more useful.

Our five calculated columns we’ve created are:

1. “**Date Completed**” - Uses the “Today” reference to mark when a task was completed.
2. “**How many days left to complete?**” - Visual indicator of time left on a task.
3. “**Days Overdue**” - Advanced version of the “How many days left to complete?” column with better information.
4. “**Completed Early?**” - Visually indication of tasks completed ahead of schedule.
5. “**Overdue?**” - At-a-glance view of tasks that are overdue.

Through each of these articles, we've worked with multiple functions (some separate, some in concert with each other):

- **IF** - Conditional statement containing three parts (logical condition to check, value if true, and value if false) that allows us to display a particular result based on a check of a given condition.
- **AND** - Boolean operator that allows us to check multiple conditions at the same time where each individual check must equal true in order for the whole (combined) condition to result in true.
- **TEXT** - Conversion function that takes a value and converts it to text in a specified "number" format.
- **INT** - Math function that simply rounds a given number down to it's nearest integer.
- **ISERROR** - "Error trapping" function that gives us the ability to decide what will be displayed if an error is discovered somewhere in our calculation.

And, we've also worked quite a bit with the "**Today**" reference and how to use it in a Calculated Column (despite Microsoft's own documentation stating that it can't be done) by using a temporary column to bypass the infamous SharePoint error messages.

In the "Out of the box" implementation of the Tasks list, the functionality is limited, as it only gives us information as to when the task is due, a percentage of its current completion and status.

By adding in our custom columns, we can now see if a task is overdue, how many days it's overdue or have left, the date it was completed, and whether or not it was completed ahead of schedule. With these new columns, the possibility for custom views begin to surface allowing us more advanced ways of filtering tasks for the users they are assigned to, those that review the tasks, and those that are responsible for reporting and auditing task history after task completion.

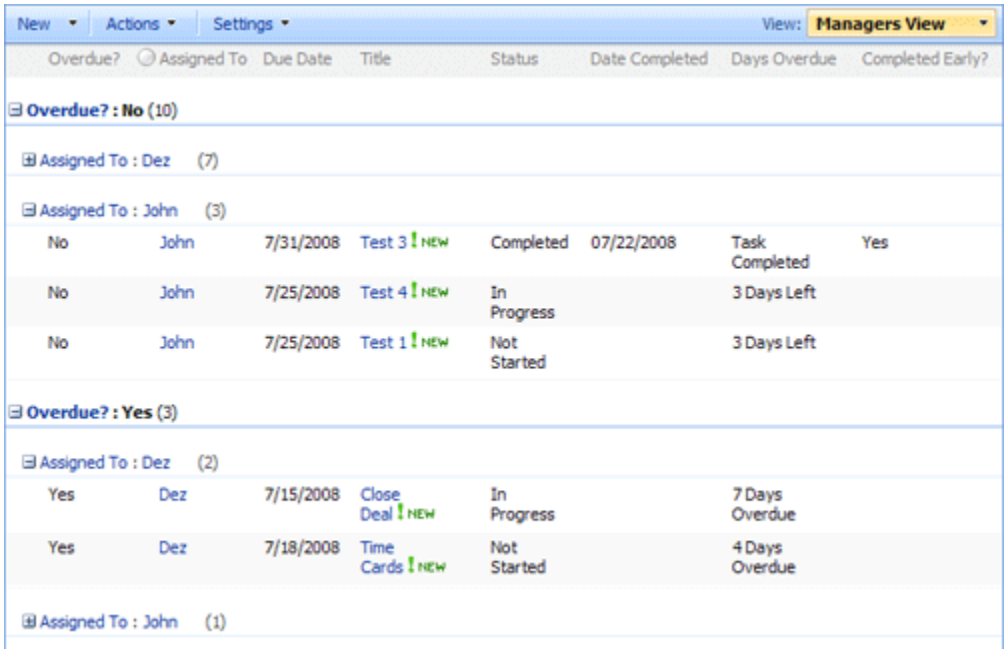
Simple Custom Views

Example of custom view for **assignees**:

New | Actions | Settings | View: Tasks Assigned to Me

Overdue?	Due Date	Title	Status	Date Completed	Days Overdue	Completed Early?
Overdue? : Yes (2)						
Status : In Progress (1)						
Yes	7/15/2008	Close Deal NEW	In Progress		7 Days Overdue	
Status : Not Started (1)						
Yes	7/18/2008	Time Cards NEW	Not Started		4 Days Overdue	
Overdue? : No (3)						
Status : Not Started (3)						
No	7/22/2008	Presentation NEW	Not Started			
No	7/25/2008	Meet with Investors NEW	Not Started		3 Days Left	
No	7/31/2008	Write Report NEW	Not Started		9 Days Left	

Example of custom view for **mangers**:



New | Actions | Settings | View: Managers View

Overdue?	Assigned To	Due Date	Title	Status	Date Completed	Days Overdue	Completed Early?
Overdue? : No (10)							
Assigned To : Dez (7)							
Assigned To : John (3)							
No	John	7/31/2008	Test 3 NEW	Completed	07/22/2008	Task Completed	Yes
No	John	7/25/2008	Test 4 NEW	In Progress		3 Days Left	
No	John	7/25/2008	Test 1 NEW	Not Started		3 Days Left	
Overdue? : Yes (3)							
Assigned To : Dez (2)							
Yes	Dez	7/15/2008	Close Deal NEW	In Progress		7 Days Overdue	
Yes	Dez	7/18/2008	Time Cards NEW	Not Started		4 Days Overdue	
Assigned To : John (1)							

Example of custom view for **auditors**:

In future articles, we'll be covering more functions and more advanced formulas that will give us even more power in how we view data (perhaps even a few more additions to the "Tasks" list).

Till next time…
- Dessie