Applied TAM / Microsoft® Word - IF, ASK, Math, SET, Bookmarks, INCLUDETEXT

SESSION HANDOUT

APPLIED NET
CLIENT NETWORK CONFERENCE
www.appliednet.com

Prepared for Applied Systems Client Network and Applied Systems

Applied Systems Client Network
2340 South River Road
Des Plaines, IL 60018
Phone: 800-605-1045
Fax: 224.220.1443

Applied Systems, Inc.
200 Applied Parkway
University Park, IL 60484
Phone: 708-534-5575
Fax: 708-534-8016

Original Author:

Graham Blundell, CIC, CPCU, GB Consulting Services, Middletown, NJ

Updated By:

Marcia Priest, CIC AB SOLUTIONS, INC AGENCY OPERATIONS CONSULTING 8/12/15

Target Audience:

	Accountant/Bookkeeper	X	Producer
X	CSR	X	Trainer
	Carrier		Vendor
X	IT Manager/Systems Coordinator		ALL
X	Operations	X	Other: Document Setup Specialist
X	Principal/Owner		

Table of Contents

Seminar Type: Microsoft Products

Seminar Level: Advanced: An advanced-level course facilitates the development of thoughts (ideas, theories, procedures) to the most advanced stage possible. For functional courses, the objectives taught at this level challenge learners to analyze and make business decisions on how to utilize the agency management system or software in their agency, or apply a high level of technical aptitude to understand in-depth features and functionality.

Class Description: Using the previously installed set of documents, Microsoft ® Word skills for customizing the output of the templates will be reviewed. The goal is to generate a client-friendly proposal while improving the efficiency of the agency's internal process to create them.

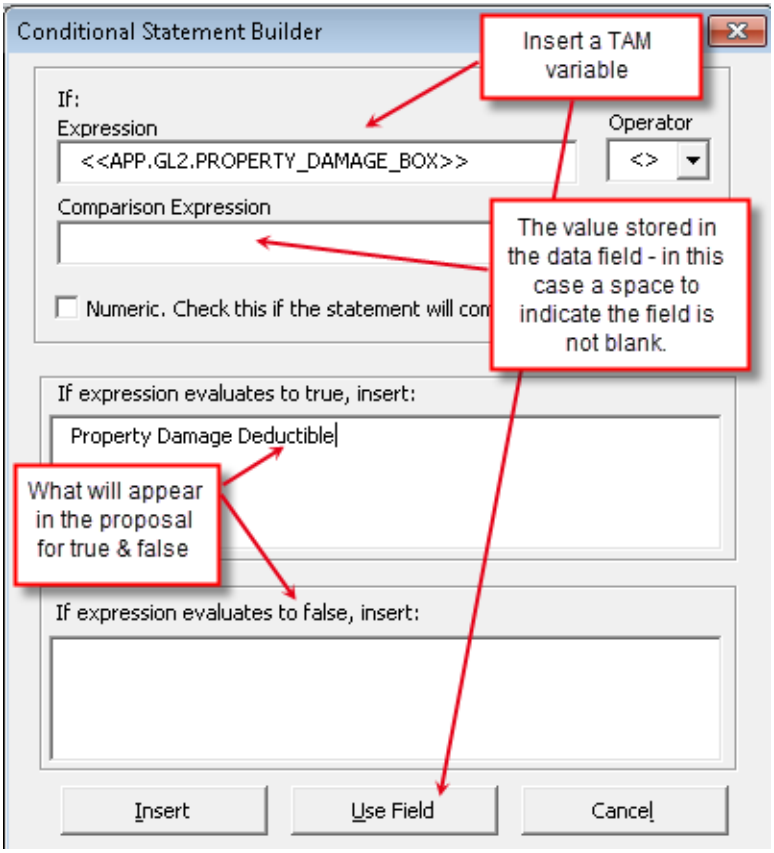
Learning Outcomes:

- Implement "No Coverage" wording if a specific vehicle does not have physical damage deductibles entered.
- Include or exclude coverage sections depending upon the information in the application.
- Include the specifics on the Service Team page for the servicing personnel assigned to the account – contact info, bios, photos.

Assumptions: This seminar is based on the following
Applied TAM Version 2015
Microsoft ® Word Version 2010/2013

Basic IF Statement Review

Using the Applied/Word interface, we can build an IF statement for the General Liability proposal page to test for the type of deductibles on the policy.



In this case, because we are checking to see if the Property Damage Deductible checkbox is not blank, the true would be the wording “Property Damage Deductible” and the false is blank. The IF statement looks like this when inserted into the template:

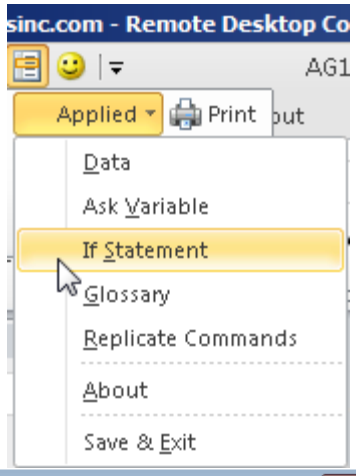
{ IF "{ DOCVARIABLE APP.GL2.PROPERTY_DAMAGE_BOX }" <> " " "Property Damage Deductible" "" }

When using the Applied menu to insert an IF statement into your document, you are limited in what you can insert in the true and false sections of the statement. However, you can edit the statement in your document after it has been created and there are virtually no limits on what you can put in each section.

Basic IF Statement Creation

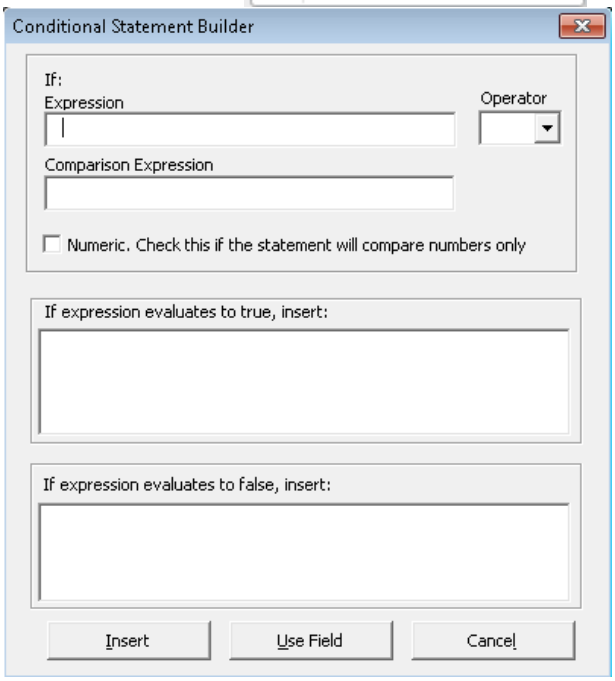
To create an IF statement:

- Confirm your paragraph markers are on (click the ¶ icon on your Home ribbon)
- Reveal Code – Hold down the ALT key and then click the F9 key on your keyboard
- Place cursor where you want the IF statement result to appear in the merged document.
- Click Applied Menu, IF Statement

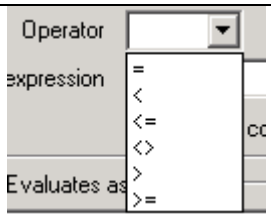


Conditional Statement Builder Box opens:

- Expression field is generally based on an Applied TAM Data Variable but can also be based on the results of an ASK variable.
- To Add a Data Variable, with cursor in the Expression field, click on Use Field, or you may use a “placeholder” and place the data field later.
- Click on each applicable folder to navigate to the field you want to use in the IF Statement.
- Click OK.



- The next field to complete is the Operator Field and the available values from the drop-down menu are:

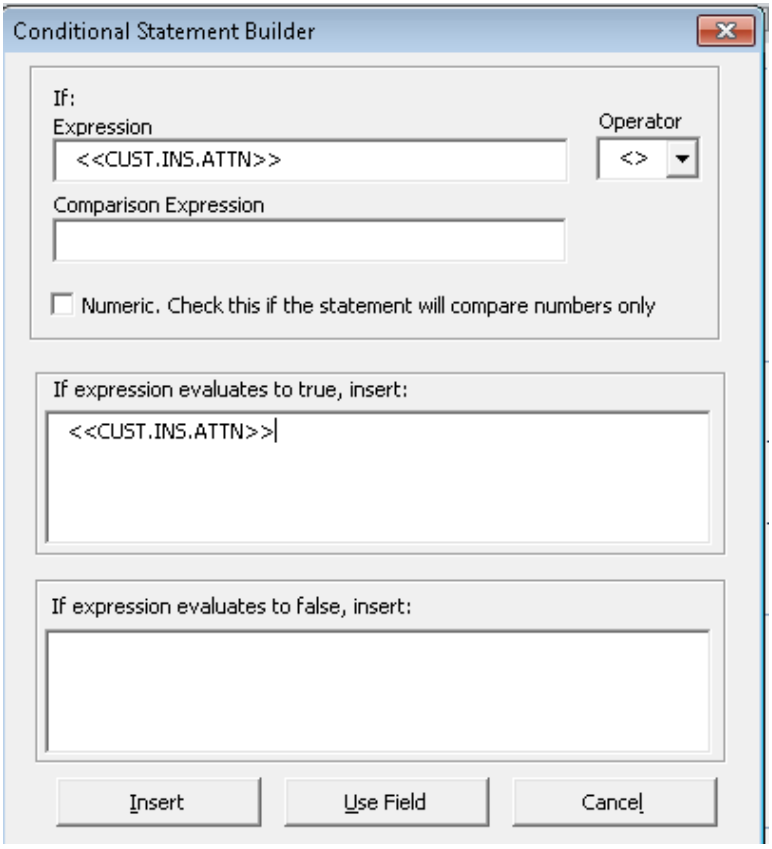


=	Looks for an exact match in comparison expression
<	Less than the value in comparison expression
<=	Less than or equal to comparison expression
< >	Not equal to comparison expression
>	Greater than comparison expression
>=	Greater than or equal to comparison expression

Select the appropriate value for the type of IF statement being built.

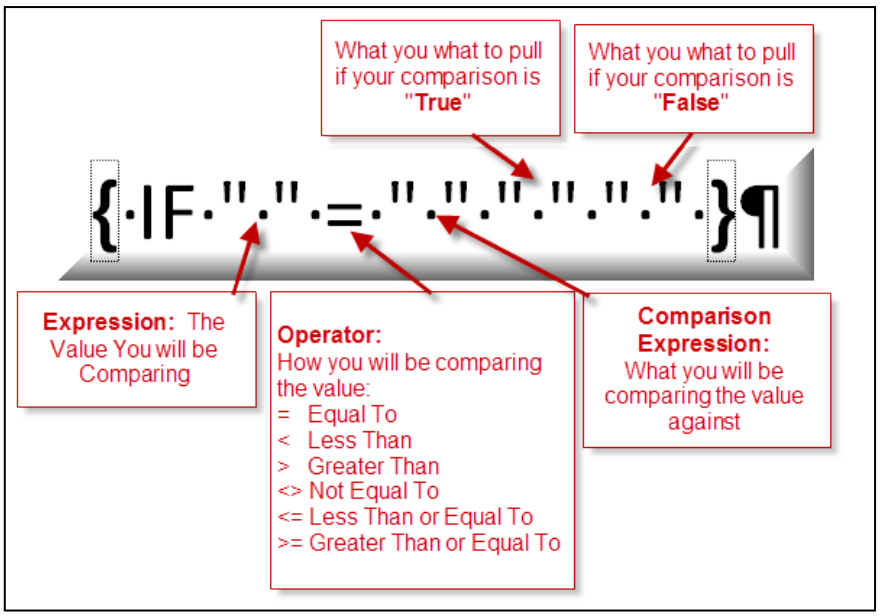
- The Comparison Expression field should be completed with the value you are comparing the value in the Expression field with.
- The TRUE field should contain what you want to pull into your merged document if the IF statement has a true result.
- The FALSE field should contain what you want to pull into your merged document if the IF statement has a false result (which could be nothing).
- Click Insert.

An example of a completed IF statement checking to see if the Attention field is blank:



Let's break down what each piece of this IF statement means:

You can use place markers in any of these fields if you want to Insert the IF statement and then complete it manually.



Notes about True and False portions:

- You do not have to have ANY value in these fields if you don't want anything inserted into the document (example, if there is information in the second address line of the customer's address you want whatever is in that field to pull into the document PLUS put a line return before it to separate it from whatever the first address line value is, but if the field is empty, you don't want anything to pull or any line returns entered to separate. Just leave the False portion blank. The IF statement will look like this:

  { IF "{ DOCVARIABLE CUST.INS.ATTN }" <> " " "¶
  "{ DOCVARIABLE CUST.INS.ATTN }"" "" }

- You can format the text within the True and False portions of the IF statement any way you like. It can be bolded, italicized, bulleted, etc.
- Scanned images can be inserted into the True and False portions (an example would be, if Pr1 is equal to JEC, enter this picture or signature graphic, etc.).

To create an IF statement in your document, and then modify it beyond what is possible using the Applied interface box:

- Have your cursor placed in the document where the results of the IF statement will reside.
- Click Applied, IF Statement.
- Complete the first three items (Expression, Operator, Comparison Expression) with the information you want in them.
- In "true" and "false" sections, insert placeholders for now (you could use "aaa" and "bbb", or even "true" and "false").
- Click on Insert to put the statement in your document.

Now, after pressing Alt-F9, if necessary, to reveal your field codes, you can edit the expression, replacing the placeholders with whatever you want. There is no limit on the amount of text you can place in there, including paragraph marks, page breaks, tabs, formatting, even pictures.

Important: When editing an IF statement, make sure you keep the basic structure – especially the double quotes – in place.

Nested IF Statements

Nesting is putting IF statements inside other IF statements. You would want to use this feature if you have more than two values you are comparing.

For instance:

In your Business Auto app, some of your vehicles makes may have been entered (perhaps via download) in all uppercase (TOYOTA instead of Toyota). You want the proposal to have it in upper/lower case – but you don't want GMC, BMW or VW in upper/lower case. So we have to do multiple tests on the data.

{ IF "{ DOCVARIABLE APP.BA2.MAKE }" = "GMC" "GMC" "{ IF "{ DOCVARIABLE APP.BA2.MAKE }" = "BMW" "BMW" "{ IF "{ DOCVARIABLE APP.BA2.MAKE }" = "VW" "VW" "{ QUOTE { DOCVARIABLE APP.BA2.MAKE } * caps }" }" }" }

Let's analyze that statement:

First, we test the make to see if it is GMC. If it is, the true section kicks in, GMC is entered in the document, and the IF statement is finished. If it isn't GMC, the false section is a test for BMW. Again, if true it concludes. Otherwise it tests for VW. If it isn't VW, the false section of that IF statement instructs Word to insert the contents of the field in upper/lower case.

To create the above nested IF statement:

1. Create the first IF statement, testing for GMC. Put GMC in the "true" section and leave the "false" section blank.
2. Click Insert to close the IF statement dialog box and put the basic IF statement in the document.
3. If necessary press Alt+F9 to reveal the entire coding of the IF statement.
4. Place your cursor in the false section of the IF statement (between the two double quote marks).
5. Click Applied, IF Statement.
6. Create the second "nested" IF statement, testing for BMW.
7. Repeat steps 2-6, putting your cursor in the false section of the BMW IF statement before creating the IF statement that tests for VW.
8. Put your cursor in the false section of the VW IF statement. This time you are not going to insert an IF statement, you are inserting the datafield name, and then applying the formatting switch to it to force it into upper/lower case.

Once you are comfortable with editing IF statements in the document, you can take the next step and insert them directly into the document instead of using the Applied/Word interface dialog box.

- At the place in the document where you want the IF statement, press Ctrl-F9. This inserts an empty Word field, which looks like this: 
  They look like open and closing curly brackets but you cannot use those keyboard characters to enter them – you must use Ctrl-F9.
- You can now type, directly from your keyboard, the contents of the Word field inside those brackets. Use the Applied, Data dialog box to enter fields (or copy and paste them from elsewhere in your document if you've already got them somewhere).
- Since this is an empty Word field, it hasn't yet been defined as an IF statement. So the first thing you'll type is the word "IF": { IF }.
- It is not a bad idea to set up the basic layout of the IF statement before you input all of the details. In a complicated statement, this can help you keep track of what's going on.
  { IF "expression" = "test" "true" "false" }
  Hints:
  - Be sure you have formatting marks () turned on.
  - Use spaces between the various sections of the IF statements. These have no effect on the operation of the IF statement but help to make it clear where you're at in the statement.

You can nest several to many IF statements inside of each other. There is no maximum limit of IF statements that can be nested in Applied TAM.

Copying and Pasting Between Documents

In Applied TAM, you have complete freedom to copy fields, including datafields, within the same template and the fields will not “break”. But you cannot copy from one document to another. You can certainly copy a complicated IF statement from one document to another, but all of the field codes (including Replicate commands) will need to be replaced through the Applied Data menu.

You can, however, copy a document “outside” the Applied TAM / Word interface that includes field codes and those codes would not break. This is very similar to the process used to install the free set of proposal templates. For details, please refer to the “Copying documents so that field codes don’t break” section of the Appendix in this handout.

Putting a Picture in an IF Statement

By “picture” we are referring to any JPG, TIFF, or other file format that Word recognizes and can import. A picture can be:

- An agency logo
- A scanned signature
- An employee photo for the Service Team page
- A piece of clip art appropriate to the line of business

Insert a picture into an IF statement by putting the cursor in the correct place in the IF statement – in either the "true" section or the "false" section – before inserting the picture. For instance, perhaps there are two agency logos and you want to use logo #1 for Agency #1 and logo #2 for all others.

- Create an IF statement that tests if the Agency datafield is equal to 1
- Between the double quotes of the “true” section, insert logo#1
- Between the double quotes of the “false” section, insert logo#2.

Resulting IF statement: { IF "{ DOCVARIABLE AGCY.INS.REC }" = "1" "Agency1 logo goes here" "This logo for all other agencies" }

Select and paste each picture in turn and format it to be the size and location you want. In the master document, the pictures won't seem to be in the IF statement – they will appear on the page based on the formatting you apply, and may in fact overlap each other, but when you run the proposal, they will show up correctly.

The above example uses a simple IF statement – “If this is true insert this picture, otherwise insert this other picture”. We are not limited to an either/or decision; just as we did with the vehicle make, we can used a nested IF statement to pick among

numerous agency logos, or, alternately, we could use several IF statements “end to end,” each with a true but no false result.

Inserting a Large Block of Text in an IF Statement

Now that we have the concept of editing an IF statement directly, and copying and pasting into the “true” and “false” sections, we can apply that technique to all kinds of scenarios.

- Some applications cover more than one type of coverage or exposure
  - Accounts Receivable/Valuable Papers
  - Garage and Dealers covers Service Risks; Auto Dealers; Garagekeepers Legal Liability
- The Equipment Floater includes both Unscheduled and Scheduled Equipment screens. Many insureds only have one or the other of these exposures. If there is no list of scheduled equipment, it would be nice to suppress the table that replicates the equipment.
- We may have the same application used for more than one coverage type. The most common example is Wind coverage in Florida – it is often written on a separate policy from the rest of the property, but we still link the policy type to the Property app.

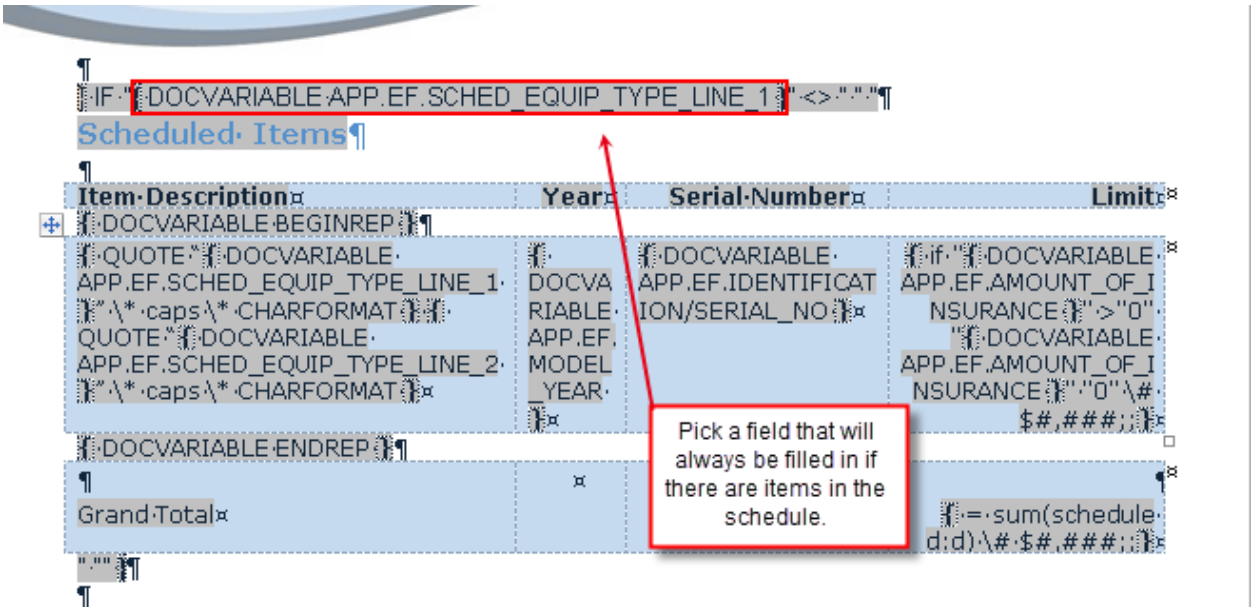
The least complicated way to build these types of tests is to create all of the text, tables, replicate commands, etc. that you want before you put them in the IF statement. It is much easier to copy and paste them into the IF statement later than to have the added complication of them already being in an IF statement while you’re setting the IF statement up.

Let’s go through an example, using the Scheduled Equipment section of the Contractors Equipment Floater. First, set up the entire section of the proposal document that you want when there is scheduled equipment on the account.

Test the document on an account that has an equipment schedule and modify it until you are satisfied with it. Now decide which field on an equipment schedule will always be completed if there is any equipment scheduled on the policy. In this case, let’s assume there will always be an item number entered. So that is the field we will use in the IF statement we’ll build for the test.

Above the beginning of the heading, insert a simple IF statement that tests for the item number field being non-blank. Start by leaving both the “true” and “false” sections of the IF statement empty.

Now select and cut the entire content you previously created for the schedule, from the heading until after the ENDREPLICATE, and paste it into the “true” section of the IF statement.



Notice that we have not only the table replicating the schedule but also the section heading - **Scheduled Items**. Sometimes the IF statement will pick up the formatting from the heading, but that is OK if that happens.

Now test the document on accounts that have a schedule and accounts that don’t have a schedule. In the first case, you should get the schedule you had previously; in the second case, you’ll get a blank document, because the IF statement evaluates to “false” and we have nothing entered there.

Note that the datafield we are testing on,
{ DOCVARIABLE APP.EF.SCHED_EQUIP_TYPE_LINE_1 } is a schedule item, but we have it outside the replicate commands in the test section of the IF statement. This means it will only read it one time, for the first item on the schedule. This what we want; if this was inside replicate commands, we would get as many schedules as there are items in the schedule!

You can follow the principles described above to make all kinds of choices in a document based on what is in a particular app. For instance, on the Garage app, you could test whether the risk is a dealer or a nondealer; put the dealer coverage descriptions in the “true” section and the nondealer coverage descriptions in the “false” section of the IF statement.

Advanced Use of ASK Variables

The simple ASK statement solicits information from the user and prints the answer in the document. However, we can do several things to make these more useful.

- Use the default answer option to guide the user as to the format of their response. For example:
  - Prompt: What is the effective date of the endorsement? Enter in the format shown below.
  - Default answer: January 1, 2014
- Use datafields in the prompt – and in the default answer – to save the user keystrokes. Use Insert Field in the ASK setup dialog box to put the datafield into the prompt and/or the default answer. For example:
  - Prompt: Is { DOCVARIABLE C_I.INS.NAME } the carrier you are proposing? Click Ok for yes, or type the carrier name exactly as you want it to appear in the proposal.
  - Default answer: { DOCVARIABLE C_I.INS.NAME }
- Use the answer to the question, not as printed output in the document, but as input for an IF statement. Details in the next section.

ASK/IF Combination

In order to use a combination of the ASK variable and IF statement, create the ASK variable first. For example, in some cases you need a deposit in order to bind coverage. So we ASK the question whether a deposit is needed, then we create an IF statement that "reads" the answer to the question and prints different text in the letter depending on the answer.

To create the ASK/IF combination:

- Create the ASK statement. It will look like this: { ASK deposit "Will a deposit be needed to bind coverage? Answer 1 for Yes or 2 for No" }{ REF deposit }
- Place the cursor where you want the results of the IF statement to appear in the template.
- Open the Conditional Statement builder box.
- Put a placeholder in the “Expression” field in the IF statement.
- Choose “=” as the Operator
- Put a 1 in the “Comparison Expression” field in the IF statement.
- Complete the true and false sections with the wording desired for deposit needed or not needed.
- Click Insert.
- Select and cut the { REF deposit } that was created immediately after the { ASK …} field.
- Paste it into the “Expression” portion of the IF statement.

The IF statement would look like this:
{ IF "{REF deposit}" = "1" "In order to bind coverage, we will need a deposit in the amount shown on the premium summary page." "No payment is needed to bind coverage; you will be billed later." }

The actual answer to the question asked is never printed in the document; it is used to decide what other text is inserted in the document.

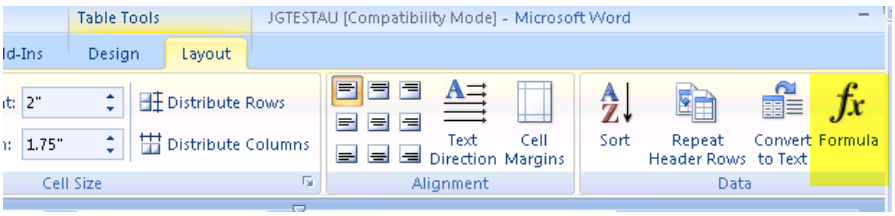
Math Functions in Tables

Although not as powerful or as varied as Excel, there are some basic math functions available within tables in Word.

Tables have cell references just like Excel EXCEPT you do not see the Column and Row labels. You have to count them out for yourself. Here is an example of how Word "labels" each cell:

A1	B1	C1	D1
A2	B2	C2	D2
A3	B3	C3	D3

Place your cursor in the cell where you want the formula results to be displayed. The table tabs on the ribbon will activate. Click on the Layout tab, then click on the formula button.



If there are numbers displayed in the cells above or to the left of the selected cell, an automated formula (=Sum(Above)) or (=Sum(Left)) will be shown in the formula field.

If neither of these is the desired formula, delete the automated formula, leaving just the equal (=) sign in the brackets. Or of course you can enter the field manually as described earlier, with Ctrl-F9 and typing inside the brackets. Then enter the formula you need based on the available functions. Examples of the available functions are shown on the next page.

Add – "+"

Add a number to a cell	=(A1)+3
Add two adjacent cells	=SUM (A1:A2) or =(A1+A2)
Add two non-adjacent cells	=(A1+A3) or =(A1)+(A3)
Add a range of cells	=SUM(B1:B3)
Add an entire column	=SUM(D:D)

Subtract – "-"

Subtract a number from a cell	=(A1)-1
Subtract two cells	=(A3-B2)

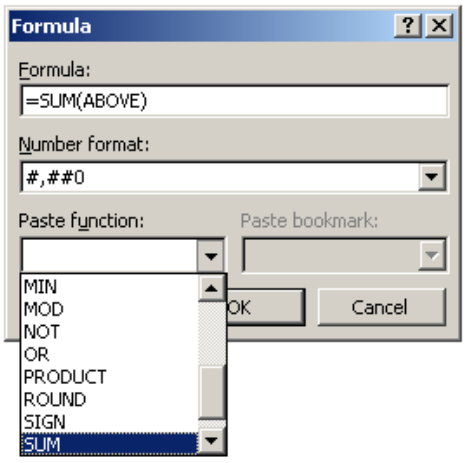
Multiply – "*"

Multiply a cell by a number	=(A1)*3
Multiply two adjacent cells	=PRODUCT(A1:A2)
Multiply two non-adjacent cells	=(B1*B3) or =PRODUCT(B1)*(B3)

Divide – "/"

Divide a cell by a number	=(C2)/3
Divide two cells	=(C2)/(A1)

If you are using the Formula button under the Layout tab, select the function from the Paste Function box. For instance, to add numbers, choose SUM from the Paste Function drop-down box. Or you can manually type these functions into the formula field.



To reference cells in formulas, use a comma to separate references to individual cells (i.e., (=SUM(A1,B3) and a colon to separate the first and last cells in a designated range (i.e., (=SUM(A1:D1). You can reference an entire row or column in a calculation in the following ways:

- Use a range that includes only the letter or number that represents it — for example, 1:1 to reference the first row in the table. This designation allows the calculation to automatically include all the cells in the row if you decide to add other cells later.

- Use a range that includes the specific cells — for example, a1:a3 to reference a column with three rows. This designation allows the calculation to include only those particular cells. If you add other cells later and you want the calculation to include them, you need to edit the calculation.

You can format the results of your formula by selecting the format from the available options in the Number format box. For example, to display the numbers as a decimal percentage, click 0.00%.

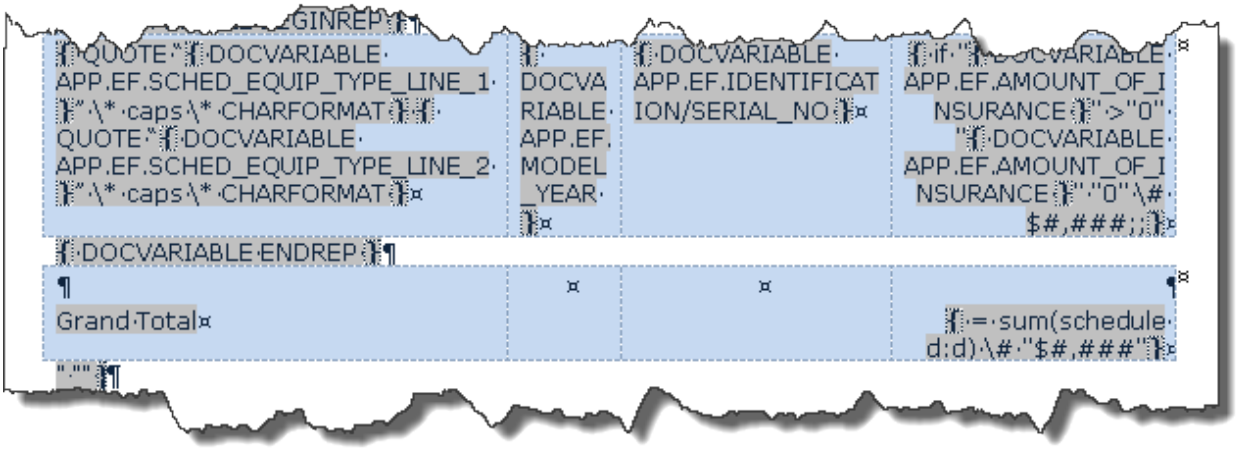
Notes:

- Word inserts the result of the calculation as a field in the cell you selected and does not automatically update the formula results if there are changes to the referenced cells. You can update the calculation by clicking once on the field (or selecting the entire table) and then pressing F9. Also, once you build the proposal at the client level, all the formulas are replaced by the actual results; if you change the numbers in the cells, the total will need to be updated manually.
- If you see codes between braces instead of the actual sum (for example, { =SUM(LEFT) } not the number result you wanted), press SHIFT+F9 or ALT+F9 to switch to the number result.

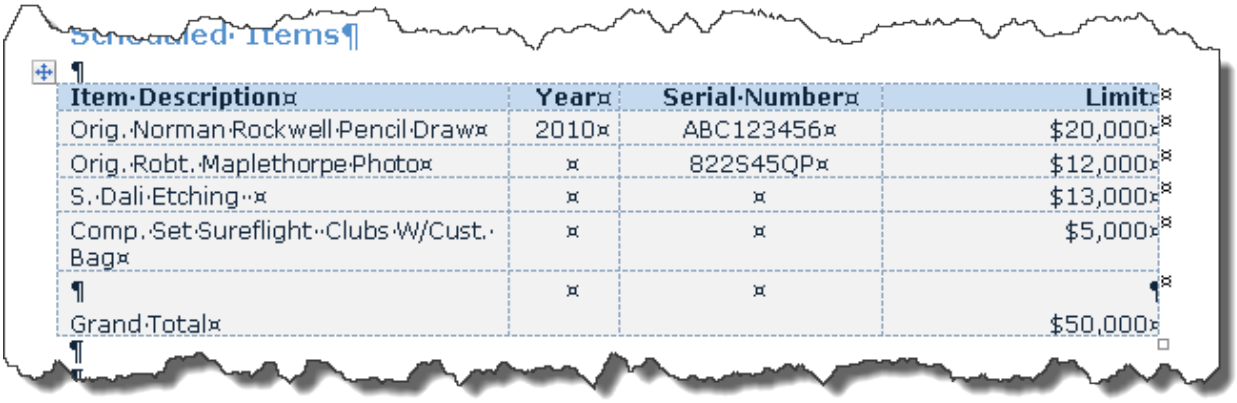
Math Sample

Here is a sample of the coding for creating a total of the values in an equipment schedule.
The total is outside the replicate commands. You only want one total!

The formula uses the SUM function referencing the fifth column (column e) in the table. By using this method instead of SUM(ABOVE), the total will be correct even if there are blank limit entries. SUM(ABOVE) stops calculating if it encounters a blank cell.
Apply the same formatting to the total formula as you apply to the individual items.



And here's the result on the client's proposal:



Scheduled Items

Item Description	Year	Serial Number	Limit
Orig. Norman Rockwell Pencil Draw	2010	ABC123456	$20,000
Orig. Robt. Maplethorpe Photo		822S45QP	$12,000
S. Dali Etching			$13,000
Comp. Set Sureflight Clubs W/Cust. Bag			$5,000
Grand Total			$50,000

SET Command and Bookmarks

A bookmark identifies a location or a selection of text that you name and identify for future reference. It is also the name Word uses for the variable in a SET field, and that is the primary way we use bookmarks in proposals. The SET field is a way to assign a value to a variable and then use that variable throughout the document.

The format of the SET field is as follows:

{ SET bookmark "text" }

Another field you need to know about is the REF field, which you've already seen if you use ASK statements – it prints (or uses in an IF statement) the value of a bookmark.

Here are some examples of ways to use the SET field in our documents:

- You have an ASK field that prompts for the answer "Yes" or "No". You will then use that answer in an IF statement to insert appropriate text in the document. The user might type "Yes", "yes", "Y" or "y" in answer to the question, which would require up to four different IF statements to catch. But a SET field can eliminate the upper/lower-case issue, and a wild card in the IF statement can take care of the Y or Yes possibilities.
    - Let's assume the ASK field generates a variable "depositrequired"
    - {SET depositrequired "{QUOTE depositrequired * upper}" }
      This forces the answer into upper case.
    - { IF "{ REF depositrequired }" = "Y*" "Yes answer text" "No answer text" }
      This just looks at the first character of the answer.

- IF you want to use a nonschedule field within a schedule – for instance in the vehicle schedule – you want to show the liability limit next to each vehicle. You cannot put nonschedule fields inside Replicate commands; you will get an error message. So what you do is use the SET field, before the Replicate, to assign the limit to a bookmark, then use the bookmark inside the Replicate. Here's a simplified example:
    - This goes before the Replicate starts:

      { SET autolimit "{ DOCVARIABLE APP.BA2.LIABILITY_CSL_BOX }" }

    - Inside the Replicate, put the limit next to each vehicle in the schedule:

      { DOCVARIABLE APP.BA2.Year } { DOCVARIABLE APP.BA2.MAKE } {REF autolimit}

    - If you need number formatting, you can apply it either in the SET command or in the REF command, as you prefer.

- You are going to use a reference file to create a master list of employee signature information so that you only have to update employee information in one place. The SET field transfers the employee code to the reference file.

INCLUDETEXT

The INCLUDETEXT function is used to pull signature information into formletters from a single reference file. In Applied TAM, we cannot get anything other than the Producer, CSR or Operator name, and if we want to put the other information in the letters, we have traditionally had to create IF statements like this:

{ IF “<CSRCODE>” = “ABC” “Email@company.com” “” }, repeated for every CSR.

And that had to go in every letter. When you hired a new employee, you had to edit every one of your letters to add the IF statement for that employee.

We now have a way to store all this “extra” information in a single reference file that you call from each of your letters. When you get a new employee all you have to do is update the reference file and you are done.

To set it up, we do have to edit every one of our letters so they will call the reference file and pass the employee code to it. This is a one-time deal – you don’t have to edit this again no matter how many employee changes you have in the future.

For Applied TAM, we use the SET field to assign the Producer code, (or CSR or Operator code, depending what you are looking to pull in) to a bookmark name. This is what that looks like:

{ SET PRODCODE <AP.INS.REC> } or
{ SET CSRCODE <CUST.INS.CSRCODE> }
{ SET OPIDCODE <AGCY.SYSTEMENV.OPID> }

Why do we need to do this? Because we are going to go out and read a second document (see the next item), and we cannot have Applied TAM datafields in this second document – they will not get converted by the Applied/Word macro. So we need to create a Word bookmark (PRODCODE, CSRCODE or OPIDCODE), which will be read in the second document.

The second thing we need to put in every letter is the { INCLUDETEXT *"filename and path"* } field, which tells Word to include the contents of the file specified in the field. This field needs to be at the place in the letter where you want the information to appear. Some important things to note about the *filename and path*:

- The entire entry must be enclosed in double quotes " ".
- There must be two \ (backslashes) between the items in the path, not single backslashes, as we are used to. So if our reference file is named ZXTNSNS.RTF and is on the L: drive, in the \DOC\CFORMLET folder, the field reads:
  { INCLUDETEXT **"**L:****DOC****CFORMLET****ZXTNSNS.RTF**"** }
- Remember, the quotes and double backslashes are critical to this process.

The path can be to any file location accessible to your workstations, and can even be a URL rather than a drive and folder reference (it must be where the actual file resides). In this example, these files were created through Document Setup in Applied TAM, so they are in the CFORMLET folder, but it is not necessary for them to be created there. If you are pointing to a URL rather than a drive and folder, the address starts with four backslashes, and all other backslashes are doubled – like this: { INCLUDETEXT \\\\servername\\etc }.

The reference file contains a string of IF statements that read the bookmark name and the IF statements put the appropriate information, based on the value of the bookmark, into the letter. There is nothing unusual about these IF statements except that they read the value of the bookmark (for instance CSRCODE) to do the comparison.

So in the CSR reference file, the IF statement begins:

{ IF { REF CSRCODE } = "GB" "GB info goes here" "" }

{ REF *bookmark* } is another Word field that reads the value of the bookmark.

Be sure to not have any extraneous text or characters (including line returns) in this reference file. *Everything* in that file will be dumped into the letter (as long as it passes the test in the IF statement).

So your reference file will look like this:

{ IF { REF CSRCODE } = "AB" "AB info goes here" "" }{ IF { REF CSRCODE } = "CD" "CD info goes here" "" }{ IF { REF CSRCODE } = "EF" "EF info goes here" "" }{ IF { REF CSRCODE } = "GH" "GH info goes here" "" }{ IF { REF CSRCODE } = "IJ" "IJ info goes here" "" }{ IF { REF CSRCODE } = "KL" "KL info goes here" "" }

Make sure there are NO line returns between the IF statements. There is a single line return at the very end of the file that will get pulled into the template, so you do need to take account of that in your individual templates.

Note that the actual info you're pulling in can be multiple lines long – like this:

{ IF {REF CSRCODE } = "GB" "Senior Account Manager
GBlundell@GBConsulting.biz
908-489-2119" "" }

It is line returns ***outside*** the IF statements that you must avoid.

You can set up another file, if desired, that works the same way, but also pulls in the actual signature image. Because we want the Producer name below the signature but above the rest of the information, incorporate it into the reference file, using the { SET PRDNAME <AP.INS.NAME> } field in the letter and the { REF PRDNAME } field in the reference file. The IF statement in the reference file will look like this:

{ IF {REF CSRCODE } = *"*GB*" "sigimagefilehere*
{ REF PRDNAME }
Senior Account Manager
GBlundell@GBConsulting.biz
908-489-2119*" ""* }

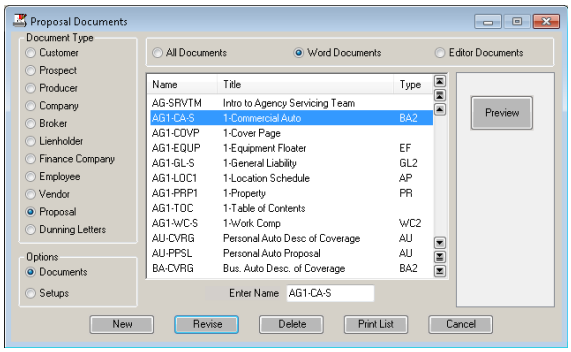
This is an excellent way to have the one reference file to edit to update staff changes.

Appendices:

Appendix A: Copying documents so that field codes do not break

Creating a new document and copying and pasting the data from the old document, or choosing Insert File, will "break" many of the field codes used. An alternative is to copy and paste the file to where Applied TAM Document Setup expects to find it, and then Add the new document in the Document Setup Utility.

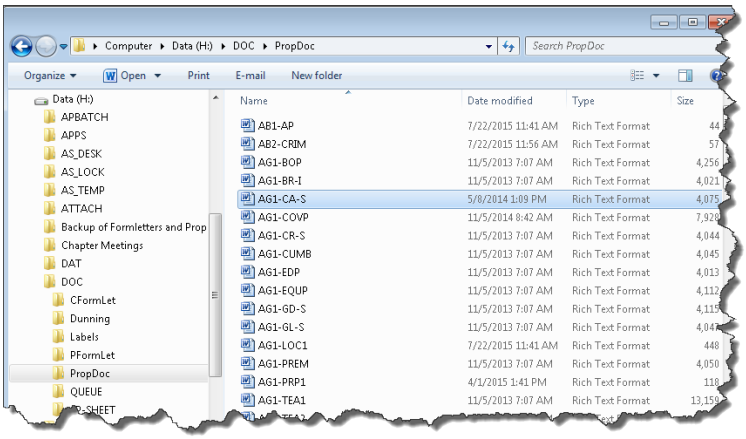
As an example, if you want to make a copy of your updated Proposal template to make a similar Summary document:



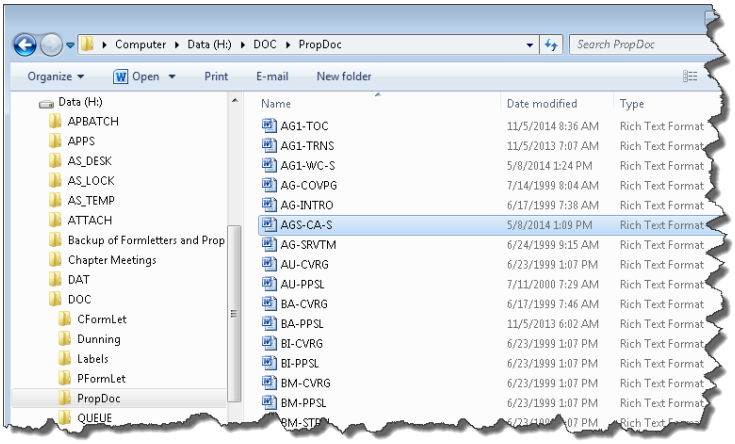
1. Using Windows Explorer or My Computer, navigate to the directory where the Applied TAM Template is stored. In this case, that would be H:\DOC\PROPDOC

2. Find the AG1-CA-S template on the list, select, right-click and Copy, then right-click in the same directory and Paste.

   For Applied TAM Online, Edit Copy and Edit Paste will also work, or CTRL-C and CTRL-V.



3. Rename the copied file to reflect the new *Name* desired, In this case AGS-CA-S. Use care not to change the document extension as it needs to remain .RTF.



4. Access Document Setup in Applied TAM and select ***New***. Add the newly created *Name* and *Description* in the appropriate fields, and the old template should default in the Word window when complete. All field codes will still function properly when using this method.

5. Revise the template as needed for your Summary template. Remember also to put it into your Proposal Setup for your Summaries.

Appendix B: Other Classes and Handouts from the Proposal Track Series

- Proposals and the Power of Graphic Design – This class also includes information on a Style Guide you may find useful for your design specifications.
- Microsoft® Word – Introduction to the Applied TAM Interface – Install Free Set – Includes instructions for downloading and installing templates.
- Intermediate Microsoft® Word Formatting – Applied TAM – includes Tables, Borders & Shading, Section Breaks, headers & footers, etc.
- Leveraging Microsoft® Word Styles - Applied TAM
- Applied TAM/Microsoft® Word - IF, ASK, Math, SET, Bookmarks, INCLUDETEXT
- Applied TAM/Microsoft® Word – Formatting Numbers, Text & Dates
- Applied TAM Proposal Workshop / Advanced Clinic – This session has no handouts and will be based upon questions and issues brought to the group by the attendees.