SEMINAR HANDOUT

Applied Epic/Microsoft[®] Word[®] – IF, ASK, Math, SET, Bookmarks, INCLUDETEXT

Applied Epic/Microsoft[®] Word[®] – IF, ASK, Math, SET, Bookmarks, INCLUDETEXT

SESSION HANDOUT



APPLIED NET CLIENT NETWORK CONFERENCE www.appliednet.com

SESSION HANDOUT

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



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

Copyright © 2015 by Applied Systems Client Network, Inc., 2340 S. River Road #311, Des Plaines, IL 60018. Protection claimed in all forms and matters of copyrightable material and information now allowed by law or hereafter granted including both electronic and conventional distribution of herein products. Reproduction or transmission unless authorized by Applied Client Network in writing. All rights reserved. Specific product information regarding Applied TAM®, Applied Vision®, Applied Epic® and related products and services, including any related manuals, documentation, and/or materials prepared by Applied Systems for purposes of Applied Net 2015 or otherwise are the exclusive property of Applied Systems, Inc. Applied Systems retains all right, title, and interest therein, including copyright interests and other intellectual property rights. Information relating to products and services owned or licensed by third parties (ex: Microsoft, Excel, etc.) and all interests therein are the property of the respective owners, and no endorsement or ownership of third party products or services should be implied by their mention and use. All workflows are suggested and common workflows. Users of this material agree that neither Applied Client Network nor Applied Systems can be held liable for any omissions or errors within the guide.

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



Original Author

Graham Blundell, CIC, CPCU, GB Consulting Services, Middletown, NJ GBlundell@GBConsulting.biz 908-489-2119

Target Audience

	1
	1
Х	(
	(

Х

Accountant/Bookkeeper

CSR

Carrier X IT Mana

IT Manager/Systems Coordinator

Operations

Principal/Owner

Х	Producer
Х	Trainer
	Vendor
	ALL
	Other: (describe)



Table of Contents

Basic IF Statement Review	6
Nested IF Statements	7
Copying and pasting between documents	9
Putting a Picture in an IF Statement	9
Inserting a Large Block of Text in an IF Statement	10
Advanced Use of ASK Variables	13
ASK/IF Combination	13
Math Functions in Tables	14
Math Sample	17
SET Command and Bookmarks	18
Servicing Roles	19
Pulling signature information into templates from a single reference file using INCLUDETEX	(T.21
Appendix	24
Other Classes and Handouts from the Proposal Track Series	24

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



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

You will review how we customize the data in order to generate a client-friendly proposal.

Learning Outcomes

- 1. Have your agency's proposal say "No Coverage" if a specific vehicle does not have physical damage deductibles entered
- 2. Include or exclude coverage sections depending on the information in the application
- 3. Include the specifics about the servicing personnel assigned to the account contact info, bios, photos

Assumptions

This seminar is based on the following Applied Epic Version 2014/2015 Microsoft ® Word Version 2010/2013



Basic IF Statement Review

Using the Applied/Word interface we build an IF statement such as this, which reads the value (Y or N) entered in the Auto app for Hired Auto Liability, and inserts more readable Included or Excluded in the proposal.

If Statement Builder		Insert an Epic
Expression YESNO1		variable
Operator =	The value	
Comparison expression Y	datafield	
Numeric comparisation	arison	
If Expression Evaluates as True	/	
What you wa	unt to	
proposa	i /	
If Expression Evaluates as False	/	
Exccluded		
	/	
Add Ask Variable Add Epic Variabl	e Insert C	ancel

The IF statement looks like this when inserted into the template:

{ IF "{ DOCVARIABLE

MasterAccountMerge.Policy.Line.LOB.BusinessAuto.Coverages.HiredBorrowedLiability. HRDBD.YesNo1*0*110 }" = "Y" "Included" "Excluded" }

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's been created and there are virtually no limits on what you can put in each section.

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 3 items (Expression, Operator, Comparison Expression) with the info 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

MasterAccountMerge.Policy.Line.LOB.BusinessAuto.RiskMerge.Vehicles.Vehicle.VehM ake*0*103 }" = "GMC" "GMC" "{ IF "{ DOCVARIABLE

MasterAccountMerge.Policy.Line.LOB.BusinessAuto.RiskMerge.Vehicles.Vehicle.VehM ake*0*103 }" = "BMW" "BMW" "{ IF "{ DOCVARIABLE

MasterAccountMerge.Policy.Line.LOB.BusinessAuto.RiskMerge.Vehicles.Vehicle.VehM ake*0*103 }" = "VW" "VW" "{ QUOTE { DOCVARIABLE

MasterAccountMerge.Policy.Line.LOB.BusinessAuto.RiskMerge.Vehicles.Vehicle.VehM ake*0*103 } * 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 filed, 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's 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 IF statements inside of each other although there is a limitation in Applied Epic that will cause an error if there are more than 18 IF statements.

Copying and pasting between documents

In Applied Epic, you have complete freedom to copy fields, including datafields, from one document to another. So you can save yourself some time – here's the process:

- Open the document that has the code you want to copy (you can open it in Word Viewer rather than Word; that way you can keep it open while you're working on the new document.)
- Select and copy the entire nested IF statement.
- In the new document, paste the copy in the relevant spot. (You may have to toggle field codes – Alt-F9 – in order to see the IF statement you just pasted in.)

This technique works with all Word fields no matter how complicated they are.

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



You 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 you insert the picture. For instance, perhaps you have 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.

{ IF "{ DOCVARIABLE MasterAccountMerge.Account.Client.ServiceBilling.Agcy.Agency .Code*0*109 }" = "1" "Agency1 logo goes here" "This logo for all other agencies" }

Select 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'll appear on the page based on the formatting you apply, and may in fact overlap each other, but when you run the proposal they'll 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.

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 it in the IF statement. It is much easier to copy and paste it into the IF statement later than to have the added complication of it already being in an IF statement while you're setting it 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 till after the ENDREPLICATE, and paste it into the "true" section of the IF statement.

IF." DOCVARIABLE. MasterAccountMerge.Policy.Line.L cheduled.SchdItemNum*0*13 ". Item.#¤ Description¤	.OB.Equipme <>·'''·''Sche ID/Serial·#¤	entFloater duled-Equ Owned¤	Scheduled.S ipment¶ Amountof-Insi
COCVARIABLE BEGINREP Control of the schedule sc	DOCVARIABLE MasterAccount Merge.Policy.Li ne.LOB.Equip mentFloater.S cheduled.Sche duled.SchdSer Num*0*15	DOCVARIA BLE MasterAcco untMerge.P olicy.Line.L OB.Equipm entFloater. Scheduled. Scheduled. Scheduled. SchdOwnd* 0*16	QUOTE DOCVARIABLE MasterAccountM, erge.Policy.Line, LOB.EquipmentF loater.Scheduled .Scheduled.Sche AmtIns*4*17 \#•\$#,##x



Notice that we have not only the table replicating the schedule but also the section heading - **Scheduled Equipment**, and that the IF statement picked up the formatting from the heading – that's OK.

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 MasterAccountMerge.Policy.Line.LOB.EquipmentFloater.Scheduled. Scheduled.SchdltemNum*0*13 } 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 non-dealer; put the dealer coverage descriptions in the "true" section and the non-dealer 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.
 - o 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 MasterAccountMerge.Policy.Line.Line.IssuingCompany.Common.Co mmon.AcctName*0*8 } 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 MasterAccountMerge.Policy.Line.Line.IssuingCompany.Common.Co mmon.AcctName*0*8 }
- 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 IF Statement builder box.
- Put a placeholder in the "Expression" field in the IF statement.
- 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's 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.

	Table Tools		JGTESTA	AU [Compatibi	lity Mode]	- Microsof	t Word			-
ld-Ins	Design	Layout								
it: 2" 1: 1.75"	: #	Distribute I Distribute	Rows Columns		Text Direction	Cell Margins	A Z↓ Sort	Repeat Header Rows	Convert to Text	f x Formula
Cell	Size		G.	A	lignment			Data		

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. Here are some examples of the available functions:

Add - "+"

Add a number to a cell Add two adjacent cells Add two non-adjacent cells Add a range of cells Add an entire column	=(A1)+3 =SUM (A1:A2) or =(A1+A2) =(A1+A3) or =(A1)+(A3) =SUM(B1:B3) =SUM(D:D)
Subtract – "-" Subtract a number from a cell Subtract two cells	=(A1)-1 =(A3-B2)
Multiply – "*" Multiply a cell by a number Multiply two adjacent cells Multiply two non-adjacent cells	=(A1)*3 =PRODUCT(A1:A2) =(B1*B3) or =PRODUCT(B1)*(B3)
Divide – "/" Divide a cell by a number Divide two cells	=(C2)/3 =(C2)/(A1)

Applied Epic/Microsoft® Word® - IF, ASK, Math, SET, Bookmarks, INCLUDETEXT // PAGE 16

If you are using the Formula button on 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.



Эк

Cancel

MOD

NOT

ROUND SIGN SUM

OR PRODUCT

SESSION HANDOUT

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



Math Sample

Here is a sample of the coding for creating a total of the values in an equipment schedule.

ipmentFl oater.Sc heduled. Schedul ed.Schd ItemNu m*0*13	ann an the second s	auled.SchdSer Num*0*15∙∬¤	entFioater: Scheduled. Scheduled. SchdOwnd* 0*16-}¤	Anfitnis*4*17}". \#•\$#,##x-}⊧
{·DOCVA	RIABLE ENDREP } ¶	<u>.</u>	il.	
д	Д	д	Total:¤	{·=·SUM·(e:e)·¤ \#·\$#,##x·}¤
	and the second s			All and a standard and a

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:

Sched	uled Equipment			<u> </u>
Item #	Description	ID/Serial #	Owned	Amount of Ins.
1	Backhoe	WERT3456	No	\$125,000
2	Digger	WERTY6789	No	\$55,000
			Total:	\$180,000



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 which 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 4 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.



- You want to use a non-schedule field within a schedule for instance in the vehicle schedule you want to show the liability limit next to each vehicle. You cannot put non-schedule 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
MasterAccountMerge.Policy.Line.LOB.BusinessAuto.Coverages.LiabilityC
ombinedSingleLimit.CSL.Limit1*8*42}" }

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

{DOCVARIABLE MasterAccountMerge.Policy.Line.LOB.BusinessAuto.RiskMerge.Vehicles. Vehicle.VehYear*0*102 } { DOCVARIABLE MasterAccountMerge.Policy.Line.LOB.BusinessAuto.RiskMerge.Vehicles. Vehicle.VehMake*0*103 } {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 (taught in another class) so that you only have to update employee information in one place. The SET field transfers the employee code to the reference file.

Servicing Roles

In Applied Epic, we can define as many servicing roles as we want, and give them whatever titles we like. This is a great asset – now you can define a CL producer, a PL producer, and a Benefits producer; and assign a claims manager to an account. But it does have a downside for us when we want to pull the specific names into a proposal or formletter. That is because Applied Epic stores all the service roles and the employees assigned to them together. So we need to use a Replicate command in combination with IF statements to get the specific information. The IF statement looks for the Servicing Role code. We can then use the SET field to pull out all the employee information – code, name, email, etc – and assign those values to bookmarks, which we can use throughout the proposal.



Here is the coding that accomplishes this:

MasterAccountMerge.Account.Client.ServiceBilling.ServRole.ServicingRole.Code*0*2 =·"CCE"·"{}·SET·CCECODE·"{}·DOCVARIABLE·

MasterAccountMerge.Account.Client.ServiceBilling.ServRole.Employee.Code*0*3]" #-SET-CCENAME " DOCVARIABLE-

MasterAccountMerge.Account.Client.ServiceBilling.ServRole.Employee.Name*0*4]"]"·

MasterAccountMerge.Account.Client.ServiceBilling.ServRole.ServicingRole.Code*0*2 = "CSM"." SET.CSMCODE." DOCVARIABLE.

MasterAccountMerge.Account.Client.ServiceBilling.ServRole.Employee.Code*0*3]")

MasterAccountMerge.Account.Client.ServiceBilling.ServRole.Employee.Name*0*4 """. DOCVARIABLE ENDREP

Let's pull it apart and analyze how it works. Immediately after the DOCVARIABLE BEGINREP, there is an IF statement:

IF: "IF: DOCVARIABLE: MasterAccountMerge.Account.Client.ServiceBilling.ServRole.ServicingRole.Code*0*2]" =: "CCE": "IF: SET: CCECODE: "IFDOCVARIABLE: MasterAccountMerge.Account.Client.ServiceBilling.ServRole.Employee.Code*0*3]" IF: SET: CCENAME: "IFDOCVARIABLE: MasterAccountMerge.Account.Client.ServiceBilling.ServRole.Employee.Name*0*4]" IF: MasterAccountMerge.Account.Client.ServiceBilling.ServRole.Employee.Name*0*4]" IF: IF: The serviceBilling.ServRole.Employee.Name*0*4]" IF: The serviceBilling.ServiceBilling.ServiceBilling.Employee.Name*0*4]" IF: The serviceBilling.ServiceBilling.ServiceBilling.ServiceBilling.ServiceBilling.Employee.Name*0*4]" IF: The serviceBilling.ServiceBilling.ServiceBilling.Employee.Name*0*4]" IF: The serviceBilling.Employee.PilleBilling.Employee.PilleBilling.Employee.PilleBilling.Emp

This reads the Servicing Role Code to see if it is equal to CCE (in our example – these codes are agency-specific). If it is, there are two SET fields: the first assigns the employee code to the bookmark CCECODE, and the second assigns the employee name to the bookmark CCENAME. You could add other SET fields here to assign the employee email address, phone extension, or any other information available in the employee file to bookmarks.

Following this IF statement there is another, almost identical IF statement that does the same thing for Servicing Role Code CSM (again, any agency-specific code). If you have more than two servicing role codes in your agency, you would add IF statements for each one, using the same format.

The IF statements are surrounded by BEGIN and END Replicate statements. Note that all line returns have been removed – that's because you don't want this code to cause any "printing" to occur in your document, and line returns will force everything on your



proposal page to move down. As part of that, if you highlight the entire code – including the two Replicate commands – and press Ctrl-F9, you enclose it all in a field that further suppresses line returns.

When this runs at the client level, it will cycle through all the service roles assigned to the account, and pull out the codes, names, etc for each of the roles and create all the bookmarks. You can then use those bookmarks throughout your proposal (repeatedly if you wish) by inserting the {REF bookmarkname } field wherever you wish.

If you put this coding in the first page of your proposal group – it needs to be a page that is always included in the proposal, such as the cover page – you can then use those bookmarks on any of the following pages without having to redefine them.

Pulling signature information into templates from a single reference file using INCLUDETEXT

In Applied Epic there are many pieces of information stored in the employee file that we can pull into the documents – title, email address, phone extension, and so on. However there is still information we might like that isn't in Applied Epic – for instance you may want to insert the signature image for the person signing the letter. We can store all this extra information in a separate reference file, and call the reference file from each of our documents. Once we set this up, when we hire a new employee we only have to update the single reference file and all the documents will pull in the new employee's information.

In Applied EpicOnline, we need to do it slightly differently because it uses Word locally at the workstation – this is described at the appropriate place in the instructions below.

To set it up we do have to edit every one of our documents in which we would like the signature image (or whatever other piece of information that isn't available in Applied Epic) 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.

• Use the technique described earlier to assign the various service role employee codes to bookmarks.

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 Epic datafields in this second document – they won't get converted by the Applied/Word macro. So we need to create a Word bookmark (CCECODE or CSMCODE in the example above), 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\PROPDOC folder, the field reads

{INCLUDETEXT "L:\\DOC\\PROPDOC\\ZXTNSNS.RTF" }

I have put the quotes and backslashes in a larger font to emphasize them in the line above.

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. (Obviously it must be where the actual file resides.) In the example I have created these files in the PROPDOC 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 <u>\\\servername\\etc</u> }. *This is where Applied EpicOnline differs – you must point it to a URL on the server.*



The reference file contains a string of IF statements that read the bookmark name and puts 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 the 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 document (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" "" }

With no line returns between the IF statements. There is a single line return at the very end of the file which will get pulled in, so you do need to take account of that in your individual documents.

Note that the actual info you're pulling in can be multiple lines long – it is line returns **outside** the IF statements that you must avoid.

In our example of pulling the signature in, we would put each signature file in the "XX info goes here" sections of the IF statements in the sample above.

SESSION HANDOUT

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



Appendix

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 Epic Interface Install Free Set Includes instructions for downloading and Importing templates.
- Intermediate Word Formatting Applied Epic includes Tables, Borders & Shading, Section Breaks, headers & footers, etc.
- Leveraging Microsoft[®] Word[®] Applied Epic
- Applied Epic/Microsoft[®] Word[®] IF, ASK, Math, SET, Bookmarks, INCLUDETEXT (Service Roles)
- Applied Epic/Microsoft[®] Word[®] Formatting Numbers, Text & Dates
- Applied Epic Proposal Workshop / Advanced Clinic this session has no handouts and will be based upon questions and issues brought to the group by the attendees.