

# Word Document Assembly Guide



Tabs3 Billing



PracticeMaster



Trust Accounting



Accounts Payable




General Ledger

# Word Document Assembly Guide

Copyright © 2022-2024

Software Technology, LLC  
1621 Cushman Drive  
Lincoln, NE 68512  
(402) 423-1440

**[Tabs3.com](https://www.tabs3.com)**

Tabs3, PracticeMaster, Tabs3Pay, and the “pinwheel” symbol () are registered trademarks of Software Technology, LLC.

Version 2023 (Monday, January 1, 2024)

---

# *Table of Contents*

<b>Purpose of Guide</b> .....	<b>5</b>
<b>Introduction</b> .....	<b>5</b>
Document Assembly Use Cases .....	5
Common Terminology .....	6
Document Assembly Overview .....	8
<b>Setting Up Document Assembly</b> .....	<b>9</b>
Install the Document Assembly Toolbar .....	9
Define the Personal Templates Location .....	10
Configure Template Locations .....	11
<b>Assembling a Document</b> .....	<b>13</b>
Document Assembly Process .....	14
<b>Creating a Document Template</b> .....	<b>17</b>
PracticeMaster Document Assembly Toolbar .....	17
Toolbar Buttons .....	17
Variable .....	18
File .....	25
Logic .....	29
Condition .....	29
Variables .....	30
Operators .....	31
Expression .....	32
Tickler .....	32
Fee Tickler .....	33
Cost Tickler .....	35
Calendar Tickler .....	36
Document Management Tickler .....	36
Include .....	37
Test Assemble .....	38

---

<b>Help</b> .....	39
<b>Step 1: Start a New Word Document Template</b> .....	39
<b>Step 2: Select a Base Letter</b> .....	41
<b>Step 3: Replacing Text with Variables</b> .....	43
<b>Step 4: Adding Logic, Internal Variables, and Ticklers</b> .....	50
<b>Completed Document Assembly Template</b> .....	52
<b>References</b> .....	<b>54</b>
<b>Index</b> .....	<b>56</b>

---

## ***Purpose of Guide***

This Word Document Assembly Guide is a great resource for both users who will be assembling documents, as well as advanced users who need to code document templates. It includes instructions on setting up and using the Word Document Assembly Toolbar, as well as procedures for creating and populating templates, testing, and tips for best use.

All guides are available on our website at:

[\*\*\*Tabs3.com/support/docs.html\*\*\*](http://Tabs3.com/support/docs.html)

## ***Introduction***

PracticeMaster includes the ability to quickly and easily assemble documents in Microsoft® Word using the built-in Word Document Assembly feature. This document assembly engine was written specifically to integrate with Microsoft Word. Specially coded document templates in Word can be used to gather information from your PracticeMaster data files to create draft documents. If the information required is not in the data files, document assembly will pause allowing you to enter the required information. This information is then saved to the document *and* written to the PracticeMaster data files.

## ***Document Assembly Use Cases***

Ron is a lawyer at a firm that uses PracticeMaster Word Document Assembly to create documents. Ron needs to create a form letter requesting medical information to send to a client. Ron clicks the **Word Document Assembly** button on the toolbar in PracticeMaster's Matter Manager (*In the Quick Launch, search for and select "Matter Manager"*) which prompts him to select the document template. He selects Medical Information Request and begins the assembly. Because all of the information for the selected client is already stored by PracticeMaster, it populates in the document and is ready to send in moments.

As another example, Wendy is a lawyer at a firm that uses Document Assembly. Wendy is opening a matter for a new client. Wendy fills in all of the information for the matter and clicks Save. The firm has a WorkFlow configured to assemble a Fee Agreement document, so as soon as the matter is saved the document is assembled automatically. Wendy can then print the document and have the new client sign it right away.

---

# Common Terminology

The following is a list of terms and features used in Document Assembly:

<b>Word Document Assembly Toolbar</b>	PracticeMaster's Word Document Assembly feature uses templates to assemble documents by pulling information out of PracticeMaster into Word using variables, creating logic conditions, and/or including additional document templates.
<b>Word Template</b>	A file that searches as a starting point for a new document. A template can contain formatting styles, text, and images, as well as variables and logic conditions.
<b>Word Document</b>	A file that contains a completed document, either created directly in Microsoft Word or after completing the assembly process. Word Documents are stored as .DOC, .DOCX, or .DOCM files.
<b>Document Management record</b>	A record in the PracticeMaster Document Management file that is used to link a document to any client or contact. Assembled documents can optionally be saved as Document Management records.
<b>Variable</b>	A value that is potentially different for each assembled document, such as who the document is being sent to or on behalf of, the date the document was sent, etc. This includes information stored in PracticeMaster data files, as well as information specified in the document assembly template or entered during the assembly process.
<b>File</b>	PracticeMaster consists of a database that includes various files that are used to record different kinds of information. Many of these files are predefined and are provided with PracticeMaster, such as the Client and Contact files. In addition to these predefined files, you can also create your own files.
<b>Record</b>	A record consists of multiple fields of information. A data file consists of a group of records such as client records in the client file. Each client record consists of fields such as work description, assigned timekeeper, and date opened.

---

**Logic**

PracticeMaster's Word Document Assembly allows you to use conditional logic when coding your templates. Conditional logic gives you the ability to control what action is taken if a particular condition is true or false. Some reasons why you would use conditional logic include:

- A conditional statement can be used in conjunction with a file access code to automatically select records that meet a specific set of criteria. For example, by using a conditional statement, you can send a request for medical records to all providers in a file who have a blank date in the Date Received field. As another example, you can print a letter to all share holders who live in California or Washington.
- A conditional statement can be used to determine if a particular paragraph, clause, or file should be included in a document. For example, if the State of jurisdiction is New Jersey, include a specific clause. As another example, if the marital status is "M," include a Spouse paragraph. As another example, if the number of children is greater than 0, include a Bequest to Children file. The Include File function is frequently used in conjunction with conditional logic.

**Tickler**

A tickler is a special code used in the document assembly program to start a new PracticeMaster record. There are four types of ticklers: Fee ticklers, Expense ticklers, Calendar ticklers, and Document Management ticklers. When a tickler is encountered during Document Assembly, the appropriate file is automatically opened and the user is allowed to enter a record.

**Include**

Document assembly templates can include another PracticeMaster document template file (\*.DOT or \*.DOTM) or other file, allowing you to reuse common information such as headers, address blocks, and other information that rarely changes. Use of include files can make maintenance of document assembly libraries easier by limiting how many document assembly templates need to be edited when information changes.

---

**Test**

Testing the coding of document assembly templates checks for coding errors. It is recommended that you test a document assembly template prior to assembling a document.

**Assembly Error**

Errors can be encountered while assembling or testing a document assembly template. This includes coding errors, logic errors, or file access errors. Any errors must be corrected in the document assembly template before a document can be assembled.

## Document Assembly Overview

The following overview is provided to give a better understanding of how all of the specific parts of PracticeMaster document assembly work together to help your firm.

**To assemble documents:**

- **Install the Word Document Assembly Toolbar Plug-in.** *(page 9)*
- **Configure document template location(s).** *(page 11)*
  - The default folder when assembling document templates can be set for each Area of Practice (AOP), with a default location for matters without an AOP assigned or assembled without a client selected.  
If starter data was installed with PracticeMaster, several example template are included in the CMSYSTEM\Docs folder.
- **Assemble documents as needed.** *(page 13)*
  - Creating a customized document for a matter using document assembly is simple. You can easily generate a document anytime one is needed, but using WorkFlows can help you seamlessly create a document when you know one will be needed. This helps eliminate errors, and prevents necessary documents from being forgotten. Additional information on using WorkFlows can be found in the [\*\*WorkFlows Guide\*\*](#).

**To create document assembly templates:**

- **Install the Word Document Assembly Toolbar Plug-in.** *(page 9)*
- **Create a library of document assembly templates.** *(page 17)*
  - The easiest way to create a document assembly template is to copy the text from a letter that you often have to send on behalf of your clients.
  - Document assembly templates can be specific to an Area of Practice (AOP), such as a request for medical records used in Workers' Compensation cases, or can be general and used in multiple Areas of Practice, such as a fee agreement sent to all clients.



---

# Setting Up Document Assembly

Each user who will be assembling or creating document templates will first need to install the Document Assembly Toolbar Plug-in.

Additionally, users must configure the **Default personal templates location** in Word Options to utilize the installed document templates.

## Install the Document Assembly Toolbar

Make sure to close Word before installing the toolbar, or close and reopen Word after the toolbar is installed.

### ► To install the Toolbar Plug-in

1. In the PracticeMaster Quick Launch, search for and select "Toolbar Plug-ins."
2. In the Toolbar Plug-ins window, select the **Word** tab. The **Word** tab is shown in Figure 1.

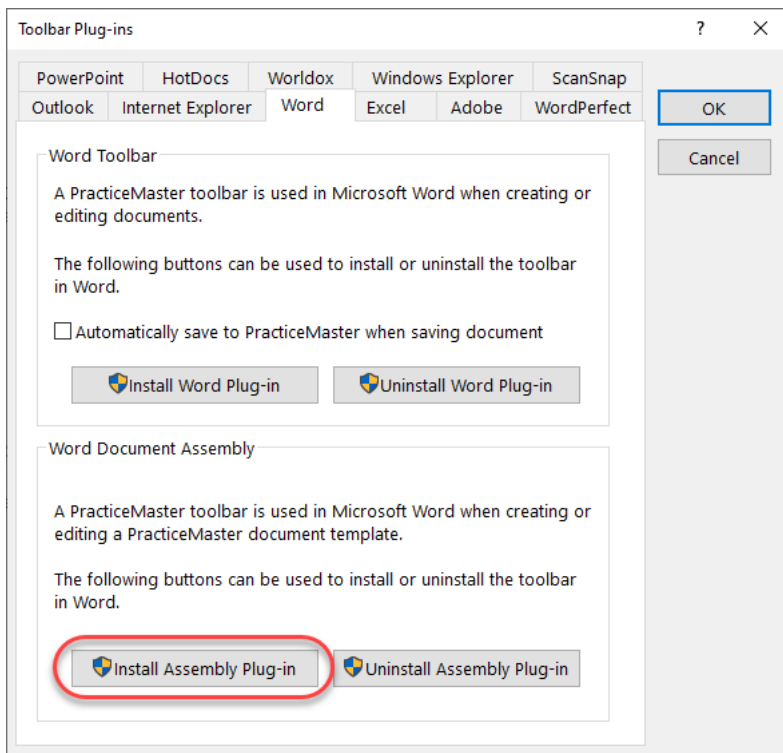


Figure 1, **Word** tab of Toolbar Plug-ins

3. Click **Install Assembly Plug-in**. This process copies template files to the local computer and creates registry entries, so you may need to provide elevated access depending on your system rights. When installation is complete, a message is displayed.
4. Click **OK** to close the message.
5. Click **OK** to close the Toolbar Plug-ins window.

**Note:** Every PracticeMaster user who will be assembling documents **must** install the toolbar on their local workstation.

## Define the Personal Templates Location

The **Default personal templates location** field in Word Options must be defined.

### ► To define the Default personal templates location field

1. In Word, click **File**.
2. Click **Options** and then select **Save**. The Word Options window is shown in Figure 2.

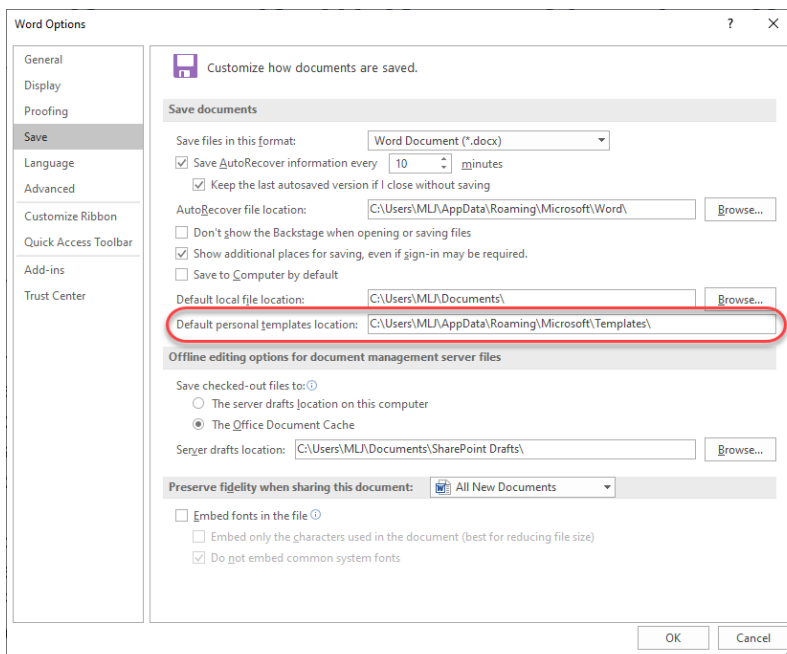


Figure 2, Word Options window

3. In the **Default personal templates location** enter "C:\Users\\AppData\Roaming\Microsoft\Templates". Replace <User Name> with the Windows user currently logged in. For example, if your computer login name is Ron, enter

---

"C:\Users\Ron\AppData\Roaming\Microsoft\Templates".

4. Click **OK** to close the Word Options window.

## Configure Template Locations

When using the Word Document Assembly feature in PracticeMaster, you will be prompted to select a client. After selecting a client, you must select a template. The default directory shown when selecting a template is based on the Template Path specified for the selected client file's Area of Practice (AOP). If no client file is specified, the default directory shown when assembling a document will be the Template Path specified for the System Files Area of Practice.

**Note:** If you change the **Template Path** for an existing Area of Practice, you may need to copy the document template files to the new subdirectory. The files are not transferred by PracticeMaster.

### ► To change the template location for an Area of Practice

1. In the PracticeMaster Quick Launch, search for and select "File Maintenance."
2. Expand the **Areas of Practice** folder, then right-click the desired Area of Practice.
3. Select **Edit AOP Information**.
4. In the **Template Path** field, enter the template location. By default, this is a folder called DOCS under the Area of Practice folder. For example, for an AOP of FAMILY, the default Template Path will be FAMILY\DOCS.

**Note:** The **Template Path** is relative to PracticeMaster's Current Working Directory, which contains all of the data files for the PracticeMaster software. To determine the location of the Current Working Directory, in the Quick Launch search for and select "About PracticeMaster."

5. Click **OK** to close the Edit Area of Practice window.
6. Click **OK** to close the File Maintenance window.

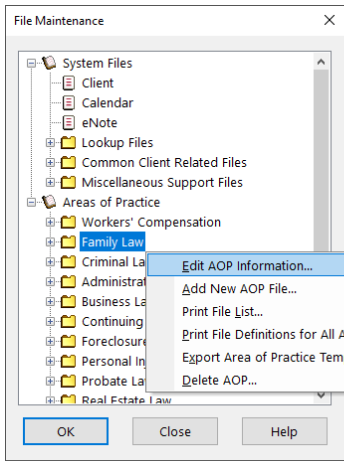


Figure 3, File Maintenance

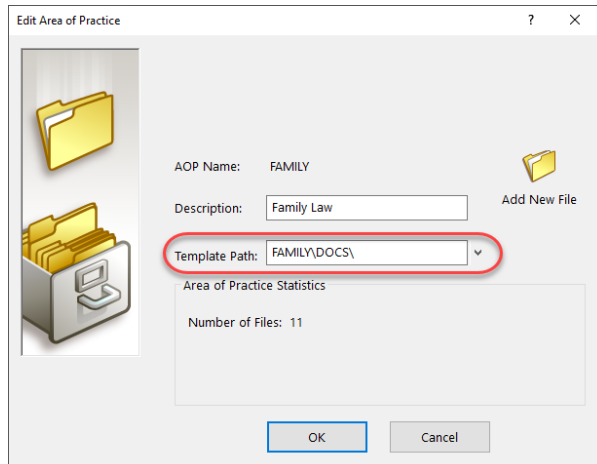


Figure 4, Edit AOP window

► **To change the template location for clients without an Area of Practice**

1. In the PracticeMaster Quick Launch, search for and select "File Maintenance."
2. Right-click the **System Files** folder.
3. Select **Edit AOP Information**.
4. In the **Template Path** field, enter the template location. By default, this is the CMSYSTEM\DOCS folder.
5. Click **OK** to close the Edit AOP Information window.
6. Click **OK** to close the File Maintenance window.

# Assembling a Document

Word Document Assembly can be started from multiple locations in the software, including the Quick Launch, the Documents menu, from All Actions or My Actions on the Home page, the Quick Clicks pane in the Client file, and in the Matter Manager. You can see these launch points in the figures below.

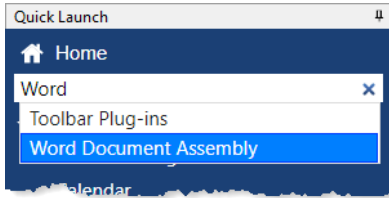


Figure 5, Quick Launch

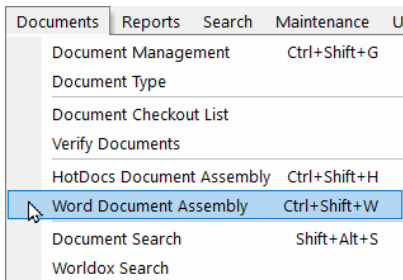


Figure 6, Documents Menu

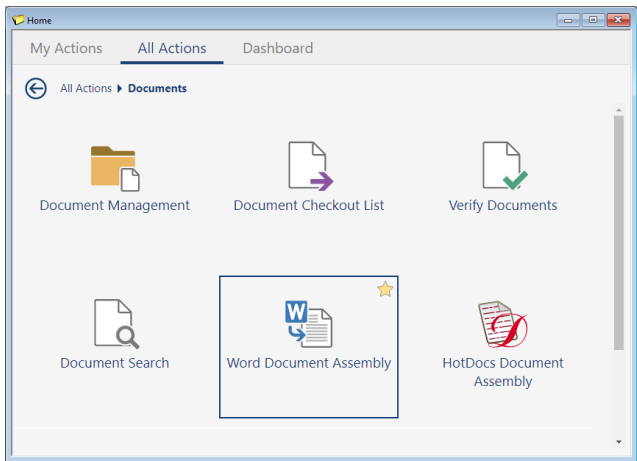


Figure 7, All Actions on the Home page

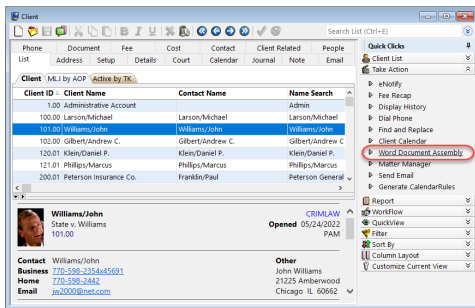


Figure 8, Client file

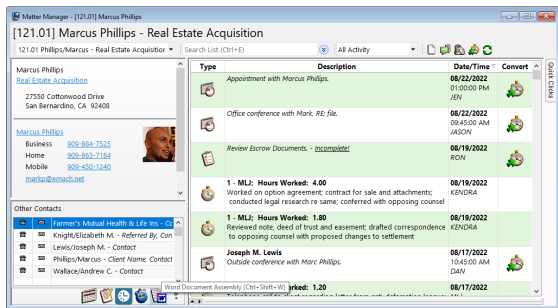


Figure 9, Matter Manager

**Note:** We recommend loading Microsoft Word before starting the document assembly process.

## Document Assembly Process

After selecting to start Word Document Assembly, you will be prompted to select a specific client and a document template.

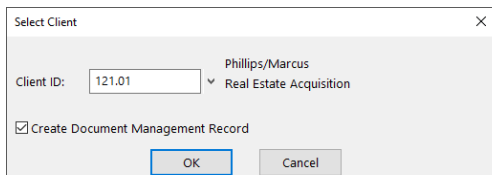


Figure 10, Select Client window

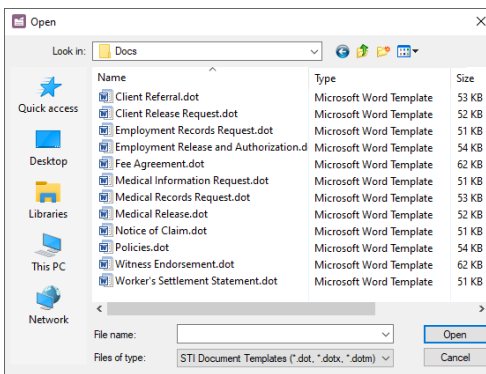


Figure 11, Select Document Template window

### ► To begin document assembly

1. Open Word Document Assembly using one of the other options shown on page 13.
2. In the Select Client window, optionally specify the **Client ID** for the matter the document is being assembled for. The Select Client window is shown in Figure 10.

**Note:** When started from the Matter Manager or the Client file, the document will be assembled for the currently selected matter. Therefore, the Select Client window will not be displayed.

**Note:** To store the assembled document so it is accessible from within PracticeMaster after assembly is complete, select the **Create Document Management Record** check box on the Select Client window or use the **PM Save** button on the Word Toolbar. PracticeMaster can optionally manage the location and file name of documents saved to the Document Management file. For additional information on PracticeMaster's Document Management feature, see the **Document Management Guide**.

2. Click **OK**.
3. In the Select Document Template window, specify the desired PracticeMaster Document Template file. Word templates have a .DOT, .DOTX, or .DOTM file extension. The Select Document Template window is shown in Figure 11.
4. Click **OK**.

**Note:** By default, the document directory specified for the client's Area of Practice will be shown in the Select Document Template window. However, if no client is specified, then the CMSYSTEM\DOCS directory (under the PracticeMaster Current Working Directory) will be shown as the default directory. Many firms use this directory as a storage area for documents that can be used for all clients that are not related to a specific Area of Practice.

Once the document assembly process begins, several optional prompts can be displayed depending on the codes that have been used in the document template. User interaction may or may not be required depending on the codes used in the template and the data present in the client's data files.

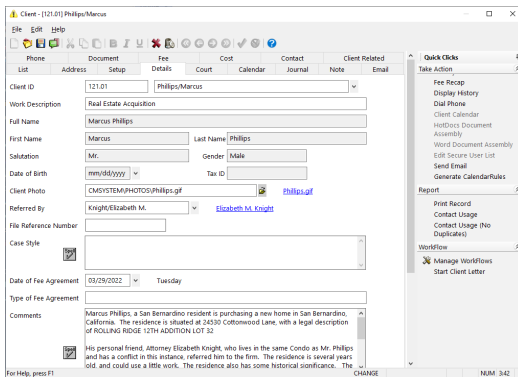


Figure 12, A PracticeMaster entry window, like the Client file shown here, is displayed if a required record or field is missing

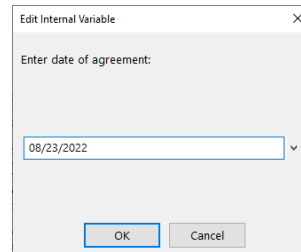


Figure 13, The Edit Internal Variable window is displayed when the template has been configured to prompt for information

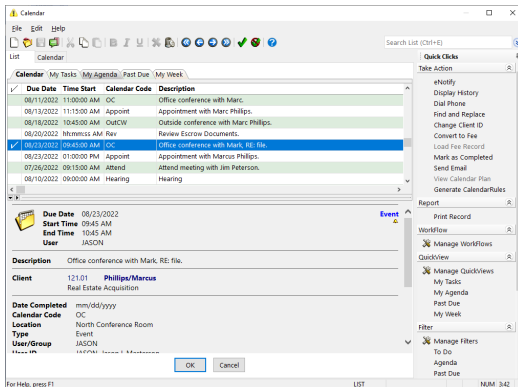


Figure 14, The Select Record window is displayed when the template has been configured to prompt the user to select the desired record

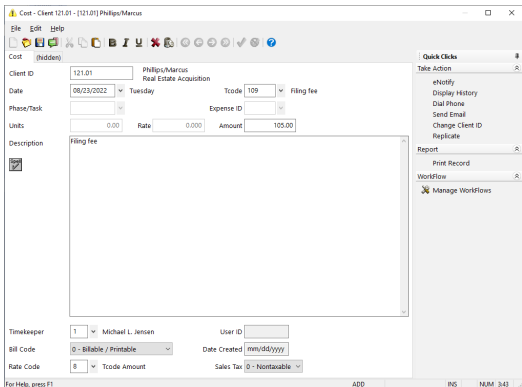
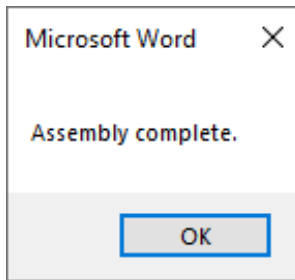


Figure 15, A PracticeMaster Fee, Cost, Calendar, or Document Management entry window is displayed if the template has been coded with a tickler to create the entry

---

After the Document Assembly process has completed, the message "Assembly complete" will be displayed. Click **OK** to close this message.



*Figure 16, Assembly complete message*



---

# Creating a Document Template

The easiest way to create a document assembly template is to copy the text from a letter that you often have to send on behalf of your clients and paste it into a new template.

1. Create a template.
2. Open the existing letter in Word.
3. Select all text (*Ctrl+A*).
4. Copy the text to the clipboard (*Ctrl+C*).
5. Close the letter and return to the template.
6. Paste the text into the template (*Ctrl+V*).

## PracticeMaster Document Assembly Toolbar

The PracticeMaster Document Assembly Toolbar is available in Microsoft Word when creating or editing a PracticeMaster document template.

**Note:** When coding document templates in Word, the workstation coding the document cannot have any other functions in PracticeMaster open. However, other users can access other functions in PracticeMaster.

**Note:** The Document Assembly toolbar will only be displayed when editing a template file (.DOT or .DOTM) based on CMTEMPLT. It will not be displayed while working with normal Word documents (.DOC or .DOCX).

## Toolbar Buttons

The buttons on the PracticeMaster Document Assembly toolbar are used to insert, modify, and test code that controls the contents of documents during the assembly process.

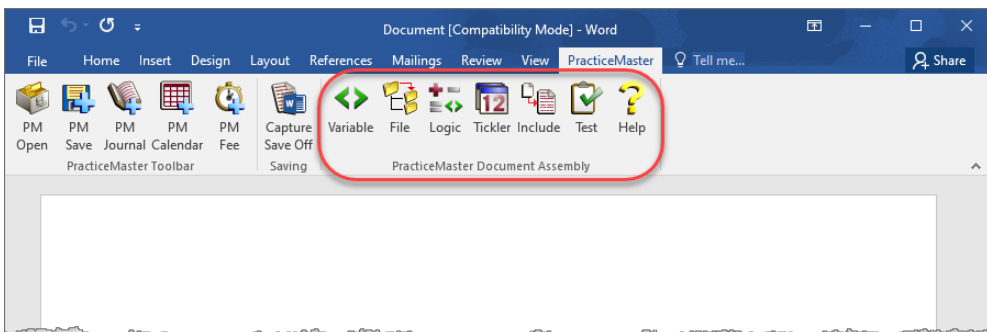
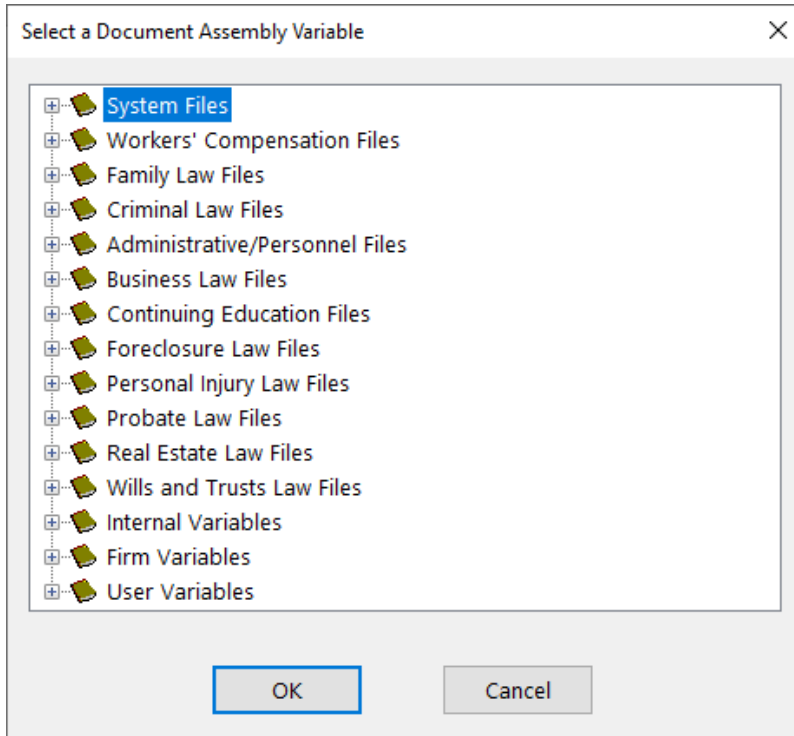


Figure 17, PracticeMaster Document Assembly toolbar in Microsoft Word 2016

---

## Variable

The **Variable** button (or Alt+R) is used to insert a variable from PracticeMaster into a document template. When selecting this option, you will be allowed to select the variable, as shown in Figure 18.



*Figure 18, Select a Document Assembly Variable window*

This window uses a file tree structure for selecting variables. Clicking the plus (+) symbol to the left of a branch expands the branch to display the various files or fields. Likewise, clicking the minus (-) symbol to the left of a branch will collapse its contents.

### **System Files**

This branch lets you select a field from the desired System file.

### **Area of Practice Files**

These branches let you select a field from the desired Area of Practice file.

---

**Internal Variables**

This branch lets you select the type of internal variable you want defined. Internal variables are used when you want the user to input the value during the document assembly process. This value is used in the current document only and is not written back to the PracticeMaster data files. Internal variable types include numeric, alpha, date, time, and boolean.

**Firm Variables**

This branch lets you select a field from the Firm Information file that is stored in System Configuration. Variables that can be selected include the firm's name, address, city/state/zip, telephone, fax, email, website, state, county, and Federal ID.

**User Variables**

This branch lets you select a field from the User Configuration file that is stored in System Configuration. Variables that can be selected include the User ID, name, initials, and email address.

Once the variable has been selected, the Variable Properties and Format window will be displayed allowing you to define the variable, as shown in Figure 19.

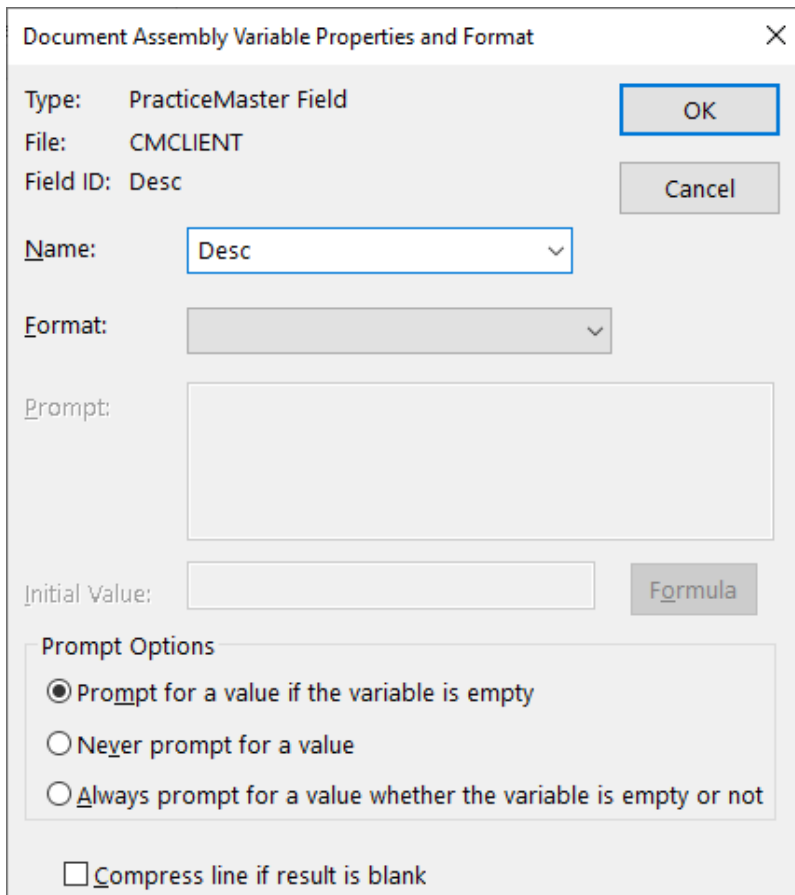


Figure 19, Variable Properties and Format window

**Type**

The selected variable is shown for display purposes only and cannot be edited.

**Name**

Each variable must be assigned a name. Characters that can be used include letters, numbers, and the underscore character (i.e., "\_"). Uppercase and lowercase letters can be used. The first character must be a letter. Variable names are case insensitive. The maximum number of characters is 25. The default name for PracticeMaster fields and User and Firm variables is the field name. You can accept the default name or change it as desired.

---

## Format

The available formats depend on the specific variable type. If the **Format** field is left blank, the value of the variable will be output based on how it is stored in PracticeMaster or the default format. If **Do not insert text into document** is selected, the value of a variable can be retrieved and stored for use in logic statements but the value is not actually printed in the document.

Alpha fields can optionally display their text as **all lowercase**, **Capitalize first word only**, **Capitalize First Letter Of All Words**, or **ALL UPPERCASE**. The default format when a blank format is selected is how it is stored in the field.

Date fields can optionally display their values as **09/08/2022**, **09/08/22**, **9/8/22**, **8 SEP 22**, **8 September 2022**, **September 8, 2022**, **Saturday, Thursday, September 8, 2022**, **8th day of September, 2022**, or **September 8th**. The default format when a blank format is selected is **09/08/2022**.

Boolean fields can optionally display their value as **Y/N**, **Yes/No**, **True/False**, or **1/0**. The default format when a blank format is selected is **Yes/No**.

Number and money fields can optionally display their values as **9999**, **9,999**, **\$9999**, **\$9,999**, **9999.0**, **9,999.0**, **9999.00**, **9,999.00**, **\$9999.00**, **\$9,999.00**, **9999.000**, **9,999.00**, **9999.0000**, or **9,999.0000**. The default format for number variables is **9999**. The default format for money variables is **9999.00**.

Time fields can optionally display their value as **01:00 PM**, **01:00p**, **1:00 P.M.**, **1:00 PM**, **1:00 p.m.**, or **1:00 pm**. The default format when a blank format is selected is **01:00 PM**.

## Prompt

The **Prompt** field is used only for internal variables. Enter the prompt text which will be shown in an Edit Internal Variable window during document assembly. If no **Prompt** is specified, then the name of the internal variable will be shown in the Edit Internal Variable window instead.

---

## Initial Value

The **Initial Value** field is used only for internal variables. This field is optional. If desired, you can enter a default value so that when document assembly is paused, the Initial Value will be shown allowing the user to accept or overwrite it. There is no limit to the size of the Initial Value.

When entering an initial value for an alpha internal variable, you must enter the value within double quotes. For example, if the initial value is New York, you would enter "New York".

When entering an initial value for a Boolean internal variable, you must enter either 1 (yes) or 0 (no). For example, if the initial value is true or yes, you would enter 1.

Click the **Formula** button to open the Document Assembly Formula window, as shown in Figure 20. This window is used to define a formula that can be used in the Initial Value of an internal variable. For additional information on creating an expression in the Document Assembly Formula window, see KB Article [R11493](#), "All About Expressions."

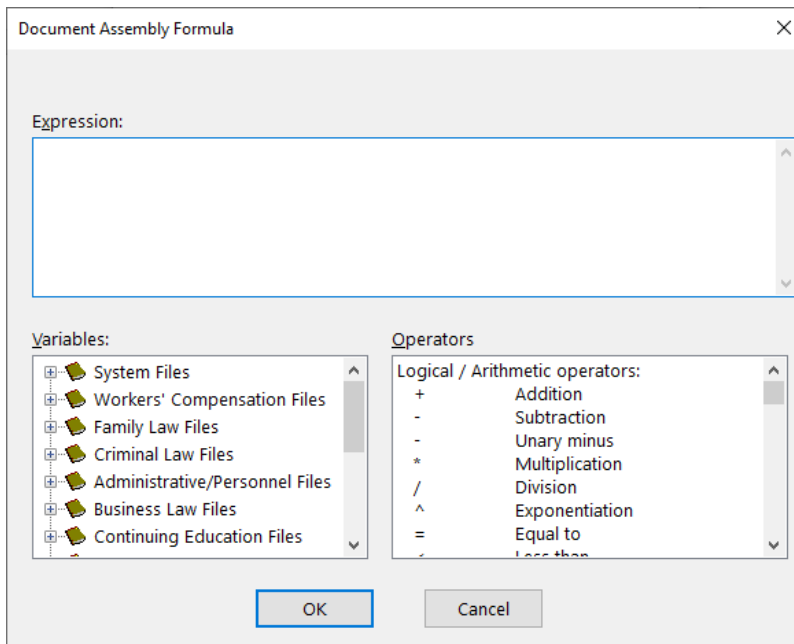


Figure 20, Document Assembly Formula window

---

## Prompt Options

Designates whether the document assembly process should be paused when the variable is encountered. If document assembly is paused for a PracticeMaster field, the appropriate Editor window will be displayed with a blinking cursor positioned in the designated field. If document assembly is paused for an Internal Variable, the Prompt specified for the variable properties will be displayed in an Edit Internal Variable window allowing the user to enter the desired information.

**Note:** Prompt options are not available for Firm Variables or User Variables. Document assembly cannot write information back to these files because this information is stored in System Configuration data files. Similarly, prompt options are not available for virtual fields.

Select **Prompt for a value if the variable is empty** if you want document assembly to pause only when the variable is empty. Selecting this option indicates you want the user to have the opportunity to enter information if the field is empty; however, you do not want the user to verify information in the field, nor do you want the user to be given the opportunity to change the information. Information entered when document assembly is paused will be written to the PracticeMaster data files.

**Note:** By default, money and number fields have a value of 0 and therefore cannot be considered empty. Likewise, boolean fields have a default value of No and therefore cannot be considered empty.

**Note:** All internal variables that have a blank Initial Value are considered empty. Therefore, document assembly will pause for these variables when this option is selected. Conversely, if an Initial Value is specified and this option is selected, document assembly will not pause for the user to enter a value.

---

**Note:** When a field's value is not defined, and document assembly prompts you to provide a value, restricted fields will not be editable if you do not have access rights to edit restricted fields. Document assembly will continue once the prompt is closed, but no value will be specified for the restricted field in the document.

Select **Never prompt for a value** if you do not want document assembly paused when this variable is encountered. This option is particularly useful when you want to assign the contents of an internal variable.

Select **Always prompt for a value whether the variable is empty or not** if you want the user to always be prompted for information during document assembly regardless of whether this field is blank or not. This option is useful when you want the user to verify or modify the information in the field.

### **Compress line if result is blank**

This option is useful when a carriage return follows a field. If the field is empty, a blank line will print due to the carriage return. This option allows you to suppress the blank line. This option is particularly useful when extra address lines are not used in an address block.

The **OK** button will close the window and add the variable to the Word document. PracticeMaster document assembly codes are shown within chevrons and are typically displayed in blue. Examples of how coded variables might look are:

```
«Name|CMSYSTEM|CMCLIENT:Name» and  
«Addr2|CMSYSTEM|CMCLIENT:Addr2|ALL UPPERCASE»  
«Prim_Tkpr_Empl|CMSYSTEM|CMCLIENT:Prim_Tkpr:Empl|9999»
```

If Microsoft Word is configured to show hidden text (in Word, select **File | Options | Display**), then the same coded variables above might look like:

```
«Var|0|2|0|||0|%H|Name|CMSYSTEM|CMCLIENT:Name» and  
«Var|0|1|1|||0|%H|Addr2|CMSYSTEM|CMCLIENT:Addr2|ALL UPPERCASE»  
«Var|0|0|2|||0|%H|Prim_Tkpr_Empl|CMSYSTEM|CMCLIENT:Prim_  
Tkpr:Empl|9999»
```

The **Cancel** button will close the window without adding the variable to the Word document.



---

## File

This button (or Alt+S) opens the File Access window, as shown in Figure 21. The File Access window allows you to insert an opening or ending file access code so records can be retrieved from a PracticeMaster file. These codes are necessary for the PracticeMaster file to be opened during the document assembly process and are used as a means for the user to indicate which records should be used. More importantly, these codes determine which portion of the document is repeated for each selected record.

File Access codes are always used in pairs and require an opening and closing code. File Access codes are required for all files except user variables, firm variables, or internal variables.

**Note:** File Access codes are not required for the Client File if the document is being written for a single client. This is because the client is selected when starting to assemble a document.

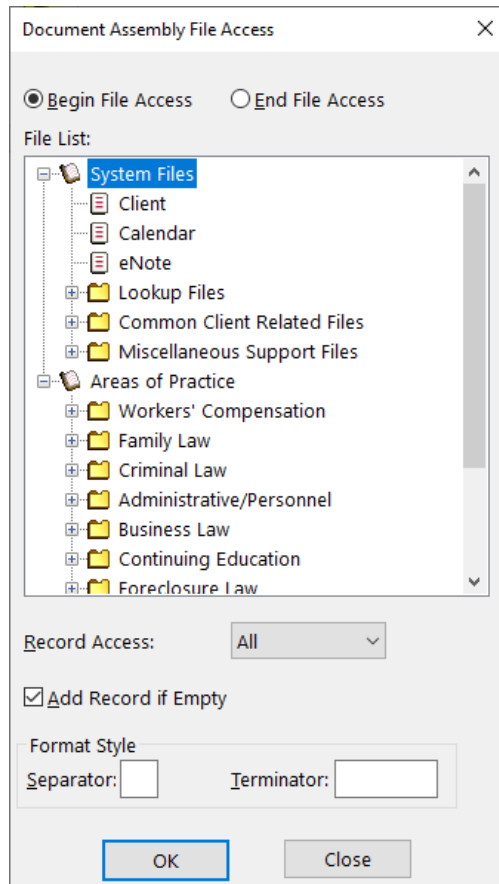


Figure 21, File Access window

---

### Begin File Access

Select **Begin File Access** if you are inserting the opening access code.

### End File Access

Select **End File Access** if you are inserting the closing access code. The **Record Access** and **Add Record if Empty** options are unavailable when inserting an End File Access code.

### File List

Select the file you want to access by clicking it. The File List, as shown in Figure 22, consists of a file tree structure that includes all files in PracticeMaster. Clicking the plus (+) symbol to the left of a branch expands the branch to display the various files. Likewise, clicking the minus (-) symbol to the left of a branch will collapse its contents.

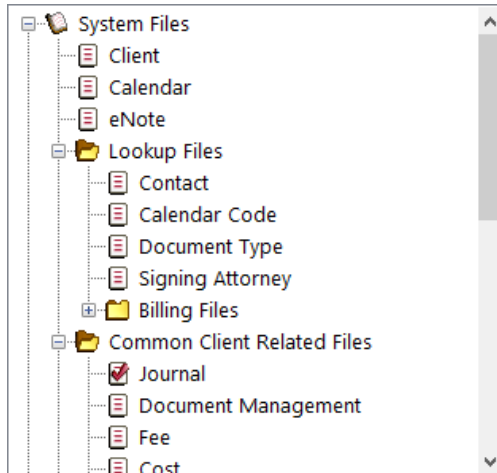


Figure 22, File List

### Record Access

When opening access to a file, you have the option to select the type of Record Access.

Select **All** if you want all records selected. The portion of the document within the beginning and ending access codes will be repeated for each record in the file. When using the All option, the user will not be asked to select a record during document assembly.

---

**Tip:** When generating a separate document for all records in a file, place a page break before the ending access code. This will cause each document to begin on a new page.

Select **First** if you want the first record in the file selected. The information from the first record added to the file will automatically be selected. When using the **First** option, the user will not be asked to select a record during document assembly.

Select **Last** if you want the last record in the file selected. The information from the most recently added record will automatically be selected. When using the **Last** option, the user will not be asked to select a record during document assembly.

**Note:** The First and Last options refer to the primary index order. For example, the primary index for the Contact file is the Contact ID field, so First and Last for this file refers to the alpha order of the Contact IDs.

Choose **Select** if you want document assembly to pause and allow the user to select the desired record(s).

### **Add Record if Empty**

Select the **Add Record if Empty** check box if you want document assembly to pause and allow the user to add a new record when the specified file has no records in it. Clear this check box if you want document assembly to skip the field variables when no record exists in the file. All references to fields in this file will be skipped.

### **Format Style**

The **Format Style** setting lets you define a Separator and Terminator that can be used when printing a series of records to specify which punctuation and terminator should be used between items. These codes are used to program for the possibility that the number of selected records can vary each time the document is assembled.

A single **Separator** character can be selected. The most commonly used character is a comma or semi-colon. The **Separator** prints following each item in a series with the exception of the last item.

---

The **Terminator** represents one or more words that are used before the last item in the series. The most commonly used word is "and". Other possibilities include but are not limited to "or" and "nor".

These fields are used during document assembly depending on how many records are selected. When only one record is selected, just the item is printed; neither the Separator nor Terminator is used. When two records are selected, the Terminator is used between the two items. When three or more records are selected, the Separator is printed after each item except the last item. The Separator followed by the Terminator print following the second to last item in a series. When no record is selected, neither the Separator nor Terminator is used.

For example, specifying a Separator of a "," (comma) and a Terminator of "and" could yield the following results: "Marcia Brady", "Marcia Brady and Jan Brady", or "Marcia Brady, Jan Brady, Cindy Brady, and Bobby Brady".

**Note:** File Access codes can be used within a table; however, there may be situations where unexpected results may occur. Generally, if both the Begin File Access and End File Access codes are placed in the same cell, the coding will be performed as desired. Placing an entire table within a File Access code performs as expected.

**Note:** File Access codes can be nested with a maximum limit of 20 nested codes.

The **OK** button will close the window and add the designated file access code. PracticeMaster document assembly codes are shown within chevrons and are typically displayed in blue. Examples of how coded variables might look are:

```
«Begin Access|CMSYSTEM|CMFEE|All|Yes» .....  
«EndAccess»
```

If Microsoft Word is configured to show hidden text (in Word, select **File | Options | Display**), then the same coded variables above might look like:

```
«FileOpen|%H|Begin Access|CMSYSTEM|CMFEE|All|Yes» .....  
«FileClose|%H|EndAccess»
```

The **Cancel** button will close the window without adding the designated file access code.

---

## Logic

This button (or Alt+L) is used to open the Logic window, as shown in Figure 23, allowing you to insert conditional logic statements. Conditional logic gives you the ability to control what action is taken if a particular condition is true or false.

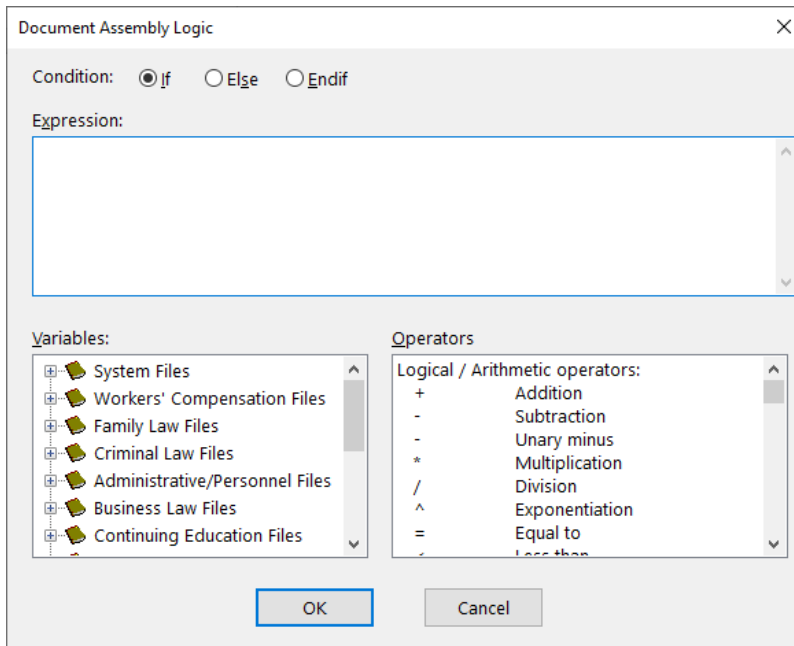


Figure 23, Logic window

The Logic window consists of the following options:

### Condition

PracticeMaster supports three conditions that can be used in a conditional logic statement. Each expression uses only one condition.

#### If

The **If** condition is used for the opening condition in a conditional logic statement. All conditional statements always start with If.

---

## Else

The **Else** condition is optional and has two uses.

The **Else** condition can be used to indicate an alternate result. For example, if A is true, do B; else do C.

The **Else** condition can be followed by another If condition to work as the OR portion of a conditional statement. For example, if A is true or if B is true, do C.

**Note:** When **Else** is selected, the Expression, Variables, and Operators area of the Logic window are dimmed.

## Endif

The **Endif** condition is used as the closing code of a conditional statement. All conditional statements always end with an expression that consists of an **Endif** condition. Complex conditional statements may have multiple **Endifs**.

**Note:** When **Endif** is selected, the Expression, Variables, and Operators area of the Logic window are dimmed.

Nested conditional statements are allowed. There is a maximum of 20 nested IF statements. An example of how coded a condition might look is:

```
«If CMSYSTEM|CMCLIENT:Inactive» .....  
«Else» .....  
«Endif»
```

## Variables

A Variable is used to select a field you want used in a formula or conditional statement. All of the various data files in PracticeMaster and other information can be accessed from this window. Once the desired variable is shown, double-click it to place it in the Expression field.

The Variables portion of this window uses a tree structure for selecting fields. Clicking the plus (+) symbol to the left of a branch expands the branch to display the various files or fields. Likewise, clicking the minus (-) symbol to the left of a branch will collapse its contents.

**Alternate Keystrokes Tip:** Pressing the *Right-Arrow* key or the *Plus* sign on the numeric keypad are alternate methods that can be used to expand a branch. Likewise, pressing the *Left-Arrow* key or the *Minus* sign on the numeric keypad will collapse a branch.

## System Files

This branch lets you select a field from the desired System File.

---

## Area of Practice Files

This branch lets you select a field from the desired Area of Practice file.

## Internal Variables

This branch lets you select any internal variable that has been defined.

**Note:** If the same internal variable has been used multiple times, it will be shown only once in the list and the formatting options from the first time the variable was assigned will be used.

## Firm Variables

This branch lets you select a field from the Firm Information file that is stored in System Configuration.

## User Variables

This branch lets you select a field from the User Configuration file that is stored in System Configuration.

## Operators

The Operators portion of the window lists all of the available operators and functions that can be used in an expression.

Double-click the appropriate operator to place it in the Expression field.

### + Addition

Used to add (e.g.,  $x + y$ ).

### - Subtraction

Used to subtract (e.g.,  $x - y$ ).

### - Unary Minus

Used to denote a negative number (e.g.,  $-x$ ).

### \* Multiplication

Used to multiply (e.g.,  $x * y$ ).

### / Division

Used to divide (e.g.,  $x / y$ ).

### ^ Exponentiation

Used to indicate that a number is an exponent (e.g.,  $x^3$ ).

### = Equals to

Used to compare equality between two values (e.g.,  $x = y$ ).

### < Less than

Used to compare if the first value is less than the second value (e.g.,  $x < y$ ).

### <= Less than or equal to

Used to compare if the first value is less than or equal to the second value (e.g.,  $x <= y$ ).

### != Not equal to

Used to compare two values and test whether they are not

### <> Not equal to

equal (e.g.,  $x != y$ , or  $x <> y$ ).

---

<b>&gt;= Greater than or equal to</b>	Used to compare if the first value is greater than or equal to the second value (e.g., x >= y).
<b>&gt; Greater than</b>	Used to compare if the first value is greater than the second value (e.g., x > y).
<b>CONTAINS</b>	Variable contains string (e.g., STATE contains "NY").
<b>IN</b>	String within variable (e.g., "NY" in STATE). The string must appear first.
<b>AND</b>	Logical and (e.g., x and y)
<b>NOT</b>	Logical negation (e.g., not INACTIVE)
<b>OR</b>	Logical or (e.g., x or y)

## Expression

Various Conditions, Variables, and Operators can be used to create an **Expression**. If an expression is evaluated as being true during document assembly, the portion of the document following the expression will be performed up until the ending code. Otherwise, it will be skipped. An expression is created by selecting a condition and then adding the required variables and operators. If desired, you can type in the expression. Expressions are case insensitive.

**Note:** Conditional logic statements can be included within a table; however, there may be situations where unexpected results may occur. Generally, if the entire expression is within a single cell, the coding will perform as desired.

The **OK** button can be used to add the logic code to the document.

The **Cancel** button will close the Logic window without adding information to the document.

An example of how a coded expression might look is:

```
«If CMSYSTEM|CMCLIENT:Orig_Tkpr:Empl < 4 or CMSYSTEM|CMCLIENT:Orig_
Tkpr:Empl = 9» .....
«Else» .....
«Endif»
```

See KB Article [R10667](#), "Word Document Assembly Advanced Coding Tips," for assistance with conditional logic expressions.

## Tickler

This button (or Alt+K) is used to open the Document Assembly Tickler window, as shown in Figure 24, allowing you to add fee, cost, calendar, and document management ticklers.



Figure 24, Document Assembly Tickler window

Whenever a tickler is encountered during document assembly, the appropriate Editor window will be opened allowing the user to add a record to the appropriate file.

Default values can be specified in the Tickler window when the tickler is defined but they are not required. During document assembly, when a Tickler Code is encountered (with or without defaults coded), the user can 1) confirm the addition of the record by clicking **OK**; 2) modify the record and then confirm the addition of the record; or 3) cancel out of the Tickler window and continue assembly without adding the tickler.

**Tabs3 Billing Note:** Fee and cost ticklers generate records for the fee and cost file which will automatically be added to the Tabs3 Billing fee and cost files during the data synchronization process.

### Fee Tickler

Whenever a fee tickler is encountered during Word document assembly, a fee data entry window will be displayed, allowing the user to add a fee record to the fee file.

---

The following fields can be coded in the Tickler window for a fee tickler. These fields are used as defaults in the data entry window and can be overwritten by the user when the document is assembled.

**Timekeeper**

The **Timekeeper** field lets you specify which timekeeper you want assigned to the fee record. Leave this field blank if you want to specify the primary, secondary, or originating timekeeper as the default timekeeper for the fee record.

**Tcode**

The **Tcode** field lets you specify which tcode will be used for the fee record. If the **Description** in the Tickler window is left blank, then the description for the specified tcode will automatically be shown in the **Description** field when the document is assembled.

**Amount**

The **Amount** field lets you specify the amount you want used for the fee record. This field is particularly useful for value-added billing when a flat amount is charged for a particular type of document.

**Note:** The **Amount** can be omitted so that the default rate can be used to automatically calculate the amount based on the Hours specified.

**Hours to Bill  
Hours Worked**

The **Hours to Bill** and **Hours Worked** fields let you specify the time you want used for the fee record. Hours are assigned using decimal hundredths of an hour. For example, a half hour would be coded as ".5". Likewise, 15 minutes would be coded as "0.25". The leading zero is optional.

**P/S/O Timekeeper**

The **P/S/O Timekeeper** represents the primary, secondary, or originating timekeeper assigned to the client. Using the primary, secondary, or originating timekeeper lets you code a tickler so it can default correctly for all clients without needing to have the user change the timekeeper during document assembly.

The value of this field is used only if the **Timekeeper** field in the Tickler window is blank.

---

**Description**

The **Description** represents the description you want used for the fee record. Leave this field blank if you want the description for the Tcode specified in the Tickler window to be shown in the **Description** field when the document is assembled.

**Cost Tickler**

Whenever a cost tickler is encountered during Word document assembly, a cost data entry window will be displayed, allowing the user to add a cost record to the cost file.

The following fields can be coded in the Tickler window for a cost tickler. These fields are used as defaults in the data entry window and can be overwritten by the user when the document is assembled.

**Timekeeper**

The **Timekeeper** field lets you specify which timekeeper you want assigned to the cost record. Leave this field blank if you want to specify the primary, secondary, or originating timekeeper as the default timekeeper for the cost record.

**Tcode**

The **Tcode** field lets you specify which tcode will be used for the cost record. If the **Description** in the Tickler window is left blank, then the description for the specified tcode will automatically be shown in the **Description** field when the document is assembled.

**Amount**

The **Amount** field lets you specify the amount you want used for the cost record.

**P/S/O Timekeeper**

The **P/S/O Timekeeper** represents the primary, secondary, or originating timekeeper assigned to the client. Using the primary, secondary, or originating timekeeper lets you code a tickler so it can default correctly for all clients without having to have the user change the timekeeper during document assembly.

The value of this field is used only if the **Timekeeper** field in the Tickler window is blank.

**Description**

The **Description** represents the description you want used for the cost record. Leave this field blank if you want the description for the **Tcode** specified in the Tickler window to be shown in the **Description** field when the document is assembled.

---

## Calendar Tickler

Whenever a calendar tickler is encountered during Word document assembly, a calendar data entry window will be displayed allowing the user to add a calendar record to the calendar file.

The following fields can be coded in the Tickler window for a calendar tickler. These fields are used as defaults in the data entry window and can be overwritten by the user when the document is assembled.

### Calendar Code

The **Calendar Code** lets you specify which calendar code will be used for the calendar record. If the **Description** in the Tickler window is left blank, then the description for the specified calendar code will automatically be shown in the **Description** field when the document is assembled.

### Days

The **Days** field represents the number of days following the system date to be used when calculating the default **Due Date** for the calendar record. The default value is "0." If **Days** is "0," the system date will be used as the default **Due Date**. The **Days** field cannot be left blank. A negative number can be used if desired.

### Description

The **Description** represents the description you want used for the calendar record. Leave this field blank if you want the description for the **Calendar Code** specified in the Tickler window to be shown in the **Description** field when the document is assembled.

## Document Management Tickler

Whenever a document management tickler is encountered during document assembly, a document management data entry window will be displayed allowing the user to add a document management record to the document management file.

The following fields can be coded in the Tickler window for a document management tickler. These fields are used as defaults in the data entry window and can be overwritten by the user when the document is assembled.

### Responsible

The **Responsible** field lets you specify which user or group you want assigned to the document management record. The User ID performing the document assembly will automatically be placed in the **Performed By** field.

### Description

The **Description** represents the description you want used for the document management record.

**Note:** Because ticklers run before a document has been saved, PracticeMaster cannot automatically populate the Assembled Document Name field when using a document management tickler. If you want the Assembled Document Name field to be automatically populated, you must use one of the following options:

- Select the **Create Document Management Record** check box when selecting the client during document assembly; or
- Use the **PM Save** button in the Word toolbar; or
- Enable the **Automatically save to PracticeMaster when saving document** option for the Word toolbar.

## Include

This button (or Alt+C) is used to insert another PracticeMaster document template file (\*.DOT) or other file into the current document template. This button opens a standard Windows Open File window, as shown in Figure 25, allowing you to specify which PracticeMaster document template or other file you want included. You can use the "Files of type" drop-down list to specify **Document Templates** or **All Files**. The Open File window will always default to the Current Working Directory.

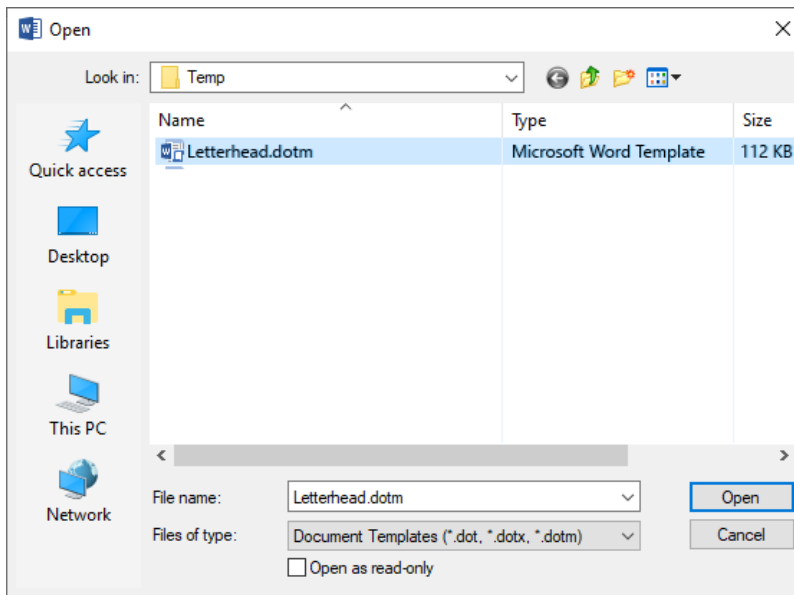


Figure 25, Open File window

If the file specified is located in a directory under the Current Working Directory, then a relative path will be used with the file name. This will eliminate potential problems on networks where mapping may not be the same for all workstations as well as make it easier to move the template from one

---

system to another (for future growth). For example, if PracticeMaster's Current Working Directory is C:\Tabs3 and the template you have included is located in C:\Tabs3\PI\Docs\CLAUSEJ.DOT, then a file name of PI\Docs\CLAUSEJ.DOT will be stored in the template. However, if the template is located in C:\Word\Library\CLAUSEJ.DOT, then the complete filename of C:\Word\Library\CLAUSEJ.DOT will be stored in the template.

An included file can contain other included files (also known as nested includes). There is no limit to the number of nested includes.

The value of internal variables is retained across documents that are assembled in the same document assembly session. For example, if an internal variable is defined in your template, it can be successfully referenced in an Include file.

**Note:** You can include files within a table.

**Tip:** Using the Include Feature to Create Document Sets

If you generate the same set of documents on a regular basis, you can create a template that consists solely of Include codes for each of the document templates within the set. For example, when setting up a new client for a specific Area of Practice, there are usually several documents that are generated at the same time.

**Tip:** Using the Include Feature to Create "Smart" Templates

The Include function makes it easy to create a modular document library. For example, you may want to create document templates for portions or "snippets" of documents and then use those portions as building blocks to build entire documents. This works best for portions of documents that are used over and over again in various types of documents. You may want to consider creating Include files for the various types of documents that you will be coding.

Including files for pages with different page sizes, margins, headers and footers, or styles requires additional coding. See KB Article [R10667](#), "Word Document Assembly Advanced Coding Tips," for more information.

## Test Assemble

This button (or Alt+B) is used to test the coding of your document template to test for coding errors. If errors are encountered, a message will be displayed; otherwise, the complete message will be displayed as shown in Figure 26.

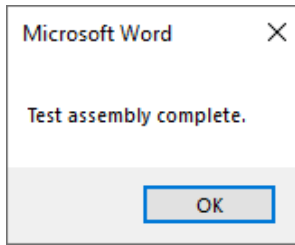


Figure 26, Test Assembly message

See Knowledge Base article [R10650](#), "Troubleshooting Word Document Assembly," for assistance with coding errors.

**Tip:** Always use the **Test Assemble** button after making changes to a document template prior to assembling a document.

## Help

This button (or Alt+P) displays PracticeMaster Word Document Assembly help.

## Step 1: Start a New Word Document Template

Before documents can be assembled, one or more document templates must first be created using Microsoft Word. All document templates that are created for Word Document Assembly must be based on the master document template provided with the software named CMTEMPLT. This file is installed when the **Document Assembly Toolbar** is installed.

Use the following steps to create a new template file (.DOT or .DOTM) based on CMTEMPLT.

### ► To create a new document template

1. Open PracticeMaster.
2. Open Word.
3. In Word, click **File | New** and then select **Personal** to access your available templates as shown in Figure 27.

**Note:** If the Personal section is not present, follow the instructions for configuring the personal templates location on page 10.

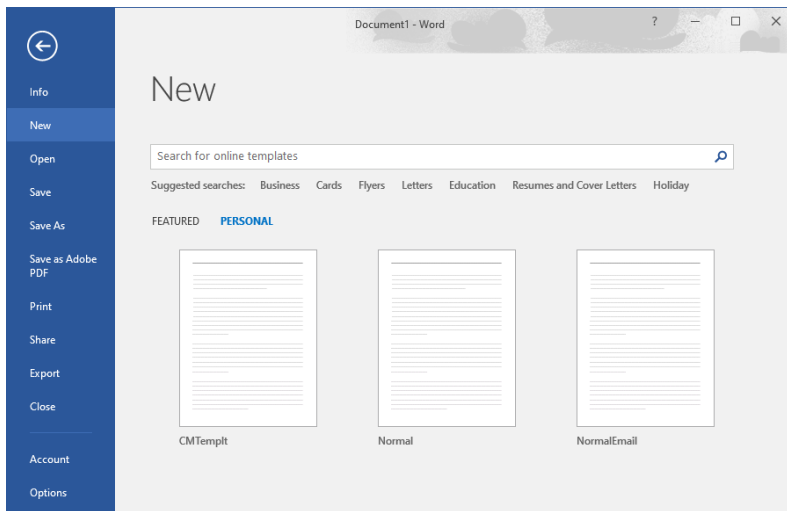


Figure 27, Personal Templates in Microsoft Word 2016

4. Select CMTEMPLT to start a new Word document template.
5. Select **File | Save As**, select a location and filename for the template, and change the **Save as type** to “Word Macro-Enabled Template (\*.dotm)” or “Word 98-2003 Template (\*.dot)” as shown in Figure 28.

**Note:** Document templates will default their location based on the **Template Path** of the Area of Practice for the selected client, as shown on page 11.

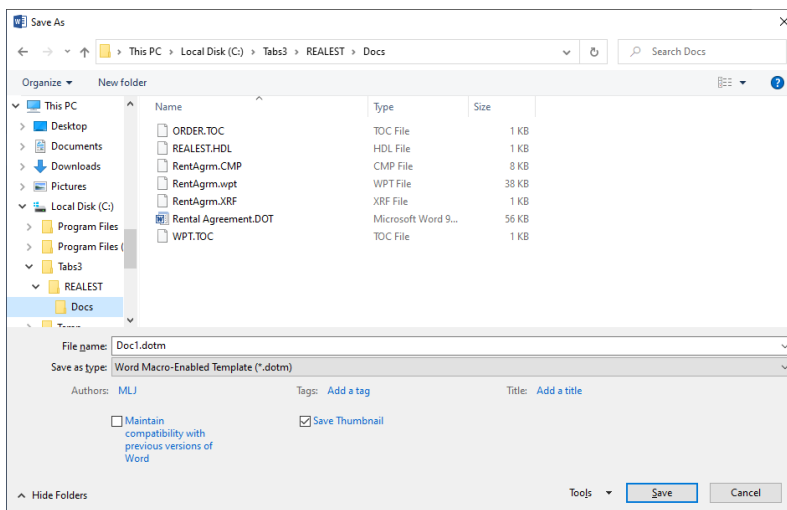
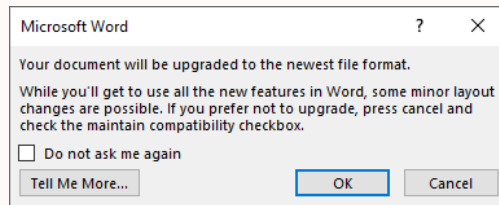


Figure 28, Save As in Microsoft Word 2016



---

**Note:** If the document template is created using Microsoft Word 2016 or later, you may encounter the following message:



If all workstations in the firm use the same version of Microsoft Word, the compatibility mode option does not need to be used. However, if another user plans to assemble the document with an earlier version of Word, click **Cancel**, select the **Maintain compatibility with previous versions of Word** check box, and then click **Save**.

Now that the template has been created, the PracticeMaster Document Assembly toolbar is available in Microsoft Word to populate the template.

## Step 2: Select a Base Letter

Here is a sample form letter that we will be using as the basis for our document template:

---

JENSEN, MARTIN & ANDERSON, P.C.  
ATTORNEYS AT LAW

—◇—  
1621 CUSHMAN DRIVE  
LINCOLN, NE 68512  
402-419-2200

August 1, 2022

Margie M. Sands  
Medical Arts Associates  
2390 Hathaway Blvd.  
Lincoln, NE 68513

**MEDICAL RECORDS REQUEST**

Re: Michael Larson  
W.C. #: WC-92-27084651  
Carrier #: KB-9076-37892  
Date of Injury: July 14, 2022

Dear Margie M. Sands,

Please provide documentation concerning all of the following:

1. Drugs administered or prescribed to the above named claimant.
2. Psychological or psychiatric conditions of above named claimant.
3. Copies of history, physical, discharge summaries, and operative reports.
4. Copy of outpatient and emergency room admissions.
5. Date treatment began and ended.
6. Type and description of treatment administered to above named claimant.
7. Copy of complete hospital chart and file, including but not limited to physician's, nurse's, surgeon's, or any and all medical personnel's notes.
8. Any and all information regarding medical bills, liens, or costs involved with the medical care of above named claimant.
9. Treating physician's name, address, and telephone number.
10. All admissions or care at this facility or by this doctor.

Please send this information to Michael L. Jensen; Jensen, Martin & Anderson, P.C.; 1621 Cushman Drive; Lincoln, NE 68512 within thirty (30) days from the date of this request. We will promptly remit payment upon receipt of records.

Thank you for your assistance with this matter.

Very truly yours,

Michael L. Jensen  
Jensen, Martin & Anderson, P.C.

copy: Michael Larson

---

## Step 3: Replacing Text with Variables

Let's look at different sections of the letter that we could potentially replace with information stored in PracticeMaster. The most obvious are the address and reference blocks, as shown below.

August 1, 2022


Margie M. Sands  
Medical Arts Associates  
2390 Hathaway Blvd.  
Lincoln, NE 68513

Re: Michael Larson  
W.C. #: WC-92-27084651  
Carrier #: KB-9076-37892  
Date of Injury: July 14, 2022

Dear Margie M. Sands,

We will leave the date for now, since it is easier to replace with an internal variable, as shown on page 51.

Let's start with replacing the "Re:" lines with the client's name we are assembling a document on behalf of.





1. Highlight "Michael Larson".
2. Click the  button. This is the **Variable** button, and is used to insert information stored in PracticeMaster as text in the assembled document.
3. Expand the *System Files*, *Client*, and *Client Name* groups.
4. Select the **Full Name** field.
5. Click **OK**.
6. Enter a **Name** for the field, such as "Full\_Name".
7. Click **OK**.
8. Repeat these steps for all other instances of the client's name.






This will replace our client's name with the following code:


```
«Var|0|0||0|0%H|Full_Name|CMSYSTEM|CMCLIENT:Name:Name»
```

Next, we will replace the address block with information from the Area of Practice files. To do this, we need to enable access to the files, and then replace the necessary variables.

**Note:** File access to System Files is automatically enabled. To enable access to any other file, use the **File Access** button. Each file that is enabled access must also have an end access specified.

- 
- Access the Medical Services file.
    1. Place the cursor at the top of the document.
    2. Click the  button. This is the **File Access** button, and is used to open or close an AOP, Lookup, or Common Client Related file in PracticeMaster.
    3. Make sure **Begin File Access** is selected.
    4. Expand the *Areas of Practice* and *Workers' Compensation* groups.
    5. Select **Medical Services**.
    6. Specify a **Record Access** of *Select*.
    7. Select the **Add Record if Empty** option.
    8. Click **OK**.
    9. Place the cursor at the end of the document.
    10. Click the  button.
    11. Select **End File Access**.
    12. Click **OK**.
  
  - Replace the name in the address block with the doctor information in the Medical Services record.
    1. Highlight "Margie M. Sands".
    2. Click the  button.
    3. Expand the *Workers' Compensation Files*, and *Medical Services* groups.
    4. Select the **Patient Seen By (Doctor)** field.
    5. Click **OK**.
    6. Enter a **Name** for the field, such as "Seen\_By".
    7. Select **Prompt for a value if the variable is empty**.
    8. Click **OK**.
    9. Repeat the steps above for the name in the salutation (e.g., "Dear Margie M. Sands,").
  
  - Replace the name of the medical office with the information in the Medical Services record.
    1. Highlight "Medical Arts Associates".
    2. Click the  button.
    3. Expand the *Workers' Compensation Files*, *Medical Services*, and *Name of Medical Provider* groups.
    4. Select the **Organization** field.
    5. Click **OK**.
    6. Enter a **Name** for the field, such as "Provider".
    7. Select **Prompt for a value if the variable is empty**.
    8. Click **OK**.
-




- 
- Replace the address of the medical office with the information in the Medical Services record.
    1. Highlight "2390 Hathaway Blvd."
    2. Click the  button.
    3. Expand the *Workers' Compensation Files, Medical Services, and Name of Medical Provider* groups.
    4. Select the **Address 1** field.
    5. Click **OK**.
    6. Enter a **Name** for the field, such as "Addr1".
    7. Click **OK**.
    8. Press *Enter*, and then click the  button.
    9. Expand the *Workers' Compensation Files, Medical Services, and Name of Medical Provider* groups.
    10. Select the **Address 2** field.
    11. Click **OK**.
    12. Enter a **Name** for the field, such as "Addr2".
    13. Select **Compress line if result is blank**.
    14. Click **OK**.
    15. Press *Enter*, and then click the  button.
    16. Expand the *Workers' Compensation Files, Medical Services, and Name of Medical Provider* groups.
    17. Select the **Address 3** field.
    18. Click **OK**.
    19. Enter a **Name** for the field, such as "Addr3".
    20. Select **Compress line if result is blank**.
    21. Click **OK**.
    22. Highlight "Lincoln".
    23. Click the  button.
    24. Expand the *Workers' Compensation Files, Medical Services, and Name of Medical Provider* groups.
    25. Select the **City** field.
    26. Click **OK**.
    27. Enter a **Name** for the field, such as "City".
    28. Click **OK**.
    29. Highlight "NE".
    30. Click the  button.
    31. Expand the *Workers' Compensation Files, Medical Services, and Name of Medical Provider* groups.
    32. Select the **State** field.
    33. Click **OK**.
    34. Enter a **Name** for the field, such as "State".
-





35. Click **OK**.
36. Highlight "68513".
37. Click the  button.
38. Expand the *Workers' Compensation Files, Medical Services, and Name of Medical Provider* groups.
39. Select the **Zip** field.
40. Click **OK**.
41. Enter a **Name** for the field, such as "Zip".
42. Click **OK**.


Our address block will appear as the following code:

```
«Var[0]1[0]||0%[H]Seen_By|WC|MEDICAL:Seen_By»
«Var[0]1[0]||0%[H]Provider|WC|MEDICAL:Name:Organization»
«Var[0]1[0]||0%[H]Addr1|WC|MEDICAL:Name:Addr1_Line1»
«Var[0]1[0]||1%[H]Addr2|WC|MEDICAL:Name:Addr1_Line2»
«Var[0]1[0]||1%[H]Addr3|WC|MEDICAL:Name:Addr1_Line3»
«Var[0]1[0]||0%[H]City|WC|MEDICAL:Name:Addr1_City», «Var[0]1[0]||0%[H]State|WC|MEDICAL:Name:Addr1_State» «Var[0]1[0]||0%[H]Zip|WC|MEDICAL:Name:Addr1_Zip»
```


Finally, we will replace the medical record information requested with information from the Area of Practice files. Again, we need to enable access to the files, and then replace the necessary variables.

- Access the Claimant Information File.
  1. Place the cursor at the top of the document.
  2. Click the  button.
  3. Make sure **Begin File Access** is selected.
  4. Expand the *Areas of Practice* and *Workers' Compensation* groups.
  5. Highlight **Claimant Information**.
  6. Specify a **Record Access** of *All*.
  7. Select the **Add Record if Empty** option.
  8. Click **OK**.
  9. Place the cursor at the end of the document.
  10. Click the  button.
  11. Select **End File Access**.
  12. Click **OK**.
  
- Replace the Workers' Compensation Number (W.C. #) line with the value stored in the Claimant Information file.
  1. Highlight "WC-92-27084651".
  2. Click the  button.

- 
3. Expand the *Workers' Compensation Files*, and *Claimant Information* groups.
  4. Select the **Workers' Comp. Number** field.
  5. Click **OK**.
  6. Enter a **Name** for the field, such as "Wc\_No".
  7. Select **Prompt for a value if the variable is empty**.
  8. Click **OK**.
- Access the Employer's Insurance file.
    1. Place the cursor at the top of the document.
    2. Click the  button.
    3. Make sure **Begin File Access** is selected.
    4. Expand the *Areas of Practice* and *Workers' Compensation* groups.
    5. Highlight **Employer's Insurance**.
    6. Specify a **Record Access** of *Select*.
    7. Select the **Add Record if Empty** option.
    8. Click **OK**.
    9. Place the cursor at the end of the document.
    10. Click the  button.
    11. Select **End File Access**.
    12. Click **OK**.
  - Replace the Policy Number (Carrier #) line with the value stored in the Employer's Insurance file.
    1. Highlight "KB-9076-37892".
    2. Click the  button.
    3. Expand the *Workers' Compensation Files*, and *Employer's Insurance* groups.
    4. Select the **Policy Number** field.
    5. Click **OK**.
    6. Enter a **Name** for the field, such as "Policy\_No".
    7. Select **Prompt for a value if the variable is empty**.
    8. Click **OK**.
  - Access the Accident Information file.
    1. Place the cursor at the top of the document.
    2. Click the  button.
    3. Make sure **Begin File Access** is selected.
    4. Expand the *Areas of Practice* and *Workers' Compensation* groups.
    5. Highlight **Accident Information**.
    6. Specify a **Record Access** of *First*.
    7. Select the **Add Record if Empty** option.
-

8. Click **OK**.
9. Place the cursor at the end of the document.
10. Click the  button.
11. Select **End File Access**.
12. Click **OK**.

- Replace the Date of Injury line with the value stored in the Accident Information file.

1. Highlight "July 14, 2022".
2. Click the  button.
3. Expand the *Workers' Compensation Files*, and *Accident Information* groups.
4. Select the **Date of Accident** field.
5. Click **OK**.
6. Enter a **Name** for the field, such as "Acdt\_Date".
7. Select **Prompt for a value if the variable is empty**.
8. Click **OK**.


The following code will be included at the top and bottom of our template:

```
«FileOpen|%H|BeginAccess|WC|MEDICAL|Select|Yes|»«FileOpen|%H|BeginAccess|WC|CLAIMANT|All|Yes|»
«FileOpen|%H|BeginAccess|WC|INSURE1|Select|Yes|»«FileOpen|%H|BeginAccess|WC|ACCIDENT|First|Yes|»
.....
«FileClose|%H|EndAccess» «FileClose|%H|EndAccess»«FileClose|%H|EndAccess»«FileClose|%H|EndAccess»
«Tickler|%H|Calendar|CS|30|Confirm all medical records have been received.»
```

And the medical records block will appear as the following code:

```
Re: «Var|0|1|0||0|%H|Full_Name|CMSYSTEM|CMCLIENT:Name:Name»
W.C. #: «Var|0|1|0||0|%H|Wc_No|WC|CLAIMANT:Wc_No»
Carrier #: «Var|0|1|0||0|%H|Policy_No|WC|INSURE1:Policy_No»
Date of Injury: «Var|0|2|0||0|%H|Acdt_Date|WC|ACCIDENT:Acdt_Date»
```

Now that the client information variables are complete, are we done? Well, what happens if we assemble this document and Mr. Jensen is not handling the case? Likely, we will want these documents to be delivered to the attorney assigned to the case instead. Let's take care of that next. Perform the following steps:

1. Highlight "Michael L. Jensen".
2. Click the  button.
3. Expand the *System Files*, *Client*, and *Primary Tkpr* groups.
4. Select **Name**.
5. Click **OK**.
6. Enter a **Name** for the field, such as "Prim\_Tkpr\_Name".
7. Select **Prompt for a value if the variable is empty**.







- 
8. Click **OK**.
  9. Repeat these steps for all other instances of the attorney's name.

Our attorney will appear as the following code:

```
«Var[0]1[0]||0%H|Prim_Tkpr_Name|CMSYSTEM|CMCLIENT:Prim_Tkpr:Name»
```

In many firms, this will be the extent of our changes. However, let's look further ahead. What if we add a partner to the firm? Do we want to have to go through all of the document templates we have, potentially hundreds, and update the text for all of them? What if the firm moves, or has a change to their phone number or zip code? PracticeMaster already has access to your firm information, through the fields specified in System Configuration, and we can pull those same values into our documents. Let's replace the firm information with lookup fields:

1. Highlight the firm name.
2. Click the  button.
3. Expand the *Firm Variables* group.
4. Select **Firm\_Name**.
5. Click **OK**.
6. Enter a **Name** for the field, such as "Firm\_Name".
7. Click **OK**.
8. Highlight the firm address.
9. Click the  button.
10. Expand the *Firm Variables* group.
11. Select **Firm\_Address**.
12. Click **OK**.
13. Enter a **Name** for the field, such as "Firm\_Address".
14. Click **OK**.
15. Highlight the city, state and zip code.
16. Click the  button.
17. Expand the *Firm Variables* group.
18. Select **Firm\_City\_State\_Zip**.
19. Click **OK**.
20. Enter a **Name** for the field, such as "Firm\_CSZ".
21. Click **OK**.
22. Highlight the firm phone number.
23. Click the  button.
24. Expand the *Firm Variables* group.
25. Select **Firm\_Telephone**.
26. Click **OK**.
27. Enter a **Name** for the field, such as "Firm\_Phone".
28. Click **OK**.

---



Our letterhead block will appear as the following code:

```
«VAR|1|1|1||0%H|FIRM_NAME|FIRM_NAME»  
ATTORNEYS AT LAW  
—◆—  
«VAR|2|1|1||0%H|FIRM_ADDRESS|FIRM_ADDRESS»  
«VAR|3|1|1||0%H|FIRM_CITY_STATE_ZIP|FIRM_CITY_STATE_ZIP»  
«VAR|4|1|1||0%H|FIRM_PHONE|FIRM_TELEPHONE»
```

## Step 4: Adding Logic, Internal Variables, and Ticklers

Using the logic functions, it is easy to create documents that add some text for one type of case, or different text for another type. This is done by comparing the value of a field with a specified value, and providing <IF> and <ELSE> statements.

For our example, let's add a line that says "SECOND REQUEST" if we are assembling the document more than 30 days after the initial case is opened:

1. Position the cursor below the regarding block, above the salutation.
2. Click the Insert  button. This is the **Logic** button, and is used to perform logical comparisons.
3. Make sure **If** is selected.
4. In the *Operators* section, double-click the Date Function of **datenum()**.
5. Position the cursor after the function in the *Expression* section.
6. In the *Operators* section, double-click the Logical/Arithmetic operator > (**greater than**).
7. In the *Variables* section, expand the *System Files* and *Client* groups.
8. Double-click the **Date Opened** field.
9. Modify the value by 30 days by entering " + 30", and surrounding the **Date Opened** variable and the +30 with parentheses.
10. Your expression should look like this (without the quotation marks): "datenum() > (CMSYSTEM|CMCLIENT:Date\_Open + 30)".
11. Click **OK**.
12. Enter the text you want to be displayed. We will enter a line break, then type the text "SECOND REQUEST", and center it on the page.
13. Click the  button.
14. Select **Endif**.
15. Click **OK**.


The following code will be shown in the template:

---

```
«Logic|%%H|If datenum() > (CMSYSTEM|CMCLIENT:Date_Open + 30)»
```

```
SECOND REQUEST«Logic|%%H|Endif»
```

As another example of a variable, let's replace the date with the current system date.


1. Highlight "August 1, 2022".
2. Click the  button.
3. Expand the *Internal Variables* group.
4. Select **Date Variable**.
5. Click **OK**.
6. Enter a **Name** for the field, such as "Todays\_Date".
7. Click the **Formula** button.
8. In the *Operators* section, double-click the Date Function of **datenum()**.
9. Click **OK**. The value "datenum()" will be shown in the **Initial Value** field.
10. Click **OK**.

This will replace our date with the following code:

```
«Var[32|2|1|datenum()]|0%|H|Todays_Date|Date Variable»
```

Finally, let's add a calendar task reminding us to follow up on the receipt of the medical records.

### To add a calendar tickler

1. Position the cursor at the end of the template, and click the  button. This button is the **Tickler** button, and is used to create a new record in PracticeMaster when the document is assembled.
2. Select **Calendar**.
3. Specify a **Calendar Code**. You can use the lookup button to find a code, if necessary. We will use CS, which is a "Confirmation of Service" calendar code.
4. In the **Days** field, specify how many days after the current system date to schedule the calendar. We will use 30.
5. In the **Description** field, enter any default text. We will enter "*Confirm all medical records have been received.*"
6. Click **OK**.

The following code will be included in our template:

```
«Tickler|%%H|Calendar|CS|30|Confirm all medical records have been received.»
```

**Tip:** If we were establishing a full library of document assembly templates, we would probably want to create a second template that includes only the address information, then replace that section with an Include call to that template. This allows us to easily reformat, add information, or remove extraneous lines, if the firm's address changes. It would also allow us to utilize a letterhead image that can be changed in a single location rather than in each template.

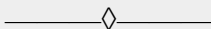
## Completed Document Assembly Template

This is how the completed template looks:

```
«FileOpen|0%|H|BeginAccess|WC|MEDICAL|Select|Yes|»«FileOpen|0%|H|BeginAccess|WC|CLAIMANT|All|Yes|»  
«FileOpen|0%|H|BeginAccess|WC|INSURE1|Select|Yes|»«FileOpen|0%|H|BeginAccess|WC|ACCIDENT|First|Yes|»
```

«VAR|1|1|1|0%|H|FIRM\_NAME|FIRM\_NAME»

ATTORNEYS AT LAW



«VAR|2|1|1|0%|H|FIRM\_ADDRESS|FIRM\_ADDRESS»

«VAR|3|1|1|0%|H|FIRM\_CITY\_STATE\_ZIP|FIRM\_CITY\_STATE\_ZIP»

«VAR|4|1|1|0%|H|FIRM\_PHONE|FIRM\_TELEPHONE»

«Var|3|2|1|datenum()|0%|H|Todays\_Date|Date Variable»

«Var|0|1|0|0%|H|Seen\_By|WC|MEDICAL:Seen\_By»

«Var|0|1|0|0%|H|Provider|WC|MEDICAL:Name:Organization»

«Var|0|1|0|0%|H|Addr1|WC|MEDICAL:Name:Addr1\_Line1»

«Var|0|1|0|1%|H|Addr2|WC|MEDICAL:Name:Addr1\_Line2»

«Var|0|1|0|1%|H|Addr3|WC|MEDICAL:Name:Addr1\_Line3»

«Var|0|1|0|0%|H|City|WC|MEDICAL:Name:Addr1\_City», «Var|0|1|0|0%|H|State|WC|MEDICAL:Name:Addr1\_State» «Var|0|1|0|0%|H|Zip|WC|MEDICAL:Name:Addr1\_Zip»

### MEDICAL RECORDS REQUEST

Re: «Var|0|1|0|0%|H|Full\_Name|CMSYSTEM|CMCLIENT:Name:Name»

W.C. #: «Var|0|1|0|0%|H|Wc\_No|WC|CLAIMANT:Wc\_No»

Carrier #: «Var|0|1|0|0%|H|Policy\_No|WC|INSURE1:Policy\_No»

Date of Injury: «Var|0|2|0|0%|H|Acct\_Date|WC|ACCIDENT:Acct\_Date»

«Logic|0%|H|If datenum() > (CMSYSTEM|CMCLIENT:Date\_Open + 30)»

SECOND REQUEST«Logic|0%|H|Endif»

Dear «Var|0|1|0|0%|H|Seen\_By|WC|MEDICAL:Seen\_By»,

Please provide documentation concerning all of the following:

1. Drugs administered or prescribed to the above named claimant.
2. Psychological or psychiatric conditions of above named claimant.

- 
3. Copies of history, physical, discharge summaries and operative reports.
  4. Copy of outpatient and emergency room admissions.
  5. Date treatment began and ended.
  6. Type and description of treatment administered to above named claimant.
  7. Copy of complete hospital chart and file, including but not limited to physician's, nurse's, surgeon's, or any and all medical personnel's notes.
  8. Any and all information regarding medical bills, liens or costs involved with the medical care of above named claimant.
  9. Treating physician's name, address and telephone number.
  10. All admissions or care at this facility or by this doctor.

Please send this information to «Var|0|1|0||0%H|Prim\_Tkpr\_Name|CMSYSTEM|CMCLIENT:Prim\_Tkpr:Name»; «Var|1|1|1||0%H|Firm\_Name|Firm\_Name»; «Var|2|1|1||0%H|Firm\_Address|Firm\_Address»; «Var|3|1|1||0%H| Firm\_City\_State\_Zip|Firm\_City\_State\_Zip» within thirty (30) days from the date of this request. We will promptly remit payment upon receipt of records.

Thank you for your assistance with this matter.

Very truly yours,

«Var|0|1|0||0%H|Prim\_Tkpr\_Name|CMSYSTEM|CMCLIENT:Prim\_Tkpr:Name»  
«Var|1|1|1||0%H|Firm\_Name|Firm\_Name»

copy: «Var|0|1|0||0%H|Full\_Name|CMSYSTEM|CMCLIENT:Name:Name»

«FileClose|%H|EndAccess» «FileClose|%H|EndAccess»«FileClose|%H|EndAccess»«FileClose|%H|EndAccess»  
«Tickler|%H|Calendar|CS|30|Confirm all medical records have been received.»

---

# References

## Knowledge Base

The following articles contain additional detailed information about the Word Document Assembly feature:

KB Article [R11447](#), "Working with PracticeMaster Document Assembly," includes much of the same information as this guide, as well as links to additional sources of information about Word Document Assembly.

KB Article [R10650](#), "Troubleshooting Word Document Assembly," contains information on how to troubleshoot errors encountered when creating or assembling Word Document Assembly templates.

KB Article [R11015](#), "How to Start a New Word Document Template," provides instructions for starting a new Word Document Assembly template.


KB Article [R10667](#), "Word Document Assembly Advanced Coding Tips," contains Word Document Assembly coding used to perform specific functions. This includes examples of coding beyond those included in this guide, such as creating conditional logic, adding records into a table, changing margins and preventing line breaks when combining templates, and more.

Our Knowledge Base is accessible in the software via the **Help | Internet Resources | Knowledge Base** menu and at:

[support.Tabs3.com](http://support.Tabs3.com)

## Help

The PracticeMaster Help includes comprehensive information about each window and process of the Word Document Assembly feature.

Help is context-sensitive; you can access help for the current window in PracticeMaster by pressing *F1* or clicking the  button.

Help is also available from the PracticeMaster Document Assembly Toolbar in Microsoft Word by pressing *Alt+P* or clicking the **Help** button.

---

## Training Videos

You can view the following training videos for more information. Clicking a link will open the associated training video in your browser.

 [Intro to Word Document Assembly](#)

 [Intro to Coding Document Assembly Templates](#)

All training videos are also accessible in the software via the **Help | Training Videos** menu and at:

[\*\*\*Tabs3.com/video\*\*\*](#)

---

# Index

<b>A</b>		<b>E</b>	
Adding		Error	
Internal Variables .....	50	definition .....	8
Logic .....	50		
Ticklers .....	50		
Assembling			
documents .....	13		
Assembly Error			
definition .....	8		
<b>C</b>		<b>F</b>	
Common Terminology .....	6		
Completed Document Assembly Template ..	52		
Creating a Document Template .....	17		
<b>D</b>		<b>I</b>	
Document		Include	
definition .....	6	definition .....	7
Document Assembly		Installing	
overview .....	8	Document Assembly Toolbar .....	9
process .....	13	Personal Templates location .....	10
setting up .....	9	Intro to Coding Document Assembly	
template .....	39	Templates	
use cases .....	5	training video .....	55
Document Assembly Toolbar		Intro to Word Document Assembly	
buttons .....	17	training video .....	55
installing .....	9		
personal templates location .....	10		
Document Management Records			
definition .....	6		
		<b>L</b>	
		Logic	
		definition .....	7
		<b>R</b>	
		Record	
		definition .....	6
		Replacing Text with Variables .....	43



---

## S

Setting Up	
document assembly .....	9
template locations .....	11

## T

Template	
creating .....	39
definition .....	6
locations .....	11
Terminology .....	6
Test	
definition .....	8
Tickler	
definition .....	7
Toolbar .....	17
Toolbar Buttons .....	17
Training Videos	
intro to coding document assembly	
templates .....	55
intro to Word document assembly .....	55

## V

Variable	
definition .....	6
Videos	
intro to coding document assembly	
templates .....	55
intro to Word document assembly .....	55

## W

Word Document	
definition .....	6
Word Template	
definition .....	6