INCREASED EFFICIENCY: FORMULARY DRUG CONVERSION

AUTOMATION USING VISUAL BASIC-BASED MACROS WITH

ATTACHMATE REFLECTIONS IN THE PHARMACY SETTING

Dr. Chad A. Naville, PharmD

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Master of Health Informatics
in the Department of Informatics
Indiana University

December 2013

Accepted by the Faculty of Indiana University, in partial fulfillment of the requirements for the degree of Master of Health Informatics.

_____

Dr. Josette Jones, Chair

Master's Thesis Committee

_____

Dr. Roland Gamache

_____

Dr. Mark Triboletti

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Dedicated to my mother and father, for their unconditional love and encouragement.

ABSTRACT

Dr. Chad Naville

INCREASED EFFICIENCY: FORMULARY DRUG CONVERSION AUTOMATION USING VISUAL BASIC-

BASED MACROS WITH ATTACHMATE REFLECTIONS IN THE PHARMACY SETTING

Health care automation provides opportunities for health care agencies to save time, save money, and increase patient safety. The Department of Veterans Affairs medical centers use a program, Attachmate Reflections, for pharmacy medication order verification. This program is a command line interface that allows the use of macros, or programmed automated routines, that have the ability to automate repetitive tasks. Through the use of macro programming at the VISN 11 VA medical centers, this author was able to automate converting patients from Combivent MDI inhalers to its successor Combivent Respimat inhalers due to the MDI inhaler being withdrawn from the market. Usage of the macro resulted in a time savings of 649.1 hours, cost savings of $32,748.36, and increased patient safety by providing consistent medication instructions, correct dispense quantities, correct prescription day supply, and correct number of refills remaining on the prescription.

INTRODUCTION TO SUBJECT

Automation and informatics are two of the biggest buzzwords in the healthcare industry today. The phrase "pharmacy automation" in Google yields many results for automated medicine cabinets, pill counters, and injectable compounders. One meaning of pharmacy automation is rarely used however: automation of pharmacy via coding. The intent of automation is to minimize errors, provide consistency, save time, and help make employees more productive. The aforementioned automated devices have proven to be valuable in all of these traits; however the use of pharmacy coding to automate repetitive tasks does not appear to have widespread usage.

Coding is synonymous with computer programming, which is a term used for "developing and implementing various sets of instructions to enable a computer to do a certain task" (Computer programming, n.d.). Sets of instructions are written using a programming language which communicates instructions in human form to a computer (Computer language, n.d.). There are several programming languages used today by programmers such as BASIC, FORTRAN, COBAL, JAVA, Javascript, .NET, Visual Basic, C, and C++.

A survey conducted by the American Society of Health-Systems Pharmacists found that informatics and technology are widely present in all steps of the medication-use process (Pedersen & Gumpper, 2008). Approximately 30.4% of hospitals had their pharmacy information technology (IT) team, in particular, work on informatics and technology implementations in their current medication-use processes. This number is estimated to continue increasing as facilities, such as the Department of Veterans Affairs (VA), employ full time pharmacy IT personnel to optimize medication systems (Pedersen & Gumpper, 2008).

As previously mentioned, automation systems are commonly referenced as automated cabinets, pill counters, etc. Automation for the scope of this paper however will refer to the use of coding to optimize medication conversions. For the scope of this project, a conversion is a term to describe the actions of changing a patient's prescription from one medication to a different, replacement medication. Resources gathered mention little of using programming or coding in the pharmacy to optimize medication systems, however as learned by this author, programming should be greatly explored as a means to help pharmacists work more efficiently by mitigating conversion costs, medication errors, and time spent on the conversion process. Veterans Affairs hospitals use software for medication order entry and verification that has the ability to run macros on top of it written in the programming language Visual Basic developed by Microsoft.

The purpose of this project was to use the ability to develop macros to automate the conversion of patients on the albuterol/ipratropium metered dose inhaler (MDI), Combivent MDI, to the newly released replacement albuterol/ipratropium inhalation spray, Combivent Respimat. The reason for this conversion is the Combivent MDI inhaler is being withdrawn from market over concerns of the use of chlorofluorocarbons (CFC's) in the MDI inhaler (Phase Out of Combivent Inhalation Aerosol - Questions and Answers, 2013). Therefore, the manufacturer has

discontinued production of the MDI inhaler as of December 31, 2013 resulting in the need to convert patients from the MDI inhaler to the Respimat inhaler (Boehringer Ingelheim, n.d.). The conversion process itself is not automatic and requires a pharmacist to manually change a patient from the MDI inhaler to the Respimat inhaler. The project is to evaluate the effectiveness of developing a macro to convert patients from the MDI inhaler to the Respimat inhaler and analyze its efficiency at saving time, saving money, and increasing patient safety. This project will also identify the strengths, weaknesses, complications, and user impressions of deploying a macro-based form of pharmacy automation.

BACKGROUND

The Richard L. Roudebush Veterans Affairs Medical Center in Indianapolis, Indiana uses a hospital information system termed VistA, or Veterans Health Information Systems and Technology Architecture, which has also been released for public use under the name Open VistA which is coordinated by MedSphere. VistA has the ability to interface with many different programs including Computerized Patient Record System (CPRS), Bar Code Medication Administration (BCMA), and Decentralized Hospital Computer Program (DHCP, also commonly referred to as VistA). The pharmacy department uses the terminal emulation program Reflections by Attachmate as the DHCP interface that provides a DOS-like command line interface to VistA used for all pharmacy purposes including but not limited to medication order entry, verification, checking, and reporting.

Reflections itself is a powerful terminal emulator that allows the user to create, record, and edit macros. According to Microsoft Bing, a macro is a computer instruction that initiates a series of additional instructions. Using Reflections, this allows the user to create a set of instructions that can quickly manipulate the users screen. A macro in Reflections is able to send keyboard commands, as if the user is typing very rapidly, to the screen and navigate quickly around VistA. Macros can also perform calculations based on any number of variables or input. As previously mentioned, Microsoft Visual Basic is the programming language used to write the macro. Therefore, in order to build a macro, the user must have a working knowledge of Microsoft Visual Basic and possess the skills to analyze, research, and synthesize the macro based on the steps required of a given task. When done properly, a user can write a macro to complete any number of tasks such as taking a user to a specific prompt, running a report, or in the instance of this project, performing automated conversions taking into account a wide variety of scenarios and performing calculations based on different patient variables.

The VA frequently changes formulary medications in an effort to save health costs. A current problem that the VA experiences with these formulary changes are that patients who are on the medications that are being changed require a pharmacist to manually go into the patients profile, select the medication, discontinue the old prescription, enter a new prescription for the new medication product, and ensure the new prescription is accurate and correct. The steps when manually converting a patient, without the aid of macro automation, from one pharmaceutical item to another consumes a significant amount of time and requires a high focus on the details such as old prescription dosing, new prescription dosing, patient's doctor,

patient's clinic, calculating correct day supply, calculating correct dispense quantities, calculating remaining refills, old dispense date, and calculating a new dispense date. There are at minimum 30 different individual keyboard actions/selections required to manually convert a patient from one mediation to another that require pharmacist thought. Unfortunately, there is significant room for error and when this procedure is repeatedly several thousand times, the risk of making a medication error increases significantly (Preventing medication errors, 2010). It has been measured by this author in previous formulary conversions that the average time required to complete one medication change on one patient takes approximately 4-6 minutes. It is realistic to estimate however that other pharmacists may be able to complete a conversion more quickly (less than 4 minutes) or require more time (greater than 6 minutes) depending on the pharmacists skill with converting medications, experience using Reflections, comprehension speed, and calculation speed.

Macros are a potential solution to the problem of expediting and automating formulary conversions. Since a macro can take into account a multitude of variables, depending on its programming, and is already incorporated into the VA system, implementing formulary conversion automation is a logical and efficient solution that has the ability to prevent medication errors, save time, and save money. It is estimated that medication errors make up 20% of all medical errors (Ibrahim, Bahgat-Ibrahim, & Reeves, 2010). A macro that has the ability to automate calculations and generate consistent, accurate, and correct prescriptions serve as a way minimize errors while controlling healthcare costs. This project will review and analyze the gained efficiency of using a macro to automate a medication conversion and validate the usage of macros in future high volume formulary conversions.

Formulary conversions happen typically several times per year as drug contracts or drug costs change. The change and associated impact on patient care is reviewed by the Pharmacy and Therapeutics Committee and if approved, criteria is set for the conversion, equivalent doses verified, and reports are generated listing all of the patients on the medication that needs to be converted to the new formulary equivalent. A pharmacist in the outpatient pharmacy then must use the list to enter the patients profile, select the medication and make note of the following existing prescription parameters:

- dosing frequency/schedule
- original order date
- refill fill dates (if applicable)
- remaining refills
- future fill date (if applicable)

- day supply
- quantity
- ordering provider
- provider clinic

Each of these parameters are used to determine the dosing of the new equivalent medication, number of refills to apply to new prescription, when to send the medication to the patient, what the day supply of new medication should be, as well as how many unit doses to send to the patient. The provider information is retained on the prescription as required by federal law as well as the prescribing clinic information. As stated, it would typically take an outpatient pharmacist approximately 4 to 6 minutes to perform one medication conversion. Depending on the size of the facility, there may be several thousand patients that require conversion resulting in a significant investment of time and money.

METHOD

Requirements

The requirement and needs of the macro had to be identified and specified. The macro would be required to:

- open a patients electronic medication profile
- identify and select the correct medication to convert to the new formulary item
  - Combivent MDI to Combivent Respimat
- open the existing medication record and read/record parameters from the existing prescription including:
  - original order date
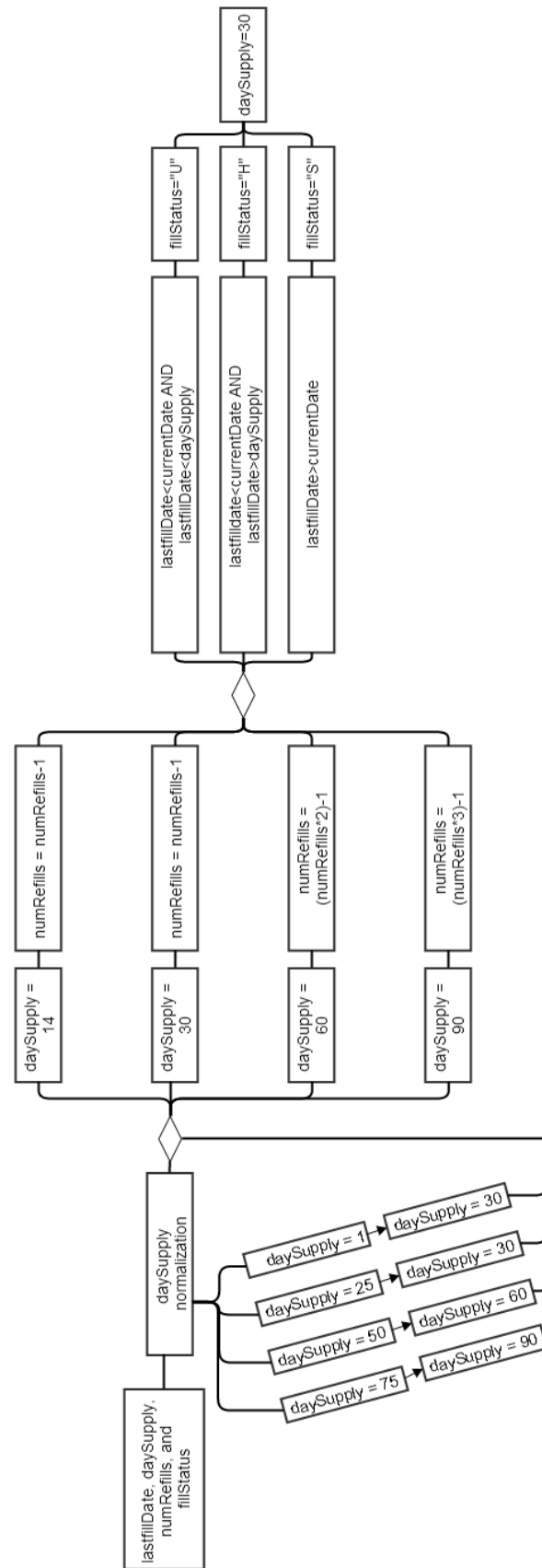 - refill date (if applicable)
  - future fill date (if

Figure 1: Calculation Workflow

applicable)
- o remaining refills
- initiate the conversion process if the prescription has remaining refills or has a future fill date
- adjust patient medication instructions according to pre-specified and approved directions for the new prescription
- adjust original fill date of new prescription to match original order date of previous prescription for prescription expiration purposes
  - o non-controlled substance prescriptions are valid for one year from issue date, Ind. Code § 25-26-13-25 (2013).
- adjust day supply to pre-determined number
- adjust quantity of unit doses dispensed
- adjust refills to match refills remaining from original prescription
- adjust fill date to match future fill date (if applicable)
- allow a pharmacist to determine if the prescription will be covered by VA or will require a copay (service connection)
- allow pharmacist to intervene if patient has documented allergy
- allow pharmacist to prevent duplicate entry and discontinue duplicate entries (if applicable)
- verify prescription based on fill dates with three potential outcomes:
  - o suspend for mail in future
  - o hold if patient has not had filled in a while
  - o leave in active status without fill if no future fill date has been set but patient has filled recently
- select next patient and repeat above actions

In addition to converting the medication, the macro also would be required to not bypass any medication safety checks. The VA pharmacy system uses a drug safety check from First Databank and the macro should not bypass an event requiring the judgment of a pharmacist when a potential interaction, duplication, or other form of poly-pharmacy or alert exists. Therefore, the macro will not include any ability to override safety checks and will instead pause and wait for the pharmacist to provide a response to the alert as per normal prescription verification routine.

The macro would also require significant error adjustments. There is the slight, but possible, circumstance where a patient may already be converted to the Combivent Respimat. The macro would need to be able to alert the user that there is no medication to convert and halt or move to the next patient in the patient work list. Figure 1 also depicts another error control for day supply exceptions. In the VA healthcare system, patients are billed on 30, 60, or 90 day supplies. Any other day supply quantities that are billed are rounded up to the nearest 30, 60, or 90. However, in an effort to standardize the day supply, the macro would need to be able to take into account other day supply quantities. Figure 1 shows a diagram of how the macro will read the day supply and make an adjustment to the new prescription day supply to accommodate both current billing practices as well as day supply standardization. The

quantities 1, 25, 50, and 75 were identified from the patient work list as other possible day supply quantities that were used instead of the standardized 30, 60, or 90.

Calculations

Taking into consideration these variables, the macro will need to perform calculations on dates, refills, and verify status to finish the prescription. Figure 1 is the decision tree that is used for calculating the different variables: daySupply, numRefills, fillStatus, and lastFillDate.

- daySupply: the day supply of the original prescription that is converted to the approved static value of 30.
- numRefills: the number of refills from the original prescription that is then re-calculated when changing from a day supply other than 30 to the adjusted value for 30 day supplies.
- fillStatus: variable that holds the decision the macro makes based on the patients refill history of the original prescription. Outcomes possible:
  - U: convert prescription and leave as active
  - H: place prescription on hold
  - S: suspend/queue prescription to be filled and mailed on future date
- lastFillDate: contains the date of the previous or queued refill date. If the prescription has never been filled, then this will be a null value.

A requirement of the macro is that it must not directly access the facility servers. The variables the macro obtains are gathered from a method dubbed "screen scraping." Using screen scraping, the macro is programmed with specific static coordinates of the information being sought on the Reflections display, i.e. daySupply, numRefills, fillStatus, and lastFillDate. The coordinates follow the row and column axis. Visual Basic screen scrapes the information via the command GetText. Using this screen scraping process, no information is directly received nor transmitted to the facility servers therefore does not require any special permissions or special user access other than requiring a user with system designation of "pharmacist" and the respective electronic keys that a "pharmacist" user receives for prescription verification.

Prior to the macro performing any changes to the prescription, it first confirms if the prescription is able to be converted by ensuring it has either more than zero refills or a future fill date. The macro captures the lastFillDate and does a calculation based on the date of the conversion (Visual Basic code: "Now") and if the prescription has a date greater than "now," i.e. in the future, and it has zero refills, then it is allowed to continue with the conversion. Additionally, if the previous prescription has greater than zero refills remaining, then it is allowed to continue with the conversion. However, if the prescription has zero refills and no future fill date then this means that the maximum number of refills has been used and the existing prescription is simply waiting to expire, be discontinued by the provider, or be renewed by the provider. Facility policy prevents pharmacists from renewing prescriptions without confirmation by the patient's provider. Therefore, if there are zero refills and no future fill

date, then the macro alerts the user that the prescription has no remaining refills and turns itself off.

The macro is capable of converting prescriptions that have a 1, 25, 30, 50, 60, 75, or 90 day supply to the pre-approved new value of all prescriptions being a 30 day supply. The number of refills on the original prescription is adjusted based on the day supply to compensate for the change to a 30 day supply. Ex: A prescription with a 90 day supply and 3 refills will be converted to a 30 day supply with 10 refills. When the macro does a conversion, the algorithm is designed to include the new prescription as one of the refills; therefore all fills are the adjusted value of refills minus one. The refills are deducted one refill because after the conversion, the prescriptions are able to be queued to mail without having a refill deducted which, in essence, is giving the patient an additional refill. Prescriptions that are queued for mail do not have a refill deducted since this medication has already been deducted a refill. If a day supply is anything other than the aforementioned numbers, then the prescription may have special characteristics that the macro is unable to account for and informs the user that the specific prescription requires manual conversion and turns itself off.

The fillStatus is determined based on the patient's refill history. If a patient has had a Combivent MDI fill within the time span of the day supply but does not have a future refill queued for mail, then the macro converts the prescription and leaves it as active so the patient may refill the prescription without pharmacy intervention. Ex: A patient with an original prescription with a 30 day supply and the lastFillDate was 15 days before the date the original prescription is converted; it is assumed the patient is actively using the prescription but has not called the pharmacy or used MyHealtheVet, the Veterans online health record that can refills prescriptions, to request another refill. Since the patient is actively using the medication, the converted prescription is left as active allowing the patient to request a refill at their convenience, which will result in the fillStatus being designated as "U." If a patient has an original prescription with a 30 day supply but the lastFillDate is 45 days previous of the date of conversion, then the patient is not using the prescription regularly and results in the converted prescription being placed on hold and fillStatus receives designation of "H." When a prescription is placed on hold, the patient needs to talk to an outpatient pharmacist to release the prescription from hold in the event a patient wants the prescription to be refilled. If the patient has an original prescription with the lastFillDate in the future from the date the prescription is converted, then this means the patient has previously requested a refill and it will be mailed at a later date. The converted prescription then is also queued for mail for the same date as the original prescription and fillStatus receives the designation of "S."

Stepwise Process

The macro was designed to require two components, a Microsoft Excel spreadsheet consisting of a list of prescription numbers and any other information that a user may find useful (i.e. patient name, social security number, prescription original date, refill date, quantity, day supply, etc) as well as Reflections, which has a direct connection to the patients electronic medication record. The macro is programmed to use the list of prescription numbers from the

Excel spreadsheet located in column A, starting in row one. The macro copies the prescription number to the computers clipboard memory and pastes the prescription number with a pre-appended hashtag (#) to the Reflections patient selection screen. This opens the correct patient profile as it is based off of the prescription number and not the patient's name which eliminates the possibility of selecting an incorrect patient with a similar or same name. The macro opens the patients profile and then searches for the specific phrase "ALBUTEROL 90/IPRATROP 18MCG" as this is the precise name of the inhaler within the pharmacy system. If the macro does not find the phrase, it prompts the user and since there is nothing to convert, turns itself off. Otherwise, the macro captures the row of the inhaler and captures the refills and day supply via screen scraping. Next, it opens the prescription and captures the issue date and last fill date and performs a calculation based on the last fill date and refills to determine if the prescription is appropriate for conversion.

The macro then does the calculation to convert the prescription from a 25, 50, 60, or 90 day supply to a 30 day supply. If the prescription originally had a 14 day supply, then this is left intact as these day supplies are used in special circumstances and not to be changed in accordance with VA policy. The respective quantities and refills are also converted and fillStatus designation assigned per the algorithm shown in Figure 1.

After performing the necessary calculations and conversions, the macro then proceeds to convert the old prescription to a new prescription via editing. When a prescription is edited to a new drug product, it is automatically assigned a new prescription number and all the previous values ordered on the original issue date are reinstated. This means that if a prescription only has one refill left, but originally had three refills, the new prescription created from editing also has three refills, which will be later adjusted. The macro proceeds with changing the drug item from the Combivent MDI inhaler to Combivent Respimat and then corrects the directions to match a predetermined set of instructions: "inhale one puff orally four times a day as needed."

Once the directions are adjusted based on the new inhaler, Reflections then presents the patient's complete prescription editing screen as seen in Figure 2. The macro then makes the adjustments to the patient's day supply, refills, quantity, issue date, fill date, patient comments, and remarks. The day supply was changed to be either fourteen or thirty, depending on the original prescription. The quantity has a static number assigned of one inhaler per fourteen or thirty day supply. The refills were calculated as previously described. The issue date and fill date retained the same information that was screen scraped from the original prescription. The patient comments were edited to include "for shortness of breath" for an indication as mandated by the Joint Commission on Accreditation of Healthcare Organizations MM.04.01.01, MM 05.01.01 EP 8, and NPSG 03.06.01 (Rich, 2011).

The macro was developed by this author to flow in a series of groups. The various groups are denoted by the use of asterisks with a brief phrase of the central idea of the respective group, ex:

'********* Opens excel spreadsheet *********

8

The macro was coded using rapid prototyping techniques where needs are consistently assessed and the software improved as more needs were identified (Gordon & Bieman, 1997). The macro's groups were tested for accuracy as the group was completed to verify information exiting the group was consistent based on the information going into the group. An example is the check that is done to ensure a prescription has refills remaining on it. If a prescription does not have a future fill date and also does not have any remaining refills, then that prescription is not eligible for conversion because it would inappropriately add another fill to a prescription that has been filled by the patient the maximum number of times permitted by their provider. The group "check to make sure refills are not zero" would be fed a dummy number just before it by declaring a value for the specified variable. In the case of the refills group, if numRefills was set to 1, then the variable should pass through the group without triggering a prompt from the macro. However, if numRefills is set to 0, then the group would look also at lastFillDate to ensure that it is set in the future. If it is not set in the future and there are no refills and the macro would prompt the user there are no refills remaining and [the macro] would turn itself off to prevent inappropriate conversion. In order to fully test this section, the variables numRefills and lastFillDate would have to have values declared just prior to the group:

```
numRefills = 0                    'testing variable
lastFillDate = 1/1/2014           'testing variable

'********* Check to make sure refills are not zero *********
If numRefills < 0 Then
    MsgBox ("Something is wrong... no value could be calculated from refills,_
      exiting macro.")
    Exit Sub
ElseIf numRefills = 0 And lastFillDate < Now Then
    MsgBox ("There are no refills remaining on this patient's Combivent rx,_
      exiting macro.")
    Exit Sub
End If
```

The macro would then assume that these variables that are declared in the testing process are the same variables that are in the prescription and the decision process by the macro could be tested. Again, if there were no refills, then the macro would also consider the fill date. If the fill date is not in the future, then the macro would prompt the user that "there are no refills remaining on this patient's Combivent prescription, exiting macro" and also the macro would turn itself off. Conversely, if the number of refills was one or more, then nothing would happen and the macro would proceed with the conversion. In order to consider all cases possible, error control was built into this section. If the macro reads there are less than zero refills, then this is indicative of an error and the macro should not proceed without user intervention and error checking. Therefore, in the event the number of refills are less than zero, which should be impossible under normal circumstances, the macro will prompt the user that "something is wrong… no value could be calculated from refills, exiting macro" and the macro will turn itself off.

All groups were tested individually in this fashion of setting the variables used with a specific value, passing the value through the group, and observing the results the macro provides. This ensured each section was working appropriately and ensured rapid development and quality assurance that the information was being handled and adjusted correctly.

After completion of all edits and adjustment, the macro then continues finalizing the prescription and presents the finished product to the pharmacist for final verification as is normal process for prescription verification sans macro. If the prescription is complete and accurate, the pharmacist then enters "yes" and the macro verifies the prescription under the name of the pharmacist user. After verification of the prescription, the macro will conclude with one of three different outcomes: active, hold, or suspend. Depending on the value of fillStatus (U, H, or S), the macro will take the respective action as described in the fillStatus calculation.

All of the aforementioned codes and calculations are contained within a large loop. Once the macro finishes the prescription, the macro then loops back to the beginning and advances down the Excel spreadsheet by one cell to the next prescription number. All variables at the beginning of the macro are set with static quantities of zero and all strings of "" [null] in order to prevent any information carryover from the previous prescription. As the macro progresses down the Excel list of prescription numbers, it highlights each prescription that is being converted so the user knows which prescriptions have been completed.

John Smith DOB: 1/1/1980 Gender: Male	Ht(cm): 170 Wt(kg): 75kg PID: 0000

```
                    Rx #:X99999999X
   (1)   Orderable Item:ALBUTEROL/IPRATROPIUM (INHALER) RESPIMAT INHL,ORAL
             CMOP Drug:ALBUTEROL 100/IPRATRO 20MCG 120D PO INHL
                 (2) Verb:INHALE
              (3) Dosage:1 PUFF
              (4) Route:ORAL
              (5) Schedule:QID PRN
     (6) Pat Instructions:FOR SHORTNESS OF BREATH
                       SIG:INHALE 1 PUFF ORALLY 4 TIMES A DAY AS NEEDED FOR SHORTNESS OF BREATH
       (7) Patient Status:SERVICE CONNECTION
           (8) Issue Date:1/9/13               (9) Fill Date: 5/1/13
           Last Fill Date:N/A
               Expires:1/9/14
           (9) Day Supply:30                   (10) QTY (EA): 1
           (11) # of Refills:7
             (12) Provider:Patient's Provider
              (13) Routing:Mail
                (14) Clinic:Location of Patients Clinic
             (15) Remarks:FORMULARY CHANGE
```

Figure 2: Reflections Prescription Edit Screen

RESULTS

The use of the macro is intended to save time, save money, and increase patient safety. In order to quantify and validate these intentions, several measurements were looked at both from an intuitive and calculated point of view. When a pharmacist converts a prescription manually, it takes an average of 4-6 minutes per patient, per prescription. Using the macro, the average is approximately 12.2 seconds (Range: 6-20, Median: 12) from a sampling of 100 conversions. While 12.2 seconds as the average is very quick, a more conservative estimate of 15 is more realistic given the number of pauses the macro makes and requests user input. The macro was used in VISN 11 VA's located in Indiana, Illinois, and Michigan to convert a total of 10,386 prescriptions. If the macro had not been used, it would take an estimated 692.4-1,038.6 hours to fully complete the conversion. Using the macro resulted in the conversion only taking 43.3 hours, a difference of 649.1 hours at minimum for the region. At one VA, there were approximately 2,222 prescriptions to convert and it was able to be completed in one weekend by one person over 9.3 theoretically uninterrupted hours (the macro was used by the staff pharmacist in addition to their normal duties). It was anticipated the conversion would have taken approximately 148.1 hours at minimum for these 2,222 prescriptions prior to the development of the macro.

Location	Quantity	Hrs (NM)*	Hrs (M)	Hrs Diff	Cost (NM)**	Cost (M)**	Cost Diff**
Ann Arbor, MI	664	44.3	2.8	41.5	$2,224.40	$139.03	$2,085.38
Battle Creek, MI	1,731	115.4	7.2	108.2	$5,798.85	$362.43	$5,436.42
Danville, IL	1,475	98.3	6.1	92.2	$4,941.25	$308.83	$4,632.42
Detroit, MI	1,448	96.5	6.0	90.5	$4,850.80	$303.18	$4,547.63
Indianapolis, IN	2,222	148.1	9.3	138.9	$7,443.70	$465.23	$6,978.47
Marion, IN	1,820	121.3	7.6	113.8	$6,097.00	$381.06	$5,715.94
Saginaw, MI	1,087	72.5	4.5	67.9	$3,641.45	$227.59	$3,413.86
Total:	**10,386**	**692.4**	**43.3**	**649.1**	**$34,931.58**	**$2,183.22**	**$32,748.36**

(NM)=no macro; (M)= with macro
*Calculated using 4 minutes/conversion; **Salary=$50.45/hr as averaged from salary survey from Salary.com

Table 1: Time and Cost Savings Breakdown by Location

In addition to work hour's savings, the macro presented an opportunity to save healthcare dollars as well. The situation of using the Combivent macro, developed by this author, is unique in that it posed no savings by conversion from a more expensive item to another cheaper item secondary to formulary or contractual reasons. Instead the conversion happened out of necessity due to a ruling by the FDA under the "Montreal Protocol on Substances that Deplete the Ozone Layer" where products containing chlorofluorocarbons are to be removed from the market (Title 42-The Public Health and Welfare, 2008). The result being there is no defined drug cost savings since the previous product is being discontinued and the conversion is occurring out of necessity. Table 1 shows the cost breakdown by facility within the region. Using data obtained of pharmacist salaries in the regions listed on Table 1 reflect an average pharmacist salary of $50.45 per hour. Applying the average value of $50.45 per hour for each respective hour in Table 1, the results show cost savings of using the macro when compared to not using the macro and completing the conversions manually. The total difference between

using the macro for conversion compared to manually converting patients is an estimated $32,748.36. This value is generated to provide a minimal estimation of savings since the calculations were based on an average salary as well as a shorter interval when completing the conversion manually of 4 minutes.

Actual results and savings are likely to be higher since the estimated range of manually converting a patient is 4-6 minutes without the pharmacist being distracted (Silver, 2010). In the case of the conversion taking place at the Indianapolis, Indiana VAMC, the pharmacists processed the patient conversions in addition to their normal patient care duties. Since the macro runs semi-autonomously, pharmacist involvement in conversions is minimized and only requires user intervention when drug utilization review alerts (DUR), normal "press enter" pauses, or any of the aforementioned pauses occur. The longest running session where the macro

Location	Annual Salary*
Ann Arbor, MI	$112,047.00
Battle Creek, MI	$100,287.00
Chicago, IL	$112,154.00
Detroit, MI	$107,664.00
Fort Wayne, IN	$101,570.00
Indianapolis, IN	$100,454.00
Muncie, IN	$100,393.00
Average:	**$104,938.43**
Hourly:	**$50.45**

*Data Accurate as of June 2013 from Salary.com

Table 2: Regional Salary Survey

did not force exit or find errors that could not be overcome was a total of 97 patient conversions. This means the macro was able to successfully, without being stopped and restarted, complete 97 conversions where the pharmacist only had to press enter, fix the patient instructions, approve DUR overrides, or clarify service connections for patients. It is unknown if there was a session that was able to go longer without requiring the user to stop and restart the macro to overcome an event that could not be taken into consideration by the macro, i.e. the macro turns itself off due to unexpected information.

Given that the macro contains hard-coded directions, patient safety is increased as long as the macro is able to compensate sufficiently for a variety of situations or if given an unknown event, the macro turns itself off autonomously. The macro was designed for mass usage such that if there is a situation that the macro cannot account for or is unable to compensate, it alerts the user what the exception is (i.e. abnormal day supply or quantity) and exits to prevent inappropriate filling of the prescription. The macro is based on the terminal emulator screen syntax and therefore if there are words, numbers, or directions that are unable to be understood, the macro halts and alerts the user. Since the macro also does not have direct database access, it does not bypass any patient safety measures that a pharmacist conducting a manual conversion would encounter. The purpose of the macro is to automate actions that the user would have to perform repeatedly, making the overall conversion experience more efficient. The actions that are performed on each conversion are repeated and therefore all prescriptions end with the same day supply, same quantity, same directions, same previous but compensated for refills, same original ordering provider, and same ordering provider clinic. Taking into consideration the possibility of alert fatigue a pharmacist may experience while converting the prescription manually, the overall risk of a conversion error is lower than that of a user not utilizing the macro (Baker, 2009; Heed this warning! Don't miss important computer alerts [Heed], 2007).

CONCLUSION

The usage of a macro when converting patients from one medication to another using proper coding and testing techniques presents an opportunity for a healthcare system to save time, save money, and increase patient safety. While the time commitment required developing and fully testing an automation macro is relatively high, in the case of this Combivent macro, the time commitment was only that of one pharmacist that resulted in a savings of approximately $32,618.53 and reduced conversion completion time from several weeks by several pharmacists to one weekend by one pharmacist at the Indianapolis VAMC. Of note, the macro was able to be written and tested during normal work hours, thus requiring no extra salary costs or overtime for development or deployment.

A lessons learned in the development of macros is that most pharmacists do not like seeing text rapidly scrolling on the users terminal. In previous macro user trials, the rapidly scrolling text resulted in higher levels of stress, confusion, and decreased the ability of the pharmacist to understand what the macro is doing. Therefore, in this Combivent macro, specific code statements were written to "hide" portions of the processing and changes being made to the prescription because it did not pertain to the user. An example was when the macro was changing the refills from 90 day supply to a 30 day supply. The user did not need to see when the macro was making this change because it was accepted that it was being done. If the macro failed to make the change, then there would be an error and the macro would either lose track of its progress and stall or alert the user that the syntax is unrecognized and exit, thus requiring the user to identify the problem and either restart the macro or complete that specific conversion manually. Once most of the changes were modified to be background events, users appeared less stressed and more open to using the macro because it only showed them the pieces that pertained to them, such as DUR overrides, service connection changes, or when they needed to "press enter to continue."

When the macro would perform calculations and manipulations on the prescriptions, it was learned that the user could still interact with the prescription and accidentally input an incorrect variable if they pushed a button on the keyboard. Since the macro is heavily syntax dependent, this would often result in the macro stalling because of an unrecognized occurrence appearing on the screen resulting in the macro needing to be restarted. In order to prevent inadvertent button pushing, the macro was designed so that when it is manipulating the prescription autonomously or performing calculations, the keyboard is locked out from the terminal emulator, thus solving the problem of random key presses by the user.

In addition to preventing usage anxiety by hiding extraneous operations, the macro needed to have special code added to prevent alert fatigue (Baker, 2009; Heed, 2007). The macro was made to loop twenty times (complete twenty conversions) and then automatically shut off. During previous macro experiences, the user often would be limited to just pushing the "enter" key repeatedly. Eventually, this may result in the pharmacist overriding an alert that should be more carefully considered or more fully researched. Choosing the number of twenty was arbitrary and chosen based on the personal experiences of this author and when the mind

would start to wander. The result of the author's alert fatigue was then extrapolated to all users and the number twenty was chosen so that the user would have to do something different, i.e. restart the macro, after twenty conversions instead of pushing the "enter" key indefinitely until the entire patient list has been converted. The alert fatigue limitation was temporarily lifted to 100 conversions to test the maximum number of conversions that could take place without requiring the macro be stopped and restarted by this author as mentioned previously.

When the macro was deployed, unusual compatibility circumstances were encountered that still remain a question. A compatibility issue was learned that all computers in the VA network were being upgraded from Microsoft Windows XP to Microsoft Windows 7. In a few instances, the facilities would report back that the macro was not compatible on a Windows 7 machine but would work on a Windows XP machine. The macro was originally written in a Windows XP environment, however when this bug was tested, it was unable to be recreated.

Another compatibility issue was the different syntax between VAMC's. The drug spelling at one VA facility may not be the exact same as the other. In order to prevent the macro from incorrectly choosing a medication, the prescription name was hard-coded into the macro which resulted in a few of the facilities requiring their own specially adjusted macro to account for their different drug name syntax. The same issue presented itself for the route and schedule fields. The route at one facility may read "1 inhalation" while another facility may read "1 orally," resulting in the macro stalling because it was designed only to discern one of these and not the other. The schedule field for the majority of the facilities contained the syntax "QID prn" while at other facilities required to be spelled out "four times daily as needed," again resulting in the macro stalling and the facility requiring their own adjusted macro.

A major weakness of the macro was its ability to continue to the next patient if no conversion was needed or if the previous prescription had been placed on hold. The macro was designed only to convert patients with an *active* order for Combivent MDI. If there was no prescription for Combivent MDI, then the macro would alert the user that no prescription was found and then shut itself off, thus requiring the user to edit the Excel spreadsheet, delete the top row so that the macro was able to be resumed at cell A1 and subsequently be restarted. If a prescription was already on *hold* and not *active*, then the macro could not perform a conversion as it was not programmed to take the previous prescription off of hold. At the time of build and deployment, no message was built to inform the user of this bug and thus required an education point during instruction of how to diagnose why the macro had stalled.

A final weakness of the macro is that copying and pasting a prescription number from the Excel spreadsheet to Reflections, the clipboard memory of the computer was constantly preoccupied. If the user was working on other activities as the same time as running the macro, then then user could not copy and paste anything from other programs because if they were to copy and paste information at an inopportune time, the macro would paste that information into Reflections, not pick the correct patient, and cause the macro to stall. Example: if a user is typing a document at the same time as running the macro and they copy a word from their

document with the original intent to paste that word at another place in the document, if it were copied at the wrong moment then the macro would paste that word into Reflections with the intention to call the next patient, but instead create an error and stall the macro. Therefore it was highly advised not to copy and paste information from other programs or activities while the macro was running.

Overall, the macro was widely accepted and liked by users. Specific feedback emails have stated:

> "Combivent conversion is completed - the macro was wonderful, this was the fastest conversion ever!"
> - Detroit VAMC

> "I spoke with the pharmacy chief at Danville yesterday who was just raving about the macro and said we should use it for all large-scale conversions. He was all about it."
> - Indianapolis VAMC re: Danville VAMC

While the actual total duration of conversion time for each facility is unknown due to different people at each facility using it at different times on any given day, it was highly praised and there was immediate discussion regarding the next medication conversion macro. Taking into account the weaknesses and lessons learned, future macros may be developed and adjusted accordingly for even greater adoption and user friendliness.

Health informatics is "the application of information management and technology to the planning and delivery of high-quality and cost-effective healthcare" (Norris, 2002). This encompasses all the components of health information management such as the electronic medical record, e-prescribing, human-computer interaction, and standardized terminologies to accomplish the goal of ensuring cost effective care and maximizing patient safety. Using coding and pharmacy informatics, a new medication conversion technique has emerged and proven itself to be effective at rapidly changing patients from one prescription to another resulting in greater cost savings, time savings, and increased patient safety.

REFERENCES

Baker, D. E. (2009, June). Medication Alert Fatigue: The Potential for Compromised Patient Safety. *Hospital Pharmacy, 44*(6), 460-2.

Boehringer Ingelheim. (n.d.). *Combivent Respimat Inhalation Spray*. Retrieved June 10, 2013, from Combivent Respimat: http://www.combivent.com/respimat/index.html

Brock, T. P., & Smith, S. R. (2007, June 15). Instructional design and assessment. *American Journal of Pharmaceutical Education, 7*(3). Retrieved June 10, 2013

Computer language. (n.d.). Retrieved June 10, 2013, from Dictionary.com: http://dictionary.reference.com/browse/COMPUTER+LANGUAGE?s=t

Computer programming. (n.d.). Retrieved June 10, 2013, from BusinessDictionary.com: http://www.businessdictionary.com/definition/computer-programming.html

Garun, N. (2012, May 16). *Why learning to code is not just a horrible trend*. Retrieved June 10, 2013, from Digital Trends: http://www.digitaltrends.com/computing/why-learning-to-code-is-not-just-a-horrible-trend/

Gordon, V. S., & Bieman, J. M. (1997). Rapid Prototyping: Lessons Learned. In C. F. Kemerer, *Software Project Management: Readings and Cases* (1 ed.). Chicago: Irwin Publishing.

Heed this warning! Don't miss important computer alerts. (2007, February 8). *ISMP Medication Safety Alert!, 12*(3), 1-2.

Ibrahim, R. B., Bahgat-Ibrahim, L., & Reeves, D. (2010, March 15). Mandatory pharmacy residencies: one way to reduce medication errors. *Am J Health-Syst Pharm, 67*, 477-81. doi:10.2146/ajhp090138

Montreal Protocol. (2012). Retrieved June 10, 2013, from United Nations Environment Programme: Ozone Secretariat: http://ozone.unep.org/Publications/MP_Handbook/MP-Handbook-2012.pdf

Norris, A. C. (2002). Current trends and challenges in health informatics. *Health Informatics Journal, 8*, 205-13. doi:10.1177/146045820200800407

Pedersen, C. A., & Gumpper, K. F. (2008, December 1). ASHP national survey on informatics: Assessment of the adoption and use of pharmacy informatics in U.S. hospitals-2007. *American Journal of Health Systems Pharmacists, 65*, 2244-64.

Phase Out of Combivent Inhalation Aerosol - Questions and Answers. (2013, April 4). Retrieved June 10, 2013, from U.S. Food and Drug Administration: http://www.fda.gov/Drugs/DrugSafety/InformationbyDrugClass/ucm345841.htm

Preventing medication errors. (2010). Retrieved August 8, 2013, from Pharmacist's Letter: http://pharmacistsletter.therapeuticresearch.com/ce/ceCourse.aspx?cs=CPCE~PLPTLPHONLY&s=PL&pv=1&pc=10-305&quiz=

Rich, D. S. (2011). *2011 Joint Commission Medication Management Update.* The Joint Commission.

Schlesselman, L. S. (2011). *10 strategies to reduce medication errors.* Pharmacy Times Office of Continuing Professional Education.

Schmidek, J. M., & Weeks, W. B. (2005, December). What do we know about financial returns on investments in patient safety? *Journal on Quality and Patient Safety, 31*(12), 690-99. Retrieved June 10, 2013

Schuh, M. J. (2009, December 11). *A retail pharmacist examines the consequences of working in a fishbowl*. Retrieved June 15, 2013, from Drug Topics: http://drugtopics.modernmedicine.com/drug-topics/news/modernmedicine/modern-medicine-now/retail-pharmacist-examines-consequences-working-

Silver, J. (2010). *Interruptions in the pharmacy: Classification, root-cause, and frequency*. Retrieved June 10, 2013, from Institute for Safe Medication Practices: http://www.ismp.org/docs/SilverJ_SHS2010.pdf

Standards and FAQ Details. (2008, November 24). Retrieved June 10 2013, from The Joint Commission: http://www.jointcommission.org/standards_information/jcfaqdetails.aspx?StandardsFaqId=57&ProgramId=47

Title 42-The Public Health and Welfare. (2008). Retrieved June 10, 2013, from Clean Air Act: http://www.gpo.gov/fdsys/pkg/USCODE-2008-title42/pdf/USCODE-2008-title42-chap85.pdf

APPENDICES

Appendix A: Combivent Conversion Code

```
Sub Combivent_Conversion()
' ***********************************************************************************
' ******** Description: This macro changes an existing order for Combivent MDI to Respimat
' ********        based on the what was in the order for Combivent MDI. It automatically
' ********        assigns 30 day supply, 1 inhaler, and adjusted refills.  The macro maxes out at 20
' ********        rounds and then it must be restarted by the user to decrease potential problems of alert fatigue.
' ********        Most screens are not shown to the user unless they are patient specific.
' ***********************************************************************************
    Dim LF As String                    ' Chr(rcLF) = Chr(10) = Control-J
    Dim CR As String                    ' Chr(rcCR) = Chr(13) = Control-M
    Const NEVER_TIME_OUT = 0            'prevents macro from running away from user
    Dim drugCombMdi As Integer          'will be used to record what strength of Combivent MDI is being used
    Dim drugCombResp As Integer         'will be used to hold the new drug strenght of Combivent MDI
    Dim numRefills As Integer           'reads the number of refills and will subtract 1
    Dim issueDate As Date
    Dim refillDate As Date
    Dim fillStatus As String            'used to hold the value of H for hold, S for suspend, or U for up-carat
    Dim lastFillDate As Date
    Dim daySupply As Integer
    Dim fillQty As Integer
    Dim addMonth As Integer             'used to decide how many months to add based on days supply
    Dim currentDay As Integer           'used to hold current date of month (i.e. Jan 5, 2013 will be "5")
    Dim currentMonth As Integer         'used to hold current month (i.e. Jan 5, 2013 will be "1")
    Dim Found As Integer
    Dim numLine As Integer
    Dim intervalDate As Integer         'used to hold number of days between fill date and now
    Dim lastFillChk As Date             'used as temp variable to see if future fill date and has 0 refills (means it SHOULD be filled)
    Dim issueDateDiff As Integer        'used for calculation of difference between issuedate and fill date (only used when 0 refills)
    Dim numOffset As Integer            'used for incrementing the macro down the Excel sheet one by one

    LF = Chr(Reflection2.ControlCodes.rcLF)
    CR = Chr(Reflection2.ControlCodes.rcCR)

With Session

'********* Opens excel spreadsheet *********
Dim obj As Object, objWorkBook As Object, objCell As Object
Set obj = GetObject(, "Excel.Application")
obj.Visible = True
Count = 0
numOffset = 0
Set objWorkBook = obj.Worksheets(1)
objWorkBook.Range("A1").Select

'********* Start of loop to max of 20 cycles *********
Do While Count < 21         'sets limit of 20 conversions
    Count = Count + 1       'adds 1 round to total number of complete conversions in current cycle
.CapsLock = True            'turns on capslock

'********* Code to get excel cell info *********
'This copies the range of cells to the clipboard
    obj.ActiveCell.Offset(numOffset, 0).Select 'moves down 1 to select a new active cell
    obj.ActiveCell.Copy                     'copies contents of active cell
    obj.ActiveCell.Interior.ColorIndex = 6     'turns the active cell yellow
    numOffset = 1
    .Transmit "#"
    .Paste                                  'pastes value of cell into Reflections

'********* Resetting all values to be reset so that each loop starts clean *********
    drugCombMdi = 0
    drugCombResp = 0
    numRefills = 0
    issueDate = 0
    refillDate = 0
    fillStatus = ""
    lastFillDate = 0
    daySupply = 0
    fillQty = 0
    addMonth = 0
    currentDay = 0
    currentMonth = 0
    Found = 0
    numLine = 0
    intervalDate = 0
    lastFillChk = 0
    issueDateDiff = 0
```

19

```vb
'********* Start of macro and jumps all the way into patient profile screen *********
    .StatusBar = "Macro Active: RX Patient Status"
    .WaitForString "RX PATIENT STATUS", NEVER_TIME_OUT, rcAllowKeystrokes
    .Transmit CR
    .Transmit "LS" & CR                              'jumps to the end of the patient info screen
    .Transmit CR

'********* Macro searches for combivent and makes sure its on the screen *********
    .WaitForString "Select Action"                   'so that the boxes don't appear before the screen is showing
    .StatusBar = "searching list for combivent"      ' searches the list for combivent
    .Transmit "SL" & CR 'search list
    .Transmit "ALBUTEROL 90/IPRATROP 18MCG" & CR
    .Transmit "N" & CR                               'says no to continuing the search

'********* Screenscraping for combivent issue month, issue date, refill month, refill date, and refills remaining *********
    .Wait "00:00:01"
    Found = .FindText("ALBUTEROL 90/IPRATROP 18MCG", 0, 0)
    If Not Found Then        'combivent MDI not found
        MsgBox "No order for Combivent MDI found, exiting macro."
        Exit Sub
        Else                 'found combivent MDI
            numLine = .GetText(.FoundTextRow, 0, .FoundTextRow, 1)        'captures the line number of combivent
            numRefills = .GetText(.FoundTextRow, 74, .FoundTextRow, 75)   'captures refills remaining
            daySupply = .GetText(.FoundTextRow, 78, .FoundTextRow, 79)    'captures days supply
    End If

'********* Going into order for combivent and screenscraping for issue date and refill date info *********
    .Transmit numLine & CR     'selects line number of combivent to go into prescription info screen
    .Transmit "SL" & CR
    .Transmit "..." & CR
    .Transmit "Issue Date" & CR & CR
    .Transmit "N" & CR
    .Wait "00:00:01"
    Found = .FindText("Issue Date", 0, 0)
    issueDate = .GetText(.FoundTextRow, 22, .FoundTextRow, 29)

    .Transmit "SL" & CR
    .Transmit "Fill Date" & CR
    .Transmit "N" & CR
    .Wait "00:00:01"
    Found = .FindText("Fill Date", 0, 0)
    refillDate = .GetText(.FoundTextRow, 61, .FoundTextRow, 68)

    .Transmit "SL" & CR
    .Transmit "Last fill date" & CR
    .Transmit "N" & CR
    .Wait "00:00:01"
    Found = .FindText("Last Fill Date", 0, 0)
    lastFillDate = .GetText(.FoundTextRow, 22, .FoundTextRow, 29)

    If daySupply = 25 Then       'normalizes day supply so it doesn't fail out later
        daySupply = 30
    ElseIf daySupply = 75 Then
        daySupply = 90
    ElseIf daySupply = 50 Then
        daySupply = 60
    End If

'********* Check to make sure refills are not zero *********
    If numRefills < 0 Then
        MsgBox ("Something is wrong... no value could be calculated from refills, exiting macro.")
        Exit Sub
    ElseIf numRefills = 0 And lastFillDate < Now Then
        MsgBox ("There are no refills remaining on this patient's Combivent, exiting macro.")
        Exit Sub
    End If

'********* Refill date calculations to fill for desired time and rx status plan (hold, suspend, up-arrow) *********
'********* NOTES: an intervaldate<1 (i.e. negative), that means fill date is in the future. If intervaldate>0, then
'*********        that means the fill date was in the past
    Select Case daySupply
        Case 14
            If lastFillDate <> 0 Then
                intervalDate = DateDiff("d", lastFillDate, Now)
                If intervalDate > 0 And intervalDate <= daySupply Then 'if lastfilldate is earlier than today but not w/in daysupply
                        fillStatus = "U"     '"^" out of prescription
                    ElseIf intervalDate > daySupply Then 'if lastfilldate is earlier than today but greater than daysupply
                        fillStatus = "H"     'holds prescription
```

20

```vba
                    Else   'if lastfilldate is in the future or today
                        fillStatus = "S"     'suspends prescription
                        numRefills = numRefills + 1 'compensates for it being filled in the future since it has already
                                            'subtracted refill date
                End If
        Else
            fillStatus = "U"
            MsgBox "no last fill date"
        End If
        If numRefills > 1 Then
            numRefills = numRefills - 1
        End If
        'MsgBox "Case14,day supply: " & daySupply & ", number of refills: " & numRefills & ", fill status: " & fillStatus
Case 30
        If lastFillDate <> 0 Then
            intervalDate = DateDiff("d", lastFillDate, Now)
            If intervalDate > 0 And intervalDate <= daySupply Then 'if lastfilldate is earlier than today but not w/in daysupply
                    fillStatus = "U"     '"^" out of prescription
                    'MsgBox fillStatus
                ElseIf intervalDate > daySupply Then 'if lastfilldate is earlier than today but greater than daysupply
                        fillStatus = "H"     'holds prescription
                        'MsgBox fillStatus
                    Else   'if lastfilldate is in the future or today
                        fillStatus = "S"     'suspends prescription
                        numRefills = numRefills + 1 'compensates for it being filled in the future since it has already
                                            'subtracted refill date
                        'MsgBox fillStatus
                End If
        Else
            fillStatus = "U"
            MsgBox "no last fill date"
        End If
        If numRefills > 1 Then
            numRefills = numRefills - 1
        End If
Case 60
        If lastFillDate <> 0 Then
            intervalDate = DateDiff("d", lastFillDate, Now)
            If intervalDate > 0 And intervalDate <= daySupply Then 'if lastfilldate is earlier than today but not w/in daysupply
                    fillStatus = "U"     '"^" out of prescription
                ElseIf intervalDate > daySupply Then 'if lastfilldate is earlier than today but greater than daysupply
                        fillStatus = "H"     'holds prescription
                    Else   'if lastfilldate is in the future
                        fillStatus = "S"     'suspends prescription or today
                        numRefills = numRefills + 1 'takes into account that a suspended rx will have one less refill because
                                            'its queued up
                End If
        Else
            fillStatus = "U"
            MsgBox "no last fill date"
        End If
        daySupply = 30   'changes prescription to 30 day supply
        If numRefills <> 0 Then
            numRefills = (numRefills * 2) - 1
        End If
        'MsgBox "Case60,day supply: " & daySupply & ", number of refills: " & numRefills & ", fill status: " & fillStatus
Case 90
        If lastFillDate <> 0 Then
            intervalDate = DateDiff("d", lastFillDate, Now)
            If intervalDate > 0 And intervalDate <= daySupply Then 'if lastfilldate is earlier than today but not w/in daysupply
                    fillStatus = "U"     '"^" out of prescription
                ElseIf intervalDate > daySupply Then 'if lastfilldate is earlier than today but greater than daysupply
                        fillStatus = "H"     'holds prescription
                    Else   'if lastfilldate is in the future or today
                        fillStatus = "S"     'suspends prescription
                        numRefills = numRefills + 1 'takes into account that a suspended rx will have one less refill because
                                            'its queued up
                End If
        Else
            fillStatus = "U"
            MsgBox "no last fill date"
        End If
        daySupply = 30   'changes prescription to 30 day supply
        If numRefills <> 0 Then
            numRefills = (numRefills * 3) - 1
        End If
        'MsgBox "day supply: " & daySupply & ", number of refills: " & numRefills & ", fill status: " & fillStatus
Case Else
    MsgBox "This prescription does not have a normal day supply (either 14, 30, 60, or 90). Must be converted _
```

```
                manually. Exiting macro."
                    Exit Sub
            End Select

'********* Starts processing order to change CombiventMDI to Combivent Resp *********
        .Transmit "1" & CR
        .Transmit "ALBUTEROL/IPRATROPIUM,RESPIMAT" & CR & CR & CR

        .StatusBar = "Waiting for Prompt: Do You want to Edit the SIG? NO// Y"
        .WaitForString "Edit the SIG? NO//", NEVER_TIME_OUT, rcAllowKeystrokes
        .Transmit "YES" & CR
        .Transmit "1 PUFF" & CR

        .StatusBar = "Waiting for Prompt: You entered 1 PUFF is this correct? Yes//"
        .WaitForString LF & "You entered ", NEVER_TIME_OUT, rcNoDisplay
        .Transmit CR

        .StatusBar = "Waiting for Prompt: VERB: TAKE//"
        .WaitForString LF & "VERB:", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "INHALE" & CR

        .StatusBar = "Waiting for Prompt: ROUTE: ORAL//"
        .WaitForString LF & "ROUTE", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "ORAL" & CR

        .StatusBar = "Waiting for Prompt: Schedule:"
        .WaitForString LF & "Schedule", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "QID PRN" & CR

        .StatusBar = "Waiting for Prompt: LIMITED DURATION (IN MONTHS, WEEKS, DAYS, HOURS OR MINUTES):"
        .WaitForString LF & "LIMITED DURATION (IN MONTHS, WEEKS, DAYS, HOURS OR MINUTES): ", NEVER_TIME_OUT, rcNoDisplay
        .Transmit CR

        .StatusBar = "Waiting for Prompt: CONJUNCTION:"
        .WaitForString LF & "CONJUNCTION: ", NEVER_TIME_OUT, rcNoDisplay
        .Transmit CR

'********* Fixes all the parts of the prescription in the main area *********
        'Fixes issues date
        .StatusBar = "Fixing issue date"
        issueDateDiff = DateDiff("d", issueDate, lastFillDate)
        If numRefills = 0 And issueDateDiff > 30 Then          'used b/c if 0 refills and issuedate>30 days, errors out
            issueDate = DateAdd("d", -30, lastFillDate)
        End If
        .WaitForString LF & "Select Action", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "3" & CR
        .Transmit issueDate & CR

        'Fixes days supply
        .StatusBar = "Waiting for Prompt: Select Action: Next Screen//"
        .WaitForString LF & "Select Action: Next Screen// ", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "7" & CR
        .Transmit daySupply & CR

        'Fixes refills
        .StatusBar = "Fixing refills"
        .WaitForString LF & "Select Action", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "9" & CR
        .Transmit numRefills & CR

        'Fixing quantities
        .StatusBar = "Fixing Quantities"
        .WaitForString LF & "Select Action: Next Screen// ", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "8" & CR
        .Transmit "1" & CR   'sets quantity to 1

        'Fixing route -- ensures routed for mail
        .StatusBar = "Fixing route"
        .WaitForString LF & "Select Action", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "10" & CR
        .Transmit "MAIL" & CR

        'Fixes patient instructions
        .StatusBar = "Fixing patient instructions"
        .WaitForString LF & "Select Action", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "6" & CR
        .Transmit "FOR SHORTNESS OF BREATH" & CR

        'Adds comment
```

```
        .StatusBar = "Adding comments"
        .WaitForString LF & "Select Action", NEVER_TIME_OUT, rcAllowKeystrokes
        .Transmit "14" & CR
        .Transmit "..." & CR
        .Transmit "CHANGED TO FORMULARY ITEM" & CR & CR

        'Fixes fill date
        .StatusBar = "Fixing fill date"
        .WaitForString LF & "Select Action", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "4" & CR
        If lastFillDate = 0 Then    'if the lastFillDate is blank, it sets to refillDate which can't be null
            lastFillDate = refillDate
        End If
        .Transmit lastFillDate & CR

        'Should be all done with changes now
        .StatusBar = "All done now"
        .Transmit "LS" & CR
        .Transmit "AC" & CR

        .StatusBar = "Waiting for Prompt: Nature of Order"
        .WaitForString LF & "Nature of Order", NEVER_TIME_OUT, rcNoDisplay
        .Transmit "POLICY" & CR

        .StatusBar = "Waiting for Prompt: WAS THE PATIENT COUNSELED: NO//"
        .WaitForString LF & "WAS THE PATIENT COUNSELED", NEVER_TIME_OUT, rcNoDisplay
        .Transmit CR

        .StatusBar = "Waiting for Prompt: Do you want to enter a Progress Note"
        .WaitForString LF & "Do you want to enter a Progress Note", NEVER_TIME_OUT, rcAllowKeystrokes
        .Transmit CR
        .Clear rcScreen
        'may ask for service connection stuff here, requires pharmacist intervention

        .StatusBar = "Waiting for Prompt: Is this correct"
        .WaitForString LF & "Is this correct? YES// ", NEVER_TIME_OUT, rcAllowKeystrokes
        .Transmit CR

'********* Decides what to do at the very end (hold, suspend, up arrow) *********
        Select Case fillStatus
            Case "U"    'used to not fill prescription, but not put it on hold either
                .Transmit "LS" & CR & CR
                .Transmit "^" & CR
            Case "H"    'puts prescription on hold
                'Searches list for combivent so that it shows up on the users screen
                .Transmit "FS" & CR 'goes to top of screen so can search from the top and not miss it
                .Transmit "SL" & CR 'search list
                .Transmit "ALBUTEROL 100/IPRATRO 20MCG" & CR
                .Transmit "N" & CR 'says no to continuing the search
                .Wait "00:00:03" 'must be 3
                Found = .FindText("ALBUTEROL 100/IPRATRO 20MCG", 0, 0)     'looks for combivent to capture the line number
                If Not Found Then    'combivent not found
                    MsgBox "No order for Combivent Resp found, something went wrong, exiting macro."
                    Exit Sub
                Else 'found combivent
                    numLine = .GetText(.FoundTextRow, 0, .FoundTextRow, 1) 'captures the line number of combivent
                End If

                .Transmit numLine & CR
                .Transmit "HD" & CR        'hold
                .Transmit "I" & CR         'I = policy in DHCP
                .Transmit "99" & CR        'Clinical (other)
                .Transmit "PT WILL CALL WHEN WANTED,FORMULARY CHANGE" & CR   'comments entered for holding
                .Transmit "LS" & CR
                .Transmit CR
            Case "S"    'suspends prescription
                .Transmit "LS" & CR & CR
            Case Else    'something that went wrong, should never happen
                .Transmit "LS" & CR & CR
                MsgBox "Something went wrong with routing, but the prescription is complete and accurate, routing for suspense."
        End Select
    Loop    'loops back to very top for next patient
End With
End Sub
```