

ArcBITS Newsletter

Inside this issue:

| | |
|-------------------|---|
| Updox and Spanish | 1 |
| Egg Hunt Update | 1 |
| Non-spawning | 2 |

ArcSys Hot Tip

Next Egg Hunt comes up April 1-3, 7, 15 and 27. So far four users, JEF, NGILES, LDB and VGG are in the running for the \$100 gift card.

You ask, "What Egg Hunt?" See last February's newsletter for details. Simply, when you see Tweetie, Dollar Sign or Rain Cloud that appear in April, click on it to see some "fun" sayings.

You've nothing to lose and you're sure to get a chuckle.

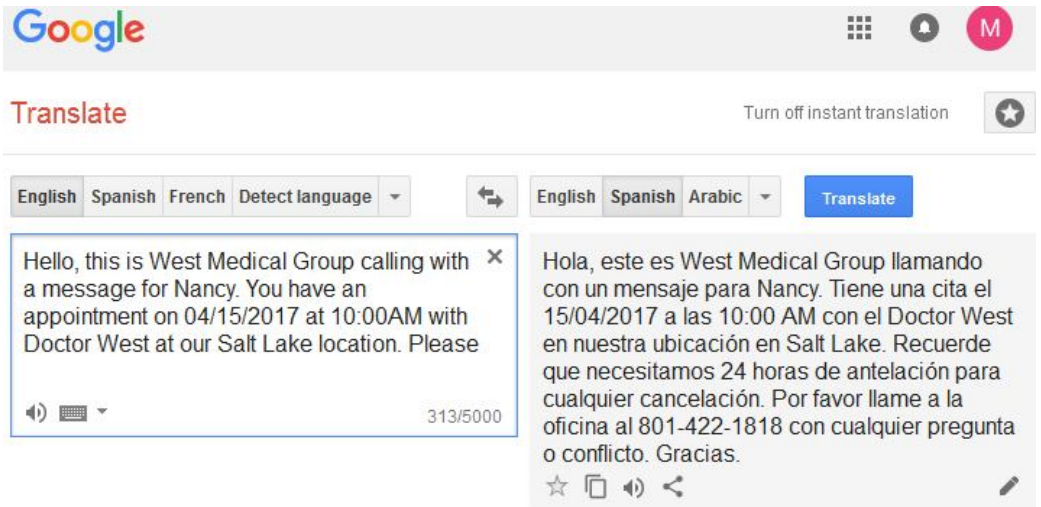
¡Hola! ¡Updox habla español!

If you want to have the capability to send text or email messages by way of Updox to your patients who are listed as speaking another language, the capability is now available through Red Planet. And, it is pretty simple to set up.

First recommended step is to get an idea of how many people in your patient data base speak a specific language. This is easily handled by typing in a statement like:

```
COUNT CM WITH LANGUAGE EQ "SPA" AND WITH LASTCHRGDATE > "1/1/2016"
```

If the number is significant your next step is to take one of your existing text or email messages, copy the contents to the Windows clipboard and then go to the Google Translate web site. Paste the contents of the message in the left window, click the language in the right window and click translate. (The translate shown here is to give you an idea how



easy it is to get an idea of what the message would look like.)

You then highlight the text in the right window and do a ctrl-C. Back inside Updox you will create a new script from an existing entry and paste the content. You would then click on "Create new based on this script". Make note of the script number assigned by Updox.

When you send your request, you include the new script number with the language as shown on the right. The first number is always the English script.
¡Muy fácil!

| | |
|--|-----------------|
| Download which date | 03/07/2017 |
| Contact DATE or leave blank for send immediately | |
| Contact TIME or leave blank for send immediately | |
| Enter DOCTOR, or (DR1,DR2,...), or * for all | 2,6,7 |
| Updox script ID for patient preference of EMAIL | 17984,17992-SPA |
| Updox script ID for patient preference of TEXT | 17741,17993-SPA |
| Updox script ID for patient preference of VOICE | 17550 |





And You Thought Only Fish Spawned

For whatever reason, it is hard for users of Red Planet to get their brain wrapped around the concept of spawning. Spawning has its origins from the old MegaWest days. Because the concept of windows didn't exist at the time MegaWest was created, we used the notion of "dot" commands to move around. The "." command would file a record, the ".X" command would exit from a screen. Over time, various letters of the alphabet would get used up as they were implemented into various functions. Along came the need to start up another screen and so the idea of "spawning" was born (heh, heh). That is, a ".S" command would form a whole new session. From that session another could be spawned and the process could repeat for ever.

To return from where you were, you would use the ".X" command or just hit the escape and eventually you would return from where you started. Sometimes we would use the notion of teleporting yourself to a whole new world.

When Windows browsers like Internet Explorer came along, they introduced the notion of a Back and a Forward button. From one page you could link to another page and then another. Using the Back button would return you to the starting point. Once in a great while, you could start on page A, go to B, go to C, go to D, and then go to B. Where should you go if you press the Back key repeatedly? To A? Or, back to D? Depending on the implementation of the browser, you might get into an endless loop and never ever return to A.

So how does Red Planet deal with this scenario? The main crux of the idea with spawning is that each time you spawn, you create a whole new working environment. When you escape, you return to the previous working environment. You never end up in an endless loop. *Spawning is handled by the buttons which appear at the bottom of a screen.*

After all that explanation, you should now have a really clear idea of what happens with spawning. But, for those who don't like the idea of spawning, there is now a new feature that can be configured for each user where spawning is totally inhibited. What this new feature means is that there is only one instance of each screen that can ever exist.

Let's look at the following scenario.

If you start in patient registration, go to scheduling, go to an appointment and now you want to go to the patient registration.

Under spawning:

You are given a message saying that you are about to launch another copy of the patient registration. If you click "Yes" then a *new instance* is displayed. (Think of yourself as being teleported to your favorite planet, Mars.)

When you escape, you *return* to appointment screen and it re-displays.

Under non-spawning:

The patient registration screen appears. It is blank. You can recall the patient you were last working on by clicking Previous or entering "-" in the Patient field. When you escape, you do not go back to the appointment screen but are returned to the menu. If you want to continue with scheduling you must click the Schedule button.

Sound simple? To turn the feature on or off, click the Rainbow icon from the menu. Choose your color scheme, choose the palette mix, and then the following screen appears. Choose spawning by selecting Y or N in the last field.

