# Building the solution

There are two solutions for “CareEnable”

CareEnableWithCAB

In TFS on Butters

at $/Mears Care V1/MCSSourceCode/DevStream/MobileClient/CareEnable

and MearsCommsWithCAB

at $/Mears Care V1/MCSSourceCode/DevStream/MobileClient/MearsComms

Source control also contains CareEnable and MearsComms solutions that do not contain the CAB project. Those are the solutions that I set up on Butters ages ago but did not actually use to follow the correct promote, build and release procedures. Those two solutions will be very out of date. So if you do go down the proper promote, build and release procedure, I’d copy the “WithCABS’s solutions and then remove the CAB projects.

Do not rely on a build of the solution to build the cab file. I often build the cab and that seems to force everything to be built. Ignore the Setup project that is no longer used.

# Distributing the CAB file

The normal folder for the latest live cab file (not a test version) is

\\Pinky\DevArea\Care\CareEnable CABs for

Note it contains both cabs (CareEnable and MearsComms)

Having dropped the CAB in the above folder, you then have to email people to say it is there. At a minimum this should be Alix, Neil, Phil Bannister , Mark Walker and sometimes Mark Blyth, Mike Hancorn and possibly others that Alix can suggest.

The latest test CAB’s go here

\\Pinky\DevArea\Care\CareEnable CABs for Test

I just put empty text files in there with the file name showing the version number for the cab’s. Unlike with MearsEnable, there is probably no need to have lots of test versions but I do keep the folders in case we need to drop back.

Having dropped a test CAB in the above folder, you then have to email people. That could be all of the above but possibly not Mike Hancorn who is a trainer and not a tester. Officially Phil is not a tester.

# Creating and distributing the OTA Update Zip file

The process of creating the zip file is the same as for MearsEnable with a similar A-Zippit.bat and all its same problems.

We have one for CareEnable and one for MearsComms. In the Care system, the OTA process is more flexible. It can send any zip file name to any PDA (employee number). But I use a similar convention for file name. EG

CareEnable-0.9992.zip

MearsComms-0.986.zip

Those file names must be copied to the correct folders. On the test system the services run on Pinga so the update folder is

\\Pinga\E$\Care\FilesToPDAs\OTA Program Updates

There are two live servers for the input and output services so the two folders that must contain the OTA file are

\\FernMcsPda01\D$\Care\FilesToPDAs\OTA Program Updates

and

\\FernMcsPda02\D$\Care\FilesToPDAs\OTA Program Updates

This could be improved so both servers look at one folder on one server

There is no CareEnable Manager so currently sending an OTA update to a PDA is via SQL scripts to write the required record to the output queue table. This is picked up by the Care Output Service and sent to the PDA’s. In reality apart from test versions, all Carers should be on the same latest version of CareEnable and MearsComms. In fact I just remembered I wrote a sproc to update a specific PDA (EmployeeNo). The text of the email I sent describing it is as follows:

------------------------------------

There is no way yet to perform an OTA update to a PDA in a nice friendly way but that will appear in time. For now it has to be done from SQL. I've just written a sproc to make it easier. The sproc is new and currently only on the test comms database MearsCareDeviceComms. The sproc is called SendOTAUpdate and takes up to 2 parameters as follows:

SendOTAUpdate 'J999137', X

'J999137' is an example EmployeeNo to receive the OTA update

X is optional and can be one of 3 values

If X is omitted completely, records are created in the output queue if they do not exists but it does not cause the update to be sent. To send the update additional manual SQL is required to set the SendStatus field in the output record to -1.

If X is set to C, an update is queued to CareEnable

If X is set to M, an update is queued to MearsComms

If X is set to U, an update is queued to AppUpdater. So far AppUpdater has never been updated. It is the program that allows CareEnable and MearsComms to appear to be updated while they are running. I would strongly suggest you never try to update this program at present but there is no need to do that anyway.

In summary:

Open SQL

To update CareEnable for EmployeeID J999137 the command is

EXEC SendOTAUpdate 'J999137', C

To update MearsComms for EmployeeID J999137 the command is

EXEC SendOTAUpdate 'J999137', M

Before updating either program, it is advised that both CareEnable and MearsComms are running on the PDA. For an immediate update of CareEnable, leave it displaying the home screen. Although it should cope with kicking off an update to both programs at the same time, I tend to update them one at a time. For example, update MearsComms first and once its new software version is sent back and updated into tblDeviceDetails, then update CareEnable.

------------------------------------

That sproc creates a record in the output table for the specified Employee No by copying a default record from a pseudo employee number of !Default!

You first have to ensure that record (one for each update type CareEnable, MearsComms and AppUpdater) has the correct OTA update file name in it (latest version to be sent). What I tend to do is run a simple script to update all the file names in all the records in the relevant records in the output table. EG

UPDATE tblMessageQueueOut SET [FileName] = 'D:\Care\FilesToPDAs\OTA Program Updates\CareEnable-0.9972.zip' WHERE MessageType = 'AppUpdateAvailable'

UPDATE tblMessageQueueOut SET [FileName] = 'D:\Care\FilesToPDAs\OTA Program Updates\MearsComms-0.9972.zip' WHERE MessageType = 'CommsUpdateAvailable'

UPDATE tblMessageQueueOut SET [FileName] = 'D:\Care\FilesToPDAs\OTA Program Updates\AppUpdater-100.zip' WHERE MessageType = 'UpdUpdateAvailable'

Note Well! The file name path is on the server running the output service. That is the path to the filename that the output service will pick up. See above for description of those folders.

The sproc I mentioned is really for people to easily do an OTA update to one or a few PDA’s. If you are sending to lots of PDA’s it is faster to do it all from SQL. Create the records via a load of inserts and then set the SendStatus in the record to -1 to get it sent.

To see all OTA records like the above, run this SQL

SELECT O.QueueOutID, O.EmployeeID, O.MessageType, O.FileName, O.WhenQueued, O.WhenSent, O.SendStatus, O.WhenReceived, O.NextRetryTime

FROM tblMessageQueueOut O WITH (NOLOCK)

WHERE (MessageType LIKE '%Available%')

ORDER BY O.EmployeeID, O.MessageType

It is VERY VERY rear to update AppUpdater. It has only had one update in a year and that was a long time ago.CareEnable gets most updates and MearsComms, far less – hopefully.

Try not to make a change that requires a database change. If that must be done, ensure both MearsComms and CareEnable are running then update MearsComms and after that has updated, update CareEnable soon after.