

To: len kawell @ iris, laura fay, linda welsh, mike palone @ iris, barry briggs, mussie shore,
laura fay, ray ozzie @ iris
cc: papows, john landry, zisman (bcc: Tom Lemberg)
From: Michael Zisman
Date: 03/15/95 11:07:59 PM
Subject: MAPI Discussion with Microsoft

I had two long conversations today with Rob Shurtleff, GM of the Workgroup Solutions Unit (Exchange, etc.) at Microsoft about the "proprietary extensions to MAPI" issue that has been raised in the press. Here is what MS is doing, as explained by Rob.

The basic mail client bundled with Win'95 delivers a set of functions that are completely covered by MAPI, as published. MS has not added any "secret" MAPI calls, nor has it defined properties that are unpublished and known only to its client and its server.

What it has done, and according to Rob is old news and well documented, is the following. The client is extensible. A service provider (SP) when being loaded, can extend the client by instructing MAPI (or perhaps the client directly) to add menu items or icons to the toolbar. When the user selects these items (which at this point appear to be part of the client) MAPI passes that event to the SP. The SP then directly supplies its own UI by opening a window and interacting with the user.

Rob gave the following example. The Exchange server has a feature to support "out of office" that is not in the basic client in Win'95 or in MS Mail 3.2. The Exchange server, therefore, extends the Win'95 client UI so that a user can select this function. When this happens the Directory SP is notified, generates a window and requests start date, end date, and text for auto-reply message. This UI code is in the SP.

To the extent that MS "extends" the client in this manner, which presumably they will do more and more over time, the client is actually manifested in both the Windows client and the SP. This is not what the world expected, in my opinion, when MS was evangelizing MAPI. The major benefit of MAPI was that it provided a clean separation of client and server so that the customer could make independent decisions about the client and server.

The reality here is that portions of the MS "client" are actually in the server - to me, the SP is simply an extension of the server that moves the interface/abstraction layer up from the on-the-wire protocol to an easier to code API. It is the "top" of the server, if you will. Hence, when someone "replaces the Exchange server" with their own server and SPs, by removing the Exchange SPs you are, in effect, removing part of the client, and you must replace that part of the client with new UI code that presumably has precisely the same look and feel as the MS code being replaced. You cannot replace the server without replacing part of the client.

Continuing with the out-of-office example, let's assume that Notes supports this function. The Notes client provides UI and there are Notes API calls to communicate this function to the server so that the N&A book can be updated. If out-of-office were truly part of the MS Win '95 client, then there would be MAPI calls to communicate with any SP. We could replace the Exchange server, the MS Windows '95 client would make MAPI calls to our SP, we would generate the proper Notes calls and we're done. We could manifest a function in our server that matches the function in their server without UI required. With the approach that MS has taken, when we replace the Exchange SPs, the UI code that deals with out-of-office is gone, and the only way that we can provide the same function to the user (who views the out-of-office dialog box as part of the client) is to write UI code in our SP that provides this dialog box. As MS adds new functions to the "client" by embedding them in the server, all competitors will need to rework their SPs even though they may already have this function in their server and in their client.

EXHIBIT

12/14/09
Shurtleff

In effect, MS creates a lock-in between their "free" client and the Exchange server. I think this is a very serious problem and is inconsistent with the position MS has tried to take with MAPI providing a level playing field by allowing the customer to make independent client and server decisions. ~~They have~~ accomplished this by distributing the client function (i.e., UI) across the client AND the server. While others are free to do similar things (i.e., others can do what MS can do) the fact that MS is bundling the client with Win '95 changes the playing field dramatically. Even though we invest in a client and server, if we want to compete for the back end we will need to invest AGAIN in duplicate client function to provide that part of the MS client that is really in their server. While Rob indicated that there are very few functions that are now in the server portion of the client, MS has a clear path to turn that knob as appropriate.

I do not believe that this is where customers thought they were going today.

Comments? IRIS and MV - is this consistent with your understanding? Is this documented in MAPI SDK or the Exchange developers SDK? Am i missing something here?

I am going to forward a copy of this to Rob, so that he can correct me if I have misunderstood the technical issue.