
From: Ken Cooper (WCCD)
Sent: Friday, November 05, 1999 1:14 AM
To: Bill Gates; Ted Peters
Cc: David Vaskevitch; Jim Allchin (Exchange); Eric Rudder; Brad Lovering (Exchange); Paul Maritz; Craig Mundie; Brian Valentine (Exchange)
Subject: RE: Webforms and Windows Terminal Server

Most of this fits with what we've been thinking lately. We've been trying to categorize the likely scenarios for future UI, and have come up with three clients that make sense to us. Some of this may be beyond our scope, but feedback would help us establish some context.

The Windows+ client - A new rich UI client. The new apps would run in a page-oriented IE shell (which could be run as a shell or as a window in a traditional windows shell) The UI library would be based on com+ for secure extensibility, and would include: a rich win32-like events and painting layer, a control model with pluggable layout, hierarchical databinding, web-ui features (looklessness, emphasis on rich text, links). The application model is page based: apps are loaded, partitioned, and cached in page increments as they are requested. Data could be local, remote, or remote with local syncing and caching for offline use. This Windows+ client could run on top of windows but potentially become its own operating environment. This client would be our client of choice for any device of sufficient power. It would also include an html browser.

Pros: Rich, Scales, Deep tie to our APIs, Can be disconnected
Cons: Speed of install if not cached and app is not partitioned well

The WTS client - Minimal client for dumber devices. Runs a small shell which can run both older Win32 as well as new Windows+ apps by processing only the graphics and input events locally. This client may also handle streaming media. Examples of WTS-only clients would be display-pads or streaming media viewers that had minimal local processing/storage. Not sure how big of a niche this is... This client would also be included in the Windows+ client to a) allow Windows+ users to run other's machine's remotely, and b) allow Windows+ users to access apps/data on another machine (for occasional use) without having to download/install.

Pros: Rich, No App Install, Deep tie to our APIs
Cons: Does not scale. Must be connected. Does not handle low-bandwidth well.

Non-Microsoft client - an HTML 3.2+ browser that we project HTML to via HTTP.

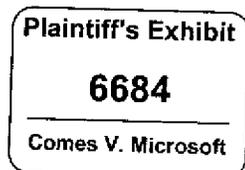
Pros: Reach, scales with stateless server, handles low-bandwidth reasonably
Cons: Limited, non-interactive UI

In the interim, we may support IE 5 specific richness in order to provide a better MS-specific experience before Windows+ ships.

The key is that there is *one* UI framework that developers use to target these clients. Developers author WebForms apps by creating an app as a set of pages. The pages contain controls that can be programmed via properties, methods, and events. Pages can navigate to other pages within the app. Built-in controls are provided that run in both reach (render html) and rich (render Windows+ graphics). App writers can build new controls by compositing these built-ins (and thus get rich/reach) Sophisticated control writers can write new controls that have both rich/reach implementations or just rich (rich text editor, excel grid, adobe premiere video editor). There is one control framework, one eventing model, one declarative persistence format, one designer, one set of docs, etc. It can be programmed in any of the com+ CLS languages.

When running apps on the Windows+ client (as explained above) the interaction logic, controls, rendering

4/30/2003



MS-CC-MDL 000000380573
HIGHLY CONFIDENTIAL

and events run locally. On a WTS client, the app is run on the server, and graphics/eventing/streaming media information is projected to the client. For an HTML client, the app is run on the server, and html is projected to the client.

This is our current thinking. One obvious question that begs: why do you think we should use the richness we have today for highly interactive ui (we assume that's what you mean by "scribble on bitmaps")?

-----Original Message-----

From: Bill Gates

Sent: Thursday, November 04, 1999 6:34 PM

To: Ted Peters; Ken Cooper (WCCD)

Cc: David Vaskevitch; Jim Allchin (Exchange); Eric Rudder; Brad Lovering (Exchange); Paul Maritz; Craig Mundie; Brian Valentine (Exchange)

Subject: Webforms and Windows Terminal Server

I may not be able to describe this properly but I think our Windows Terminal Services capabilities should be part of Webforms. I have written elsewhere that Windows Terminal server should be part of the browser but that is consistent with what I saying here.

Many places in the company people are defining low level protocols to connect to really dumb devices. Windows Terminal Server is just one example of this. (SMS and Netmeeting have overlapping but different functionality). Another example is work done in our Hardware group thinking about little pads you carry around in the home. Another example is our WebTV group playing around with sending video around the home.

I think the idea of Webforms is to have a presentation model that is a) Powerful b) Can be targeted to many target devices.

I am proposing that we think of Windows Terminal Server as being part of this for 3 big reasons.

First I think we need to embrace the idea of very low end terminals – terminals dumber than HTML 3.2. I guess if a terminal just has HTML and no support for our WTS protocol we will have to do full bitmap downloads when there is bitmap scribbling.

[The business issues are tricky but we have patent the protocol and I think we do an encrypted handshake as part of initiating a session. We don't want to give up the "CAL" element of these connections. But we also don't want to give away the RICH CLIENT webforms capability so the business model for richWebforms and for WTS are aligned]

Second I think our "forms/UI/graphics" model needs to let people scribble on bitmaps. I think we should use the richness we have today for this even though it is not perfect.

Third I think that this allows us to position WebForms as evolutionary not a totally break with the past. Applications can project with WTS. It may not use the client fully but it is there.

I am not sure I am really being clear here on why I want this and what it means but I would like to discuss this more.

4/30/2003

MS-CC-MDL 00000380574
HIGHLY CONFIDENTIAL