

Some trivia about the //build/ 2011 conference

devblogs.microsoft.com/oldnewthing/20130401-00

April 1, 2013



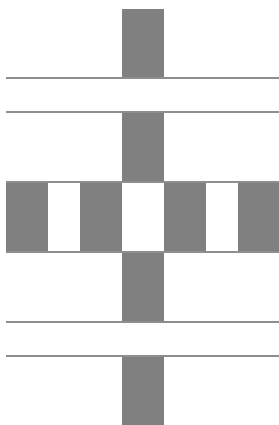
Raymond Chen

Registration for //build/ 2013 opens tomorrow. I have no idea what's in store this year, but I figure I'd whet your appetite by sharing some additional useless information about //build/ 2011.

The internal code name for the prototype tablets handed out at //build/ 2011 was *Nike*. I think we did a good job of keeping the code name from public view, but one person messed up and accidentally let it slip to Mary-Jo Foley when they said that the contact email for people having tax problems related to the device is nikedist@microsoft.com.

The advance crew spent an entire week preparing those devices. One of the first steps was unloading the devices from the pallettes. This was done in a disassembly line: The boxes were opened, the devices were fished out, then removed from the protective sleeve. At the end of this phase, you had one neat stack of boxes and one neat stack of devices.

The advance crew also configured the hall so they would be ready to start once Redmond sent down the final bits of the Developer Preview build. The hall was divided into sections, and each section consisted of eight long tables. Four of the tables were arranged in a square, and the other four tables were placed outside the square, one parallel to each side, forming four lanes.



Along the inner tables, there were docking stations, each with power, wired access to a private network, and a USB thumb drive. Along the outer tables, there were desk organizers like this one, ready to hold several devices in a vertical position, and next to the organizer was a power strip with power cables at the ready.

In this phase of the preparation, the person working the station would take a device, pop it into a docking station, and power it on with the magic sequence to boot from USB. The USB stick copied itself to a RAM drive, then ran scripts to reformat the hard drive and copy all the setup files from the private network onto the hard drive, then it installed the build onto the machine, installed Visual Studio, installed the sample applications, flashed the firmware, and otherwise prepared the machine for unboxing. (Not necessarily in that order; I didn't write the scripts, so I don't know what they did exactly. But I figure these were the basic steps.) Once the setup files were copied from the private network, the rest of the installation could proceed autonomously. It didn't need any further access to the USB stick or the network. Everything it needed was on the RAM drive or the hard drive.

The scripts changed the screen color based on what step of the process it was in, so that the person working the station could glance over all the devices to see which ones needed attention. Once all the files were copied from the network, the devices were unplugged from the docking station and moved to the vertical desk organizer. There, it got hooked up with a power cable and left to finish the installation. Moving the device to the second table freed up the docking station to accept another device.

Assuming everything went well, the screen turned green to indicate that installation was complete, and the device was unplugged, powered down, and placed in the stack of devices that were ready for quality control.

The devices that passed quality control then needed to be boxed up so they could be handed out to the conference attendees. Another assembly line formed: The devices were placed back in the protective sleeves, nestled snugly in their boxes, and the boxes closed back up.

Now, I'm describing this all as if everything ran perfectly smoothly. Of course there were problems which arose, some minor and some serious, and the process got tweaked as the days progressed in order to make things more efficient or to address a problem that was discovered.

For example, the devices were labeled *preview devices*, but shortly before the conference was set to begin, the manufacturer registered their objection to the term, since *preview* implies that the device will actually turn into a retail product. They insisted that the devices be called *prototype devices*. This meant that mere days before the conference opened, a rush print job of 5000 stickers had to be shipped down to the convention center in order to cover the word *preview* with the word *prototype*. A new step was added to the assembly line: *place sticker over offending word*.

Another example of problem-solving on the fly: The SIM chip for the wireless data plan was preinstalled in the device. The chip came on a punch-out card, and the manufacturer decided to leave the card shell in the box. Okay, I guess, except that the card shell had the SIM card's account number printed on it. Since the reassembly process didn't match up the devices with the original boxes, you had all these devices with unmatched card shells. In theory, somebody might call the service provider and give the account number on the shell rather than the number on the SIM card. To fix this, a new step was added to the assembly line: Remove the card shells. All the previously-assembled boxes had to be unpacked so the shells could be removed. (At some point, somebody discovered that you could extract the shells without removing the foam padding if you held the box at just the right angle and shook it, so that saved a few seconds.)

Now about the devices themselves: They were a very limited run of custom hardware, and they were *not cheap*. I think the manufacturing cost was in the high \$2000s per unit, and that doesn't count all the sunk costs. I found it amusing when people wrote, "What do you mean a free tablet? Obviously they baked that into the cost of the conference registration, so you paid for it anyway." Conference registration was \$2,095 (or \$1,595 if you registered early), which nowhere near covered the cost of the device.

Some people whined that Microsoft should have made these devices available to the general public for purchase. First of all, these are developer prototypes, not consumer-quality devices. They are suitable for developing Windows 8 software but aren't ready for prime time. (For one thing, they run hot. More on that later.) Second of all, *there aren't any to sell*. We gave them all away! It's not like there's a factory sitting there waiting for orders. It was a one-shot production run. When they ran out, they ran out.¹

Third, these devices, by virtue of being prototypes, had a high infant mortality rate. I don't know exactly, but I'm guessing that maybe a quarter of them ended up not being viable. One of the things that the advance crew had to do was burn in the devices to try to catch the dead devices. I remember the team being very worried that the hardware helpdesk at the conference would be overwhelmed by machines that slipped through the on-site testing. Luckily, that didn't happen. (Perhaps they were *too* successful, because everybody ended up assuming that pumping out these puppies was a piece of cake!)

Doing a little back-of-the-envelope calculations, let's say that the machines cost around \$2,750 to produce, and that a quarter of them failed burn-in. Add on top of that a 25% buffer for administrative overhead, and you're looking at a cost-per-device of over \$4,500. I doubt there would be many people interested in buying one at that price.

Especially since you could buy something very similar for around \$1100 to \$1400. It won't have the hardware customizations, but it'll be close.

The hardware glitches that occurred during the keynote never appeared during rehearsals in Redmond. But when rehearsing in Anaheim, the hardware started flaking out like crazy and eventually self-destructing. (And like I said, those devices weren't cheap!) One of my colleagues got a call from Los Angeles: "When you come down here, bring as many extra Nikes as you can. We're burning through them like mad!" My colleague ended up pissing off everybody in the airport security line behind her when she got to the X-ray machine and unloaded nine devices onto the conveyer belt. "Great, I just put tens of thousands of dollars worth of top-secret hardware on an airport X-ray machine. I hope nothing happens to them."

Why did the devices start failing during rehearsals in Anaheim, when they ran just fine in Redmond? Because in Anaheim, the devices were being run at full brightness all the time (so they show up better on camera), and they were driving giant video displays, and they were sitting under hot stage lights for hours on end. On top of that, I'm told that the HDMI protocol is bi-directional, so it's possible that the giant video displays at the convention center were feeding data back into the devices in a way that they couldn't handle. Put all that together, and you can see why the devices would start overheating.

What made it worse was that in order to cram all the extra doodads and sensors into the device, the intestines had to be rearranged, and the touch processor chip ended up being placed directly over the HDMI processor chip. That meant that when the HDMI chip overheated, it caused the touch processor to overheat, too. If you watched the keynote carefully, you'd see that shortly before the machine on stage blew up, you saw the touch sensor flip out and generate phantom touches all over the screen. That was the clue that the machine was about to die from overheating and it would be in the presenter's best interest to switch to another machine quickly. (The problem, of course, is that the presenter is looking out into the audience giving the talk, not staring at the device's screen the whole time. As a result, this helpful early warning signal typically goes unnoticed by the very person who can do the most about it.)

The day before the conference officially began, Jensen Harris did a preview presentation to the media. One of the glitches that hit during his presentation was that the system started hallucinating an invisible hand that kept swiping the Word Hunt sample game back onto the screen. Jensen quipped, "This is our new auto-Word Hunt feature. We want to make sure you always have Word Hunt when you need it. We've moved beyond touch. Now you don't even need to touch your PC to get access to Word Hunt."

Jensen's phenomenal calm in the face of adversity also manifested itself during his keynote presentation. You in the audience never noticed it, but at one point, one of the demo applications hit a bug and hung. Jensen spotted the problem before it became obvious and smoothly transitioned to another device and continued. What's more, while he was talking, he went back to the first device and surreptitiously called up Task Manager, killed the hung application, and prepared the device for the next demo. All this without skipping a beat.

We are all in awe of Jensen.

When he stopped by the booth, Jensen said to me, “I don’t know how you can stand it, Raymond. Now I can’t walk down the hallway without a dozen people coming up to me and wanting to say something or shake my hand or get my autograph!” (One of the rare times we are both in the same room.)

Welcome to nerd celebrity, Jensen. You just have to smile and be polite.

Bonus chatter: What happened to the devices that failed quality control? A good number of them were rejected for cosmetic reasons (scuff marks, mostly). As a thank-you gift to the advance crew for all their hard work, everybody was given their choice of a scuffed-up device to take home. The remaining devices that were rejected for purely cosmetic reasons were taken back to Redmond and distributed to the product team to be used for internal testing purposes.

¹ My group had one of these scuffed-up devices that we used for internal testing. Somebody dropped it, and a huge spiderweb crack covered the left third of the screen, so you had to squint to see what was on the screen through the cracks. We couldn’t order a replacement because *there was nowhere to order replacements from*. We just had to continue testing with a device that had a badly cracked screen.



Raymond Chen

Follow