A Short Description of the ServiceDesk POS system as upgraded March 2008

Prior to this upgrade, POS transactions were initiated much like any job. In other words, you'd put as much initiatory information into a Callsheet as wanted, then select Job/Sale—with one difference. Instead of the normal *left*-click on Job/Sale (or *Alt*-J from the keyboard), you'd do a *right*-click (or *Ctrl*-J). This told the system it was a POS situation, and when the little yellow Create Job/Sale form came up, it was pre-primed to take you to the FinishedForm context, in lieu of the standard scenario.

The old method can still be used, but for any intensive and dedicated POS situation, we now have a much better alternative.

Specifically, the FinishedForm interface now sports a new form, called POS.

The idea, for any station that's performing POS operations, you should open the FinishedForm interface directly (Alt-F4 is the keyboard shortcut). Select the new POS form type, and simply leave it selected. It's the form that will be used for all POS transactions. There will no longer be a need to use Callsheets, or to ever leave this one interface for POS usage.

You'll notice, after having selected the POS form, the system places your cursor in a box that, in previous operations, you only saw used for inputting an invoice number. Now, you'll see, this same box invites you (as an alternative) to input the ID for a Sales Person. The idea is that each POS transaction is *initiated* by an operator inputting his (or her) two-letter name abbreviation.

With that done, you'll see the POS form fills with beginning info, and places the cursor in the appropriate box for the operator to begin listing items being sold. At this point, operations are very much like they were, under the old method, when the transaction was initiated from a Callsheet.

There is a major substantive difference.

Under the old method, a JobRecord was created for each transaction. In most cases, this was overkill, because JobRecords are heavy-duty, designed to handle all the myriad complications and processes that go with a full-fledged repair.

Under the new method (and unless parts are being *ordered*), no JobRecord is created. The sole document that fully describes the transaction will be the POS ticket itself. Naturally, ServiceDesk automatically saves an electronic copy, and each is instantly accessible should the need arise. But (and, again, unless parts are being *ordered)* there's no JobRecord; hence these transactions will neither enter nor be exposed via the CstmrDbase system.

As indicated, the exception is if parts are being ordered (a fact that, as before, is still indicated by placing a double asterisk next to an item's part number or description). Upon seeing this, the system will notice the intent to order, and solicit user consent for

an “on-the-fly” creation of a JobRecord. It will simultaneously attend to such concerns as that the default fill-in for Customer’s Name (i.e., “COUNTER SALE”) is not appropriate when ordering parts, and prompt the user to change.

Of course, upon printing it will also solicit user consent to create an internal parts request (which, in fact, is identical to past structure).

Also, continuing with prior structure, ServiceDesk will note if parts are being used from stock, and upon printing offer to actually pull the items from inventory.

In addition, the SalesJournal-Entry function can (and likely should) still be linked, just as before. To review, you can setup that linkage by right-clicking on the radio-button (to toggle its text color to red) for any of the form Types for which you’d like it to occur. We definitely suggest doing this for the POS form Type. In other words, make sure you’ve toggled its radio-button to red, so when any operator completes a sale, they’ll automatically be linked to enter the transaction to your SalesJournal.

A general difference, with this upgrade, is there are now explicit buttons for each of the above three actions (i.e., creating an internal parts request, pulling from inventory or entering to the SalesJournal). You’re not likely to need them in typical operation (since each process can invoke automatically when you go to print). But should the odd need arise to invoke any by itself, you can use its button for the purpose.

You’ll notice, after first typing in your two-letter abbreviation to initiate a transaction, the POS form’s *InvoiceNumber* box auto-populates with the phrase “ToBePulled.” This signifies nothing is yet official. The system has not yet pulled an invoice number to assign the ticket. In fact, even as you begin filling in boxes with the items you’re intending to sell, nothing is recorded (and no invoice number is pulled) until either you click on ‘Print’, or do some other action (such as entering Funds received) for which an invoice number is needed.

Regardless of how triggered, you’ll see that at such point that “ToBePulled” text is replaced with an actual number. Simultaneously, the system saves a copy of your ticket, so there’s an instant and permanent record of the ticket that’s associated with that invoice number. You could still abort the sale at this point, but the record will remain regardless.

In regard to the invoice number that’s pulled, you’ll notice that (unless parts are being ordered) the system makes it a *negative* number (i.e., puts a *minus* sign in front). This is so, internally, ServiceDesk can distinguish the reference as one where there’s no expectation for an accompanying JobRecord. To state it differently, the negative number denotes it as involving a Raw-POS, no-JobRecord situation.

Please keep the above in mind if you want to look up a ticket that was created via this new option. If, on a Load-InvoiceNumber request, you forget to put the minus sign in front (and assuming it’s needed), the system won’t find the ticket.

As one more detail, you'll notice the system puts plain text in the top of the form for your company name, address and telephone numbers. If you'd prefer having a graphic image there, instead, it's very easy to accomplish.

Using any graphics program, create the image you want. Try to make its proportions (height compared to width) similar to what's in the POS form's company-info box (ultimately, the system will shrink/stretch whatever you create to fit that space). Save the image as *MyLogo.BMP* in the \sd\netdata folder on your server. That's it. ServiceDesk will see the file, and load it rather than the plain text.