

# Lawson User Group Atlanta, GA

- Section 1

1:30pm to 2:25pm

MSPs/CTPs and Lawson  
Customization

- Section 2

2:35pm to 3:30pm

Customization of Lawson  
with Minimal Code  
Changes

Presented by:

Todd Kastle – Kastle Consulting

# Introduction

“Lawson Made Simple”  
or  
“Simply Lawson”

Really?

How do we do that?

# Introduction

My Information:

Todd Kastle

California

(408) 621-9583

But don't call me! Call:

Bill Mitchell

California

(530) 889-8550

# Introduction

More on Todd...

What about you?

# CTPs and Lawson Customization

## “Normal” CTP Process

- Transfer CTP File from Lawson’s site to your Lawson Server
- Run lasetup (overwrites your changes)
- Reapply your changes

# CTPs and Lawson Customization

Here's what lasetup does

- Extracts files from patch.tar
- Compares version of each file
  - If newer, then existing file is overwritten
  - If older, does nothing
  - If same version...
- Creates cksum on existing file and compares cksum – **this is what effects your changes!**

# CTPs and Lawson Customization

So what if you don't want lasetup to look at customization, but instead ONLY look at the version of the files?

Based on what we know of how lasetup replaces the files the only logical choice is to change the cksum value in the file's header.

# CTPs and Lawson Customization

Here' what I mean...  
Check out the header for HR11

```
***** HR11PD 27.2.31 <1198086378> ← cksum
000100*****
000200*                HR11PD                *
000300*****
000400*****
000500 HR11S1-TRANSACTION                SECTION 24.
000600*****
000700 HR11S1-START.
```

# CTPs and Lawson Customization

Here's the way I'd approach customizations:

- 1) Make backup copies of your original and modified files
- 2) After changes (that you've tested) reset the cksum value in each modified file
- 3) Run lasetup on preview mode
- 4) Based on results of preview decide what you will do

# CTPs and Lawson Customization

Backup Strategy...

When to reset cksum?

Document your changes!

# CTPs and Lawson Customization

## The handout

- Page 3 – lasetup in preview mode
- Page 4 – review changes
- Pages 5-7 – 5 levels of change documentation and impact analysis
- Pages 8-9 – Installing the CTP
- Pages 10-11 – Updating cksum (updcksum)

# CTPs and Lawson Customization

How do update the cksum value?

Easy!

lasetup has the cksum function  
(it's almost 100 lines in that function alone)

I've taken that function and used it to create a handy perl script that you can run on each file.

Check it out and download it online!

# CTPs and Lawson Customization

## Conclusion

Website for Reference:

<http://www.lawsontech.com>

Todd's Email:

[todd.kastle@kastleconsulting.com](mailto:todd.kastle@kastleconsulting.com)

# Break

Please be back in 10 minutes

# Customization with Minimal Code Changes

Before I go further let me address on thing:

Changing standard Lawson code

vs.

Changing *a copy* of standard Lawson code

How do I copy a Lawson Program?

# Customization with Minimal Code Changes

Overview – here's what we'll cover

- Creating a custom table and “linking it” to one of Lawson’s primary tables
- Change Form Definition File (.scr) to show new field(s)
- Create Ending User Exit to process data
- Add customized report to standard Lawson report
- Use of libraries to minimize actual code changes
- (optional) Show another type of User Exit

# Customization with Minimal Code Changes

## Create EMPLOYEE2

- dbdef → Files
- EMPLOYEE2 (EM2)  
Company, Employee, Email-Address
- Create Primary Index
- blddbdict  
(yes we're ignoring relations for the sake of time)
- dbreorg -l
- dbreorg

# Customization with Minimal Code Changes

## Change HR11.scr

- backup HR11.scr to HR11.scr.orig
- Edit HR11.scr (led <PL> hr11 – select C – Form)  
Email Address on Main tab (last field)  
EM2-EMAIL-ADDRESS – AL – 60 – SB
- Compile, check for error log, check out new form

# Customization with Minimal Code Changes

## User Exits

- Two Categories:
  - 1) Form User Exits
  - 2) Payroll User Exits
- I will ONLY cover “Form User Exits” today, but will point out the others.

# Customization with Minimal Code Changes

## User Exits (cont.)

- **Beginning**  
(run when Action selected, but before ANY processing)  
Note: CAN set error state
- **Middle**  
(run after all Edit Processing, but before Action processing)  
Note: CANNOT set error state
- **Ending**  
(run after ALL processing) Note: error state does not matter

# Customization with Minimal Code Changes

Let's Make an Ending User Exit!

- Copy HR11PD → HR11EPD  
Copy HR11WS → HR11EWS
- Change HR11EWS – remove all code!
- Change HR11EPD – remove between  
HR11S-TRANSACTION Sections
- Create Logic in HR11EPD  
Need logic for “A”, “C”, “D”, and “I” Actions
- Compile with “-u” option – check for errors

# Customization with Minimal Code Changes

The ONE Place I'm stopping for Questions

Any question on:

Why I created "Secondary"  
The workings of a "Form" User Exit  
Anything else covered in this section

# Customization with Minimal Code Changes

Add Custom Report to Standard Lawson

This example will be used to show ways to change Lawson's programs while making very minor changes to the actual program code.

While this may not be a "useful" example it should be something you can build on.

# Customization with Minimal Code Changes

## Add Custom Report to Standard Lawson

- Create new workfile for report output
- Change Report Definition File (.rpt) to include new report for output
- Change Procedure Division (PD) to:
  - 1) Open and Close new work file
  - 2) Call Routines to print page heading
  - 3) Call Routines to print page records
- Compile the program

# Customization with Minimal Code Changes

## Add Custom Report to Standard Lawson (cont.)

- Library Routines? What do you mean?
- Create PD and WS (if needed) code in pdlib/wslib directories, run libdef to define them in Lawson.

# Customization with Minimal Code Changes



Final Conclusion