

# Planning a Successful Upgrade to Siebel 8.0

Robert Ponder and Subodh Patra  
Ponder Pro Serve

# If We Just Had One Slide

- Spend Time Planning
- Do a Proof of Concept Upgrade
- Upgrade – Don't Reimplement
- Upgrade a Copy of Production
- Do Practice Production Runs Until Results are Repeatable
- Get Help From Someone Experienced with Upgrades and Siebel 8.0
- Read and Follow the Siebel Upgrade Guide Carefully.
- Don't Commit During the Wide Part of the Upgrade Cone of Uncertainty
- If Possible Keep Team Size Small

# Estimating Upgrade Projects

- Estimate vs. Target vs. Commitment
- Developer's estimate = best case estimate
- What is the probability of your estimate?
  - Good estimates should provide a range and not a single point.
- Estimates unfortunately often = the minimum amount of time for which you can't prove completing your project is impossible.
- Your last upgrade is your best input to estimate your Siebel 8.0 upgrade.
- Need to do an upgrade assessment and determine upgrade goals and objectives before providing upgrade estimate.
- Counting is always better than estimating.

Adapted from Steve McConnell's Software Estimation – Demystifying the Black Art

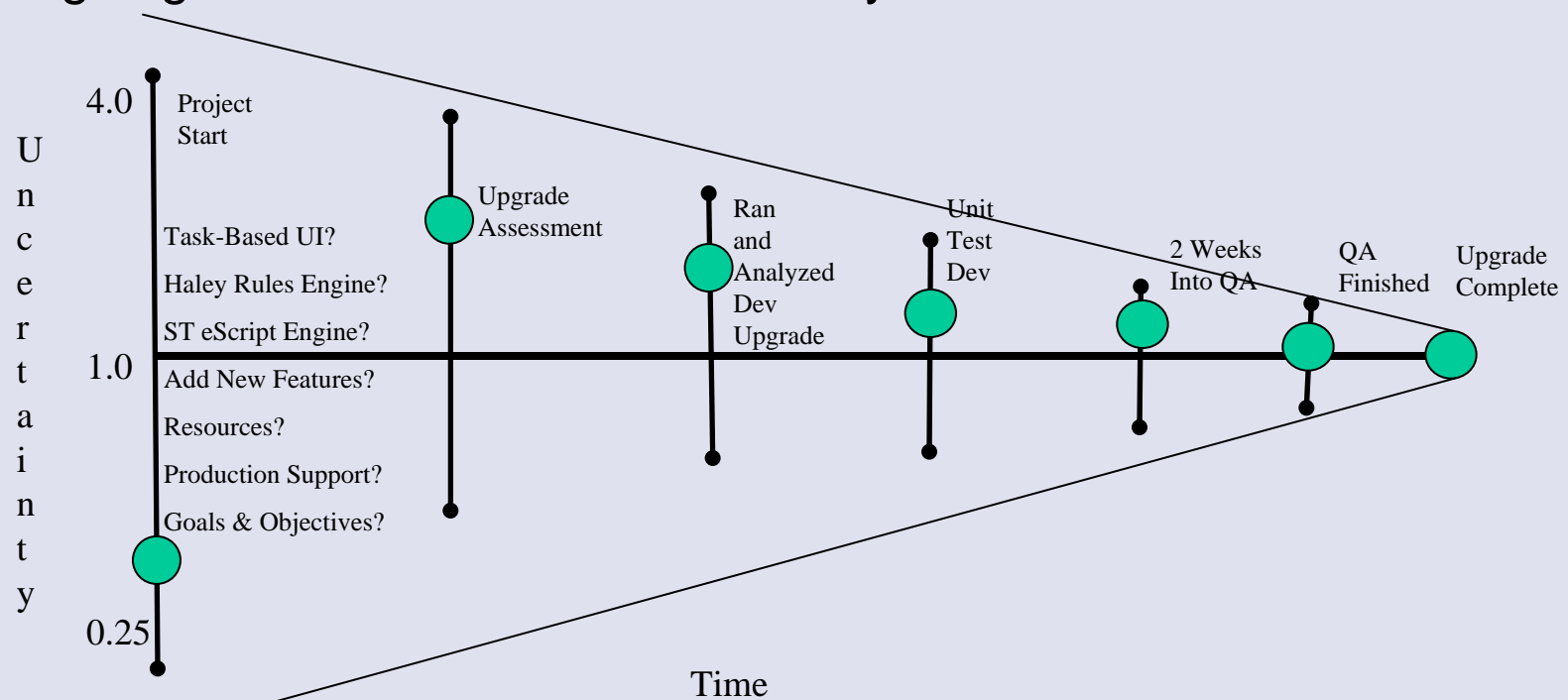
# Estimating Upgrade Projects

- Two main factors:
  - Complexity of upgrade
  - Organizational ability to execute: 10 to 1 differences
- Understanding diseconomies of scale and large upgrade teams.
- Parkinson's Law – overestimated project will expand to fill all allocated time.
- Price of underestimating is even more severe so ideally need accurate estimate
- Counting vs. computing vs. judgment

Adapted from Steve McConnell's Software Estimation – Demystifying the Black Art

# Upgrade Cone of Uncertainty

- The further out we estimate the less accurate our estimate is going to be. Don't commit too early.



Adapted from Steve McConnell's Software Estimation – Demystifying the Black Art

## Converting Counts Into Estimates

- Most upgrade estimates do a good job at counting but don't do a good job converting these counts into accurate estimates.
- 7.0 -> 8.0 upgrade and we counted 200 applets used.
- Not sure what work we are going to have to do but 2 hours per applet "feels" about right. Big mistake. Now we let judgment enter.
- $200 \times 2 \text{ hours} = 400 \text{ hours}$  for applets. Not very accurate but estimates like this are very common on Siebel upgrades.
- Need to know what work is actually going to be required and how long that work will take.
- Best to measure how long work takes and use that observation in your estimate.
- World class performance requires not just measuring how long task takes but also finding the best way to accomplish task.
  - Involves innovating and being effective and not just efficient.

## 7.0 -> 8.0 Applet LOE Example

- Since we did a POC upgrade we know for sure...
- Of the 200 applets 10 of them are extensively modified OOTB applets and these are severely mangled.
- All form applets need to be touched up. Grid applet migration is good but we want to tighten up fields to fit more on form applets.
  - But can do in 5 – 10 mins each with advanced lasso technique.
- List applets look good and don't require any modification for appearance.
- We find out later that some applet scripts need minor work.
- Previous and next buttons missing on some custom applets.

## So How Long Will My Siebel 8.0 Upgrade Take?

- The answer is: it depends.
- Depends mainly on what “league” your upgrade is in.
  - We classify your upgrade’s degree of difficulty by the technical and other challenges you will face and the ability of you and your organization to deal with these challenges.
- Also depends on what your upgrade goals and objectives are and what other tasks beyond a pure technical upgrade are to be accomplished within the upgrade project.
- Safe to say your upgrade can not be accomplished in less than four (4) months unless it is a special simple case.
  - Any estimate less than four months is in the “impossible” zone.

# What League Is Your Upgrade In?

Complexity 

## Special Case

- Simple scenarios that allow upgrade to be performed very fast
- Dev 7.8 to 8.0
- Demo Systems

## Normal

- Normal Siebel implementation without any really complicated challenges
- Hundreds of users
- Weekend of downtime available.

## Extreme Challenges

- Faces one or more really complicated issues
- 6.x with large code body
- Limited downtime
- Very large data volumes
- Organizational issues
- Thousands of users
- Huge impact to business if anything goes wrong
- Etc... There are lots of others

Timeline:

Days or weeks

4+ Months

6-12+ Months

## Two Nearly Identical Upgrades – Two Vastly Different Timelines

Attribute	Upgrade 1	Upgrade 2
Timeline (plan 2 prod)	4 Months	9 Months
Approximate Cost	X	3X
Version	6 -> 7.7	6 -> 7.8
Customization	Highly complex, eScript, interfaces, reports, etc.	Same complexity
Theme	Limited budget and limited time	Upgrade, this is our only chance to fix things, but still save time and \$
Fix existing bugs, return to OOTB and add new features?	No way – are you kidding? Those are change requests for Phase II.	Fixed existing bugs, returned to selected OOTB, added new features

- Both were phenomenally successful but in very different ways.
- What is your upgrade objective going to be???

# What Is The Siebel Upgrade?

## Different Views of the Same Thing

- An exercise in how well you can read and follow the Siebel Upgrade Guide.
- A set of manual and automated steps that:
  - Upgrades your database to Siebel 8.0.
  - Merges your Tools configuration changes with version 8.0 Siebel to produce a customized version of the Siebel 8.0 application.
- A test to see how well you and your team can overcome challenges you may never have faced before.
- An opportunity to impress or disappoint your end users depending on how well you execute the upgrade project.
- A Siebel upgrade is definitely not just a MR patch.
- Different from your original implementation in that you want to skip analysis, design and construction and instead plan, upgrade and fix.

# Siebel Upgrade Overview – Two Main Things Happen

- Upgrades your existing schema to the latest Siebel schema.
  - In place and mostly additive upgrade.
    - New tables, indexes and columns added.
    - Where needed data moved from old tables to new tables.
- Merges your Siebel Tools customizations with the latest Siebel version.
  - Customized repository merged with prior and current Siebel OOTB repositories to produce new customized repository.
  - Only OOTB objects get three-way merged but all objects including totally custom objects can be changed in the upgrade process.
- Note: Siebel upgrade runs in all environments including dev, test and production.

## Siebel Upgrade Steps – Manual and Automated

- A combination of automated and manual steps.
  - The Siebel 8.0 Upgrade Guide lists 78 steps for the development upgrade.
  - Siebel 7.7 Upgrade Guide lists 144 steps for a 6.x development upgrade
- Normally best procedure is to follow the Upgrade Guide to the letter with just a few exceptions.
  - Two-step upgrade which is not documented requires a few changes.
  - There aren't many but there are a few places where the Upgrade Guide is not always correct in all situations.
- Don't invent steps you think the Upgrade Guide might have missed.

# High Level Siebel Upgrade Phases

- Upgrade planning and assessment
- Install new Siebel version and optionally acquire new hardware
- Development upgrade, testing and fixing
- QA upgrade and testing
- Optional upgrade tuning and downtime minimization
- Train users on new system
- Production upgrade and deployment

# Dev Upgrade According to the 8.0 Siebel Database Upgrade Guide

- Check SupportWeb
- Upgrade the Servers
- Upgrade Third Party Software
- Upgrade RDBMS
- Pre-Upgrade Tasks for the Database
- Pre-Upgrade Tasks for DB2 / Oracle / SQLServer
- Pre-Upgrade Tasks for Application Data
- Preparing Developer's for the Upgrade
- **Upgrade the Database Schema (upgrep)**
- Prepare for Repository Merge
- **Repository Merge**
- **Upgrade Custom Database Schema (upgphys)**
- Initialize and Extract Developer's Local Databases
- Review the User Interface
- Postmerge Development Tasks
- Upgrade Tasks for File System and Database
- Postupgrade Tasks for Application Configuration
- System Tests
- Prepare for Transition to Production Test (QA)

## Things You Might Not Have Known About The Siebel Upgrade

- Many Siebel upgrades take longer and cost more than they should.
  - Our goal here today is to educate you so yours won't be one of them.
- System performance is not always better across the board with the new version of Siebel (until you work on it).
- Removing customization and returning to OOTB Siebel can sometimes have a price tag so we want to pick and choose or opportunities to return to OOTB.
- Upgrades often include optional items such as adding new features and functionality that probably should be performed as a separate project.
- Upgrades find and fix a fair amount of issues (25%+) that were present in the current production system but not discovered or not reported until the intensive testing performed as part of the upgrade.
- Even using non scripting alternatives can lead to some work on Siebel upgrades.

## Upgrade or Reimplement?

- Some people may tell you to reimplement instead of upgrade.
- We have yet to see a single case where reimplementing was not a mistake and consider this practice an antipattern.
- Ignores the fact that most customers really do need their customizations and simply can not run Siebel OOTB for all customizations.
- Often customers feel like their original implementation could have been better so they like the idea of being able have a second chance at doing it better than the first time.

## Selectively Going Back To OOTB and New Free Features

- Actual examples where OOTB features replaced customizations:
  - Replaced eScript Audit trail with Siebel audit trail.
  - Replaced complex 35k line eScript assignment manager with Siebel 7.8 Assignment Manager (20% of upgrade budget).
- Compare to features like 7.7+ native browser back/forward buttons, screen homepages, automatic saving of file attachments and others that come for free with the new Siebel version.
- During planning when you look for opportunities to return to OOTB don't be surprised when you can't identify tons of places where you can actually remove your customizations - but you should find some.

## Project Planning: How To Make Your Upgrade Deadline Commitment

- Perform upgrade and get essential functionality to go live accomplished first.
- Put all enhancements on hold until you are sure you have time to fit them in and make sure it is understood that enhancements will only be done if time allows.
- Put opportunities to return to OOTB such as X\_ADDR\_LINE\_2 in the parking lot and only do them if your timeline allows toward the end of the project.
- Put things that have nothing to do with the upgrade (that really should be done anyway) in the same parking lot and only do them if your timeline allows.
- Need a project manager overseeing these decisions since left to their own devices developers will “always” do the last two items even if it means they might jeopardize the project timeline.

## Development Upgrade Expectations

- Application probably won't launch after upgrade due to one or more configuration errors which will have to be fixed.
- Extensively modified OOTB applets will be mangled.
- Screen/view layout will require touchups.
- BusComp links and joins will require fixes.
- SQL errors will be seen until buscomps/tables are straightened out.
- Scripting may require changes even on Siebel 7 -> 8.0 upgrades.
- EIM tables may have new required columns and old IF tables don't exist anymore.
- Integration objects might have different XML schemas until they are touched up.
- Actuate reports will need to be recompiled but custom Actuate VB can require a lot more work.
- Most existing customers don't like 7.7-8.0 removal of page tabs from main screen list views (aggregate view require drilldowns to see tabs).

## Other Tasks To Consider

- Implementing new modules like 7.8 Order Management can be a very large project in itself.
- Migration from Siebel 6.x or SEA to SIA requires two steps.
- Pick the right version. 7.8 or 8.0?
- Migrate to new ST eScript engine or stay with old T engine?
- Migrate database to UNICODE?
- Migrate to UTC date / times?
- Change database vendors?

## How To Prepare For a Successful Siebel 8.0 Upgrade

- Determine upgrade goals and upgrade strategy.
- Read and understand the Siebel Upgrade Guide and supporting information from Support Web.
- Be sure to check System Requirements and Supported Platforms.
- Do at least one Siebel 8.0 dev upgrade so that you will understand what you are doing and have some experience doing it.
- Engage someone who knows the Siebel upgrade and knows the version of Siebel you are upgrading to.
- Do careful and detailed planning as part of your upgrade assessment to develop roadmap and staffing plan for upgrade project.

# Following Your Roadmap

Adobe Acrobat Standard - [Siebel Database Upgrade Guide]

File Edit View Document Tools Advanced Window Help

Open Save Print Email Search Create PDF Review & Comment Secure Sign

Select Text 134%

Options x

Bookmarks

Signatures

Layers

Pages

Comments

## Upgrade Siebel Database Schema (upgrep)

27 ["Renaming the Siebel Repository" on page 199.](#)

28 (Optional) ["Changing the Siebel Database Configuration Utilities Language" on page 200.](#)

29 Run the Database Configuration Wizard:

- ["Preparing to Run the Siebel Database Configuration Wizard" on page 201.](#)
- ["Running the Siebel Database Configuration Wizard Under Windows" on page 205.](#)
- ["Running the Siebel Database Configuration Wizard Under UNIX" on page 207.](#)

Choose the following settings:

- **Upgrade Options:** Upgrade Siebel database Schema (upgrep)
- **Environment Type:** Development

30 Edit generated SQL files as required by Siebel Technical Notes, Alerts, Release Notes or other publications. For a complete list, see the Installation and Upgrade Portal page on Siebel SupportWeb.

31 Resume the upgrade: ["Starting the Siebel Upgrade Wizard" on page 210.](#)

32 Review the upgrade logs and resolve errors:

- ["Summarizing Siebel Log Files Using the Logparse Utility" on page 225.](#)

8.49 x 11 in

27 of 352

# Upgrade Steps

- Nearly 80 individual steps listed for a 8.0 dev upgrade. Fewer for production.
- Many mostly manual steps leading up to running the steps below.
- Database Configuration Wizard
  - Upgrep
  - Upgphys
- Siebel Tools
  - Repository Merge
  - Post Merge Utilities
  - Generate EIM Processing Columns
- More mostly manual steps to fix and test the upgraded application.

## What SEA 7.5 -> 8.0 Dev Upgprep Does

- Issues SQL both DDL and DML (18 odbcsql steps and 16 dbsql steps).
- Modifies schema definition by comparing existing physical schema to definition in ddlctl and ddlsptblctl (8 ddlimp steps).
  - Downstream upgrades will use schema.dll from dev.
- Imports new Siebel seed data such as LOVs and views (4 dataimp steps).
- Imports Siebel repositories for the merge process (3 repimpexp steps).

Siebel Configuration Wizard - Database Configuration Wizard

Please give the network hostname of the Siebel Gateway Name Server. If the Gateway is configured to listen on a port other than 2320, specify the port in the format "<host>:<port>".

Gateway Name Server Address  
 RPONDERM70

Enterprise Server Name  
 sea80

ORACLE

InstallShield \_\_\_\_\_

< Back   Next >   Cancel

Siebel Configuration Wizard - Database Configuration Wizard

Install Database  
 Upgrade Database  
 Apply Additive Schema Changes  
 Import/Export Repository

ORACLE

InstallShield \_\_\_\_\_

< Back   Next >   Cancel

Siebel Configuration Wizard - Database Configuration Wizard

Please choose the type of environment you are upgrading:

Development  
 Production

ORACLE

InstallShield \_\_\_\_\_

< Back   Next >   Cancel

Siebel Configuration Wizard - Database Configuration Wizard

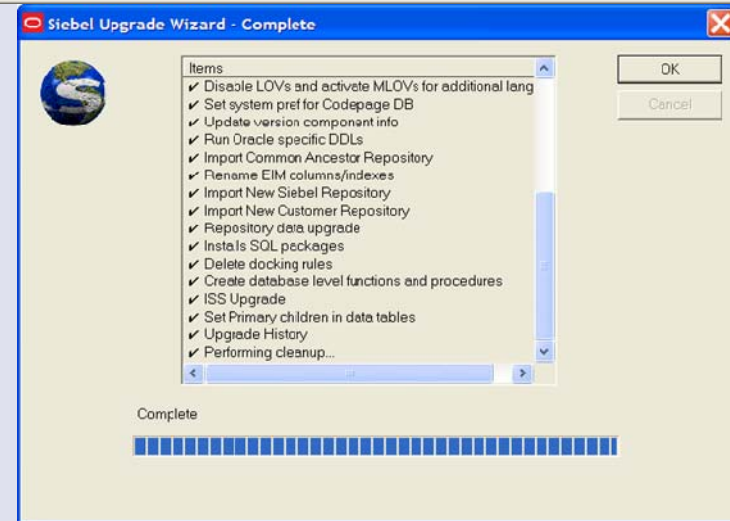
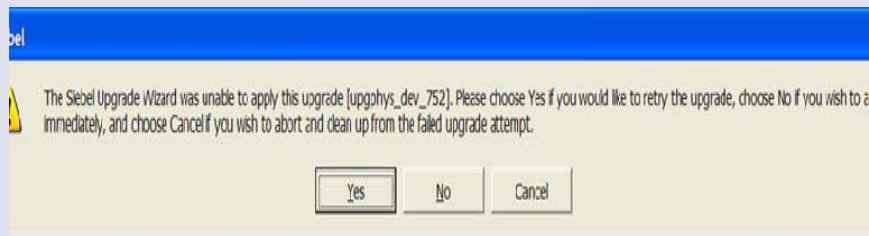
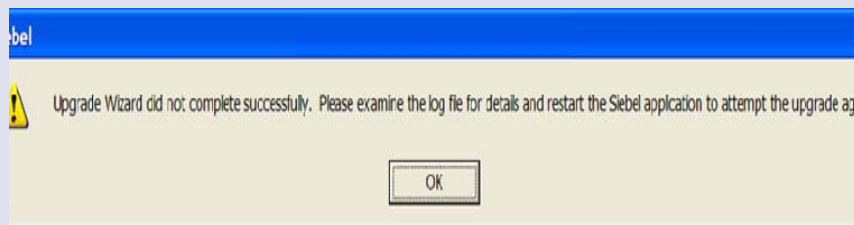
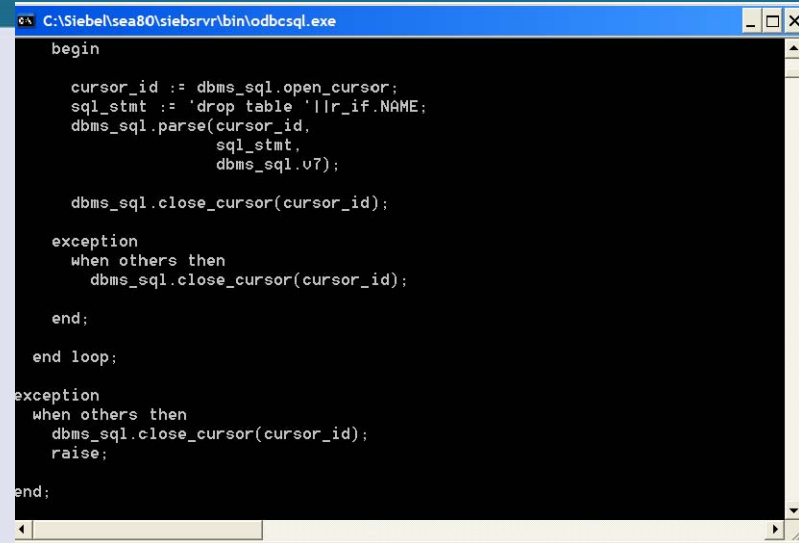
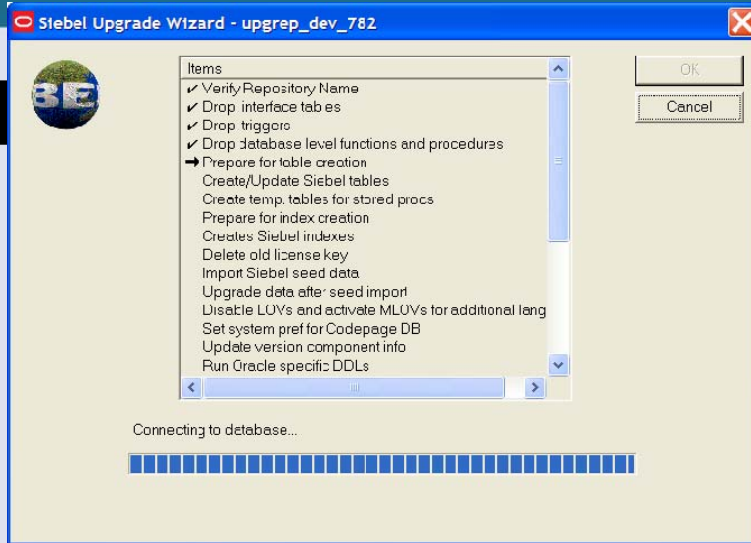
Indicates the version from which you are upgrading

v7\_5\_2  
 v7\_5\_2\_200+  
 v7\_5\_3  
 v7\_7

ORACLE

InstallShield \_\_\_\_\_

< Back   Next >   Cancel



Steps/Errors Summary for process upgrep\_dev\_752 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

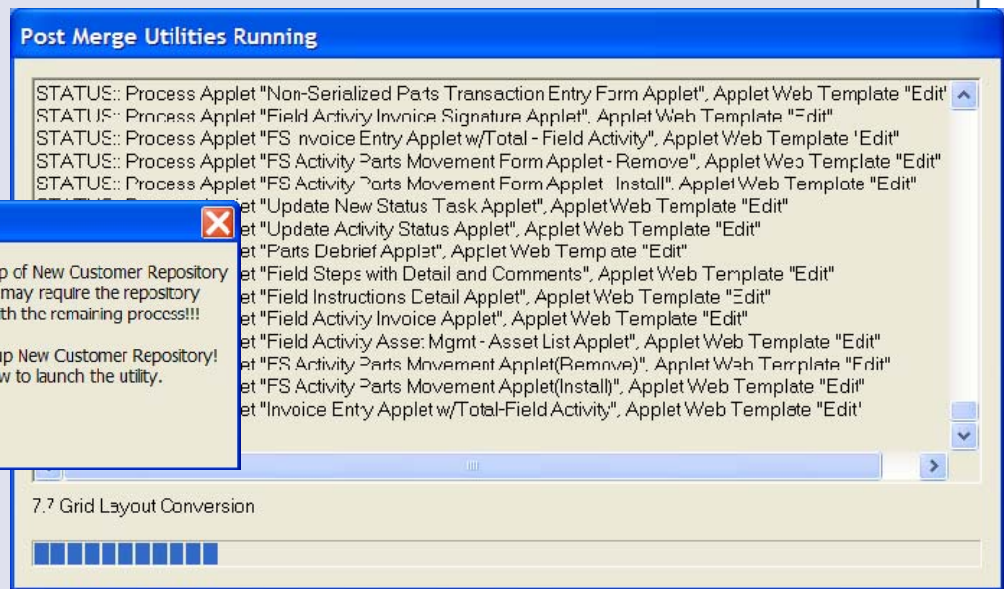
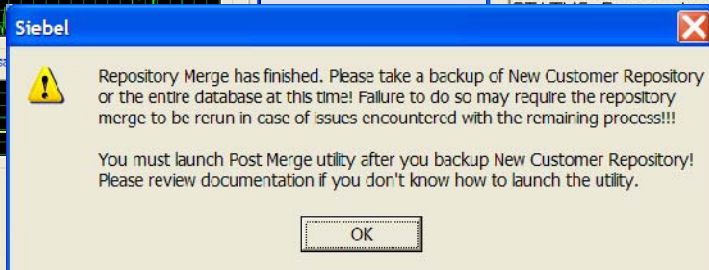
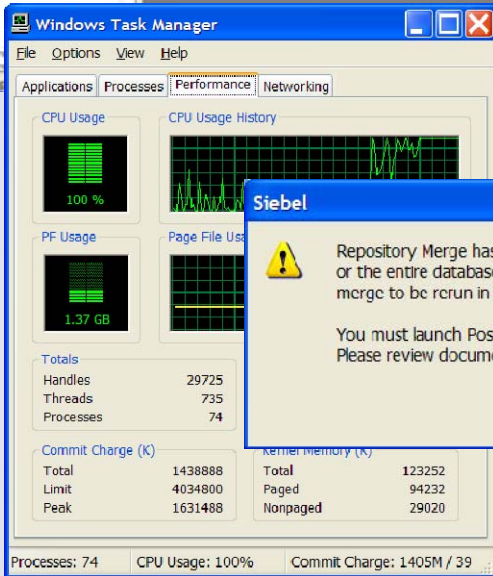
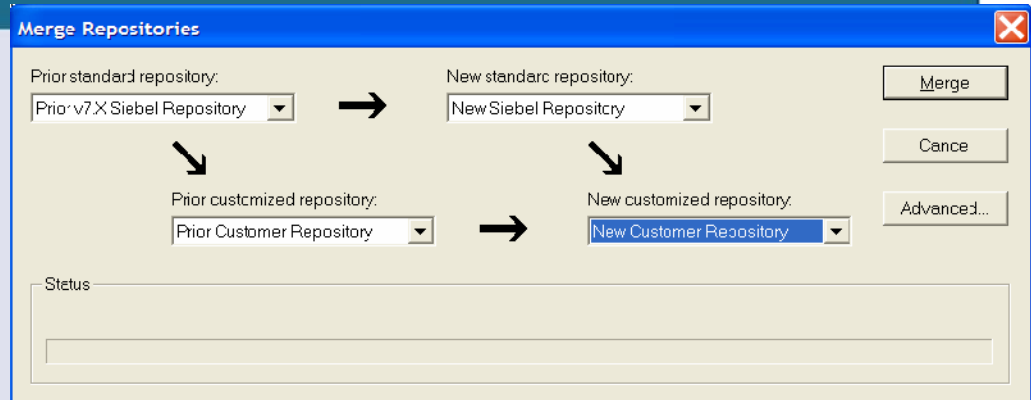
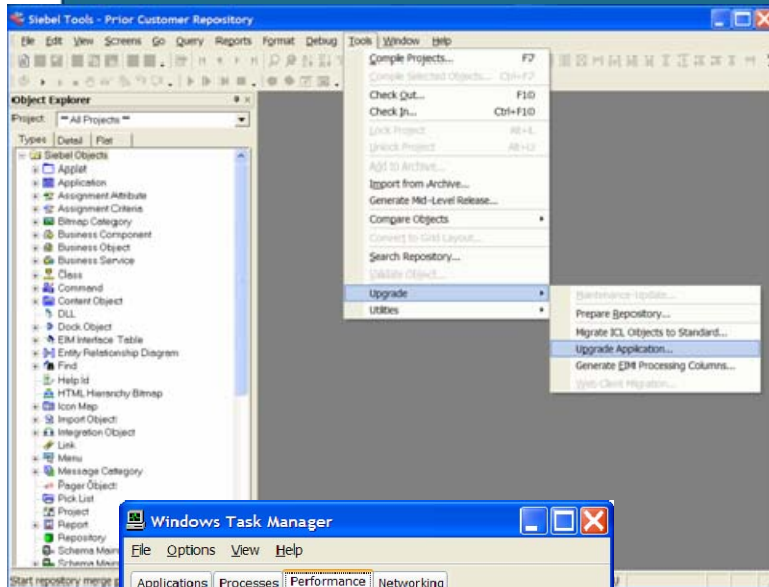
Address C:\Siebel\sea80\siebsrvr\LogSave\upgrep\_dev\_752\summary\upgrep\_dev\_752-steps.html

Step	Name	Start Time	End Time	Net Cost	Parallel	Interrupted	Status
0	Verify Repository Name	2007-02-17 17:03:32	2007-02-17 17:03:33	00h:00m:01s			Complete
1	Drop interface tables	2007-02-17 17:03:33	2007-02-17 17:04:40	00h:01m:07s			Complete
2	Drop triggers	2007-02-17 17:04:40	2007-02-17 17:04:43	00h:00m:03s			Complete
3	Drop database level functions and procedures	2007-02-17 17:04:43	2007-02-17 17:04:43	00h:00m:00s			Complete
4	Prepare for table creation	2007-02-17 17:04:44	2007-02-17 17:06:22	00h:01m:38s			Complete
5	Create/Update Siebel tables	2007-02-17 18:12:56	2007-02-17 18:17:57	00h:06m:39s		<input checked="" type="checkbox"/>	Complete
7	Create temp. tables for stored procs	2007-02-17 18:17:57	2007-02-17 18:19:39	00h:01m:42s			Complete
9	Prepare for index creation	2007-02-17 18:19:39	2007-02-17 18:21:04	00h:01m:25s			Complete
10	Creates Siebel indexes	2007-02-17 18:39:34	2007-02-17 18:43:44	00h:05m:35s		<input checked="" type="checkbox"/>	Complete
12		2007-02-17 18:43:45	2007-02-17 18:46:53	00h:03m:08s			Complete
14	Delete old license key	2007-02-17 18:46:53	2007-02-17 18:46:55	00h:00m:02s			Complete
15	Import Siebel seed data	2007-02-17 18:46:55	2007-02-17 18:51:17	00h:04m:24s	<input checked="" type="checkbox"/>		Complete
16		2007-02-17 18:46:55	2007-02-17 18:49:10	00h:02m:15s	<input checked="" type="checkbox"/>		Complete
17		2007-02-17 18:51:17	2007-02-17 18:52:09	00h:00m:52s			Complete
18		2007-02-17 18:52:09	2007-02-17 18:52:09	00h:00m:00s			Complete
19	Upgrade data after seed import	2007-02-17 18:52:09	2007-02-17 18:52:12	00h:00m:03s			Complete
20	Disable LOVs and activate MLOVs for additional languages	2007-02-17 18:52:12	2007-02-17 18:52:13	00h:00m:01s			Complete
21	Set system pref for Codepage DB	2007-02-17 18:52:13	2007-02-17 18:52:14	00h:00m:01s			Complete
23	Update version component info	2007-02-17 18:52:14	2007-02-17 18:52:15	00h:00m:01s			Complete
24	Run Oracle specific DDLs	2007-02-17 18:52:15	2007-02-17 18:52:30	00h:00m:15s			Complete
25	Import Common Ancestor Repository	2007-02-17 18:52:31	2007-02-17 19:05:40	00h:13m:09s			Complete
26	Rename EIM columns/indexes	2007-02-17 19:05:40	2007-02-17 19:06:48	00h:01m:08s			Complete
27		2007-02-17 19:06:48	2007-02-17 19:06:49	00h:00m:01s			Complete
28		2007-02-17 19:06:49	2007-02-17 19:06:49	00h:00m:00s			Complete
29	Import New Siebel Repository	2007-02-17 19:06:49	2007-02-17 20:49:53	01h:43m:04s			Complete

Done My Computer

# Tools Repository Merge

- Repository Merge
  - Three-way merge to produce New Customer Repository in Siebel 8.0.
- Post Merge Utilities
  - Does everything else required to migrate application to 8.0 such as screen/view navigation changes.
- Generate EIM Processing Columns



# Understanding Merge Conflicts

- Three-way merge compared and found all three values are different.
- If just custom is different than OOTB there is no conflict. Custom is used.
- On conflict either custom wins or standard (OOTB) wins. This varies based on the attribute and can not be changed.
- Conflict resolution gives you the ability to review and override conflicts without having to check each object out and manually make these changes.
- Normal to have a few hundred merge conflicts. We know one Siebel customer who had a few thousand merge conflicts.

# Processing Merge Conflicts

The screenshot shows the Siebel Tools interface with the 'Attribute List' window open. The 'Object Explorer' on the left lists various Siebel objects. The main window displays 'Application Upgrades' and 'Attribute Differences' tables.

**Application Upgrades Table:**

User	Operation	Status	Prior Standard Repository	Prior Customized Repository	New Standard Repository	New Customized Repository
SADMIN	Merge(NonInCL)	Complete - 1 erro	Prior v7.X Siebel Repository	Prior Customer Repository	New Siebel Repository	New Customer Repository

**Attribute Differences Table:**

Conflict	Overri	Object Name	Top Parent Name	Top Par	Object	Attribute	Resolution	Prior Standard	New Standard	Prior
✓		Account Detail - Busir	Account Detail - Business	View	View 'Account Detail - Busine	Thread Api	Use Standard Value	Account Form App	Account Entry Apple	NF All
✓		Account Note View	Account Note View	View	View 'Account Note View'	Thread Api	Use Standard Value	Account Form App	Account Entry Apple	NF All
✓		Account Organization	Account Organization Ar	View	View 'Account Organization #	Thread Api	Use Standard Value	Account Form App	Account Entry Apple	NF All

Item: 1 of 3 | Language:ENU

# Merge Conflict Examples

- Object: Field, Attribute: Column, Name: Address Line 2, Resolution: Use Standard
  - PS = null, PC = X\_ADDR\_LINE\_2, NS = ADDR\_LINE\_2
- Object: BusComp: Attribute: Sort Spec, Name = Action ToDo, Resolution: Use Custom
  - PS = Display, Due, PC = Planned, Type, NS = Display, Planned
- Object: (User Prop) BC Read Only Field: Attribute = Value, Resolution: Use Standard Value
  - PS = Null, PC = Am I Owner, NS = NoEdit

# Be Smart with Merge Conflicts

- Group conflicts and process each group instead of processing conflicts individually. If not this could take days or weeks to do!!!

```
select top_parent_type, obj_type, attribute, resolution, count(*)
from siebel.s_merge_log,
     siebel.s_merge_log_attr,
     siebel.s_merge_log_obj
where attribute not in ('Comments')
and s_merge_log.row_id = s_merge_log_obj.merge_log_id
and s_merge_log.row_id = '1-7FVO9'
and merge_obj_log_id = s_merge_log_obj.row_id
and conflict_flg = 'Y'
group by top_parent_type, obj_type, attribute, resolution
```

- Normally will accept or override the entire group the same way.
- Carefully document what you did including anything you could not resolve that will require further research or Tools changes later.
- OK to skip a few of these if you are not sure since we can always go back and fix them by checking the object out.
- Don't let any developers in until merge conflicts have been resolved since these changes happen directly to the server copy of the repository and locking is not required.
- With someone who knows the Siebel Object Types Reference and someone who knows the application you should be able to complete this step in about 4 hours (not 2 weeks).

# Well Documented Conflict Resolution

TOP_PARENT_TYPE	OBJ_TYPE	ATTRIBUTE	RESOLUTION	COUNT(*)	Override?
Applet	Applet	Popup Dimension	Use Standard Value	5	Override
Applet	Applet Locale	Redo	Use Standard Value	2	OK
Applet	Applet Web Template	Web Template	Use Standard Value	2	OK
Applet	Applet Web Template Item	Inactive	Use Standard Value	2	OK
Applet	Applet Web Template Item	Item Identifier	Use Custom Value	2	OK
Applet	Control	Sequence	Use Custom Value	2	OK
Applet	Control	Text Alignment-Label	Use Standard Value	4	OK
Applet	Control Locale	Caption	Use Custom Value	3	OK
Applet	Control Locale	Prompt Text	Use Custom Value	19	OK
Applet	Control Locale	Redo	Use Standard Value	242	Accept. We are not ml so this feature w
Applet	List Column	HTML Type	Use Standard Value	2	OK
Applet	List Column	Pick Applet	Use Standard Value	1	Override
Applet	List Column	Read Only	Use Standard Value	1	Override
Applet	List Column	Sequence	Use Custom Value	97	OK
Applet	List Column	Width	Use Custom Value	5	OK
Application	Application Find	Inactive	Use Standard Value	1	Override
Application	Application Find	Sequence	Use Custom Value	1	OK
Application	Page Tab	Sequence	Use Custom Value	7	OK
Application	Screen Menu Item	Sequence	Use Custom Value	4	OK
Business Component	Business Component	Sort Specification	Use Custom Value	2	OK
Business Component	Business Component User Prop	Inactive	Use Standard Value	2	Override
Business Component	Business Component User Prop	Value	Use Standard Value	8	Override but note which ones and fix
Business Component	Field	Calculated	Use Standard Value	2	Override
Business Component	Field	Calculated Value	Use Standard Value	4	Override
Business Component	Field	Column	Use Standard Value	1	Override - but consider moving in PUP
Business Component	Field	Currency Code Field	Use Standard Value	1	Override - but look into new field further
Business Component	Field	Force Case	Use Standard Value	1	Override
Business Component	Field	Link Specification	Use Standard Value	3	Override
Business Component	Field	Multi Valued	Use Standard Value	1	Override
Business Component	Field	No Copy	Use Standard Value	3	Partial Override - Leave Row Id as no c

# Record Count Balance Sheet

- We advise customers to verify record counts in all tables before and after the upgrade.

Microsoft Excel - Record Count Balancing For Test Upgrade 2006 05 26.xls

Type a question for help

Reply with Changes... End Review...

Table Name

A1	A	B	C	D	E	F	G	H	I	J	K
Table Name	6.0 Records	Existed In 6.0?	Upgrade Inserts	Upgrade Deletes	7.8 Records	Difference	Status	SQL Reference	Notes		
S_ORD_CRDT_ASGN	0	No	5,963,845	13	5,963,832	0		RUN_SQL_1610 - dedup4.log			
S_ORDER_POSTN	5,963,845			13	5,963,832	0		dedup1.log			
S_DOC_QUOTE_XM	4,318,054			8	4,318,046	0		dedup2.log			
S_ESCL_LOG	2,218,569				2,218,569	0					
S_ACCNT_POSTN	1,520,489		351		1,520,840	0		RUN_SQL_223			
S_POSTN_CON	1,462,931			2	1,463,577	-648		seedupg0.log			
S_OPTY_POSTN	1,183,129				1,183,119	10					
S_ORDER_ITEM	1,065,920				1,065,920	0					
S_APSRVR_REQ	1,022,646				1,022,646	0					
S_ORDER_XM	746,501			359	746,142	0		dedup3.log			
S_QUOTE_ITEM	483,493				483,493	0					
S_ORDER	337,762				337,762	0					
S_ORDER_BU	0	No	337,762		337,762	0		RUN_SQL_320			
S_REVN	0	No	254,804		254,804	0		RUN_SQL_590 + RUN_SQL_600 + RUN_SQL_610			
S_ORG_TYPE	210,128			11	210,117	0		dedup3.log			
S_DOC_QUOTE	205,729				205,729	0					
S_DOC_QUOTE_BU	0	No	205,729		205,729	0		RUN_SQL_290			
S_QUOTE_POSTN	0	No	170,454		205,714	-35,260		RUN_SQL_120			
S_DOC_ORDER	201,503				201,503	0					
S_OPTY_PROD	173,968				173,968	0					
S_PARTY	0	No	171,405		171,399	6		RUN_SQL_2430 + RUN_SQL_2440 + RUN_SQL_2450			
S_ACT_EMP	613				130,819	-130,206					

Balalance Sheet / Pre Upgrade Records / Special Calculations

# How To Not Error Out When Running The Upgrade

- Would like to avoid errors when running the upgrade.
- If we follow the Upgrade Guide and set all DBMS parameters correctly and perform all steps correctly this will help.
- Still possible and pretty likely that we will get an error or two especially when running dev upgrade or post dev upgrade for the first time.
- “OK” to get errors on upgrep and upgphys since we can restart these utilities after we diagnose and fix the problem.
  - This will probably happen to you at least once.
  - Not OK for actual production upgrade.
    - That is why we want to practice until perfect.
- If merge crashes we can’t just restart but instead have to restore DBMS and rerun.
  - This probably won’t happen to you but you might get !!Error in your merge.log but that is different.

## Some Very Common Errors We See When Running the Upgrade:

- Failure to allocate enough space for DBMS or RBS/UTS growth/extents.
- DBMS memory issues due to parameters not being set correctly.
- Duplicate indexes between OOTB and custom indexes.
- Conflicts between column attributes such as not null defined in Tools vs. actual database schema.
- Users canceling the upgrade because they think it is stuck/hung.
- Users entering parameters incorrectly (we like to copy and paste from our install notes).

# Planning Best Practices

- Careful planning will be required for your success.
- Do a trial or real upgrade early in the planning processes to get trained and help with your LOE estimation.
- Get the help of someone experienced with upgrades.
- Don't bring a large team of consultants in until you figure out what needs to be done and actually need them.
- Ramp consultants up slowly since everyone will bill 40 hours even if they are not doing much.
- Produce three deliverables: roadmap, staffing plan and project timeline.
- Get granular – e.g. no task > 40 hours.
- On staffing plan try not to equate 1 skill = 1 person. Look for people with multiple skills. Can drastically reduce project costs and timeline.
- Remember software projects including upgrades expand to fill all available time. 201 Principles of Software Development by Alan M. Davis.
  - Dammed if you underestimate and dammed if overestimate as well.
- If you estimate with formulas like 100 applets \* 3 hours each = 300 hours then your estimate is not very good.
- Good, better and best time to fix something. Applet alignment example.
- Try to be innovative and effective instead of just efficient. Slack by Tom DeMarco. Mismatched field labels and controls example.
- Don't do too many things at the same time. Just do Siebel upgrade as a project.
  - Opposite of Siebel at Siebel upgrade presentation at OOW 2006

# Tools Merge Best Practices

- Be sure to carefully follow all Upgrade Guide settings for performance.
- Tools performance is important.
  - Single CPU/core will be used, network very important, memory important too.
    - Don't try this at home. Remote access over home high speed internet is tool slow unless you are using MSTC or similar.
  - If Windows app server consider installing Tools here.
- Delete old repositories especially if there are a lot of them.
- Look at conflicts as a group instead of one at a time.
  - See me for this SQL and example of how to apply this technique.
- We don't like Incorporate Custom Layout (ICL) or Upgrade Ancestors and don't use them.
- We don't like the Return to Standard step required for post ICL upgrades. Customers very upset when they find out about it.
- Check `\tools\bin\MERGE0.TXT` for errors.
- Check `\tools\reppatch\reputility.log` for issues that represent work you will need to do.

# Dev Upgrade Best Practices

- Only let 1 or 2 developers in until the application launches and the main bugs are fixed.
- After you compile use `SIEBEL_LOG_EVENTS = 4` or `5` to find and fix errors that prevent application from launching.
- Don't put your srf on the server until it works.
- Find and fix all level 1 errors. Use Perl to scan logs for these errors.
- Start mining the OM logs for other goodies and watching for FDR files.
- Don't delete your old customized repository until upgrade project is finished.
- Look for ways to automate repetitive tasks.
- Automated daily build and srf and browser script push to server. Twice daily for offshore model.
- Automated daily repository export with 30 days of history kept just in case.
- Single project checkout on nearly everything.
- Easier to start preliminary testing in dev until things get stable since migrations to QA take time.
- Get handle on configuration management (CM) early.
- Often best to divide Tools work by area (Activities, Contacts) instead of by task (UI, buscomps, scripts) if team is talented enough. Exception would be things like EIM, actuate, AM, WF policies, etc. that should be treated as specialties.

## QA Upgrade Best Practices

- Current copy of production data would be ideal for Production Test sequence.
- Start perfecting CM migration from dev early.
- Continue monitoring OM logs for performance, level 1 errors, etc.
- Consider development shakedowns before migrating new builds.
- Time to start building your scripts here if you did not already start in dev. E.g. scripts to restart upgrade, parse logs, verify DBMS parms, etc.
- If you took the time in dev to script configuring your Siebel components then you are going to save a lot of time here and in prod. If not then do it now.
- Can install and upgrade QA way before dev is finalized. We are going to be doing several dev2prods anyway so we can get started on this environment early.
- May want to consider using new prod as QA and then build real QA after go-live. Can make scheduling practice production runs difficult if QA and practice runs need same environment.

# Production Upgrade Best Practices

- Perform several practice runs exactly the same way as the real production upgrade will be performed.
- Don't change things from the way you did them in dev and test.
- Should have detailed document that lists exact steps, who will do each, duration, etc.
  - Sometimes have both a .mpp and .doc/.xls.
  - Need to identify not only step but also any file script names including .sql or .bat files.
  - Need to also determine how each step is to be QA'ed or assured to be accurate.
- Install new infrastructure hardware and upgrade a copy of production DBMS and leave old Siebel version available on standby.
- Migrate and test your CM tasks such as SRF migration, Siebel server configuration well ahead of the go-live weekend. That will leave just the upgrade itself and post upgrade database CM migrations like List of Values to be performed on go-live weekend.
  - Question: What is one installation task that will have to be redone if database is restored and upgrade run again?
- Think trough shifts and who is going to work when since we can't all go for 48 hours with zero sleep.
- Remember that we can't declare victory after upgphys has run. We have post upgrade manual task and many hours of testing.
- Produce before and after record counts and balance your differences with information in the Siebel logs.
- Produce "hash totals" from old and new databases and make sure these balance too. E.g. Sum all order amounts from old and new Siebel. Developing these scripts will take time.

# Upgrep Best Practices

- Allocate enough disk space for growth and be sure rollback/undo is large. The better your DBA's the worse the problems will be here.
- Don't pick parallel in dev as mainly empty EIM indexes get built.
- Primarily uses DBMS resources but network connectivity is also important.
- Monitor closely including 10G long ops using TOAD or V\$ tables, etc.
- Use logparser after upgrade has run.
  - Use /t parameter in dev to get a hint of what might run long in downstream environments.
  - Don't forget that when lots of errors occur they don't show up in the output but instead hide and are indicated as:
    - Errors : 850 errors found
- Make sure Siebel servers don't go down while running and that services don't start on accident.
  - Set to manual until upgrade is finished.

# Upgrade Tuning Best Practices

- Index builds require sorts and sorts run faster when performed in memory so large PGA / sort\_area\_size will help here.
- For upgrade tuning selectively turn parallel indexing off for tables in the EIM and small table swim lane.
- Upgrade tuning normally benefits from /\*+ append \*/ hint on inserts and /\*+ parallel \*/ hint on updates.
- Biggest improvements are often eliminating certain statements and these don't always show up as zero row statements.
  - E.g. Updated 50M rows but set an all null column equal to null for all rows.
- CTAS and index rebuilds often much faster than updates on huge tables.
- Be sure parallel\_max\_servers is set high enough.
  - See <http://www.ponderproserve.com/ParallelProcessingSiebelUpgrade.html> for details.
- Siebel 8.0 has additive schema wizard to allow schema changes to be made ahead of production go-live weekend.
- If you require ES upgrade tuning don't waste time with the Siebel Upgrade Tuner. Your changes will have to be discarded.
  - Note: Upgrade Tuner only runs on Windows OS.

# Questions and Answers

rponder at ponderproserve.com  
770.490.2767

<http://www.ponderproserve.com/support.html>

# Upgrade Planning

- Minimum of four weeks of planning recommended.
- Don't just assess what you have. Need to determine actual work required to move to new version.
- Ideally do the real or trial upgrade early in the planning process.
  - Provides JIT hands-on upgrade training.
  - Identifies exactly what will happen and what will break after the real upgrade runs.
  - If real dev upgrade done during planning that much less time required in next phase.
- Want to produce these deliverables:
  - Identify all work tasks required for upgrade project.
  - Task durations and dependencies.
  - Staffing plan for resource types and counts.

# Development Upgrade

- Potentially very large on 6.x upgrades.
- For 7.x upgrades normally can be measured in weeks.
- Tasks vary depending on a number of factors including current Siebel version, amounts and type of Tools configuration, specific modules, etc.
- Certain parts of your Siebel configuration will “break” after the upgrade and we need to determine what these items are and how we are going to fix them.
- Need to decide which new features to take advantage of.
- Determine which parts of implementation should return to OOTB and/or be refactored.

## QA Testing Time

- Largest component on 7.x -> 8.0 upgrades.
- Duration frequently underestimated.
  - Don't use last point release as a basis. Instead use testing time from last upgrade or initial implementation.
- Plan to test everything. Siebel UI, interfaces, reports, etc.
- Be sure to add time for performance and scalability testing of Siebel application and infrastructure.
  - Expect to add time to fix performance issues since better performance in all places won't happen – some areas will be slower until tuned/fixed.
- For large databases upgrade tuning can take a very long time when very limited downtime must be achieved.

## Production Upgrade and Rollout

- Allow time for several practice runs before doing the real thing.
- End user training and change management can be a very large task depending on number of users.
- Ideally would like to have some type of phased rollout but in practice this can be difficult and very costly to achieve.
- Always leave old Siebel instance and database alone and upgrade a copy of production in order to have fallback. Also allows practice runs of real upgrade on actual hardware. Can use new prod as QA as well until go-live.
  - Weekend upgrade does not allow time to restore databases and application to old version if it was uninstalled.