



# Planning for a Successful Siebel 7.8 / 8.0 Upgrade

---

Robert Ponder  
Ponder Pro Serve  
rponder at ponderproserve.com  
770.490.2767



# About Me – Robert Ponder

---

- Joined Siebel in 1998.
- Leading speaker on Siebel upgrades while at Siebel.
- World-wide PS performance and scalability lead for Siebel red account team.
- Ran first Siebel upgrade in 2000.
- Assisted/lead numerous upgrades for Siebel.
- Left Siebel in November 2005 to work for my wife!
- Currently working on Siebel upgrades and building cool Siebel utilities such as PPS Tools Helper© and PPS Transaction Profiler©.
- Lead Upgrade Architect and Director of Upgrade Services at Ponder Pro Serve.
- Upgrade Automation Analyst interested in improving the speed and accuracy of the Siebel upgrade by replacing manual tasks with automation.



# Agenda

---

- Understanding the Siebel upgrade
- Siebel upgrade tasks
- Planning your Siebel upgrade
- Live pseudo demo running the Siebel upgrade
- How long will my upgrade take?
- PPS upgrade tips
- Q&A



# Why Upgrade Siebel?

- Additional Siebel features and functionality.
  - Including better performance and scalability than 7.0 and 7.5.
  - Siebel 8.0 features like task-based UI and fix and go eScript.
- Support issues with older Siebel versions such as Siebel 2000/6.x.
- Incorporate latest technology and improve ROI/TCO.
- Better usability and customer satisfaction.
- Standardize Siebel version across divisions.
- And lots of other reasons...
  - What are some of yours???



# What Is Your Siebel Philosophy?

## ■ Siebel Puritan

- OOTB is always better than customized and OOTB should always be the goal regardless of time/cost.
- As new Siebel versions are released customers will be able to remove more and more of their customizations until one day when everyone will be close to OOTB.
- Customers don't really want their customizations – they want OOTB Siebel.

## ■ Siebel Realist

- Customers need/needed their customizations since there was/is a gap between Siebel OOTB and requirements.
- Customers want to get thru the upgrade process without having to spend a ton of time and money.
- Removing customizations and scripting is good but we have to pick and choose where we are going to return to OOTB.



# 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.
- Originated when inexperienced teams upgraded from 6 to 7 and the upgrade was painful so some generalized this into believing that the Siebel upgrade is best avoided and this is simply not correct.
- Ignores the fact that most customers really do need their customizations and simply can not run Siebel OOTB.
- 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.
- Favored by some system integrators because reimplementing means there will be analysis and design phases which are normally minimized/skipped on upgrade projects.
  - We know of cases where system integrators have told Siebel customers that the upgrade would not work and they had no choice except to reimplement.



# Reimplementation Example

- Customer was advised to use upgrade process to migrate data to new schema but throw away all Tools changes and start over. They did!
- Upgrade project required  $1 + 2 = 3$  years!
- Went live with 7.5 (not 7.7) in late 2005 on obsolete hardware.
- 70/30 customization before and 60/40 customization after.



# Upgrade Example - Demo System

- Demo system said to be minimally configured with just a little sample product data in database.
- We thought we had found our first case where we would be better off installing a new instance instead of upgrading and reentering the sample data.
- Began researching on Monday and by Tuesday realized there were weeks of customizations and lots of custom data.
  - Point: We tend to forget about all the hard work put into configuring Siebel.
- Two-step upgrade from SEA 7.5 to SIA 7.7 started Tuesday night and we had a mostly working application by Friday.
  - Results not typical – very strong 3 person team and remember this was just a demo system with no live production users.
- After about 20 hours of fixes over the following weeks the system was working fully and Siebel to SAP integration could be demonstrated using UAN and middleware.
  - Some of the 20 hours were spent adding enhancements having nothing to do with the upgrade.



# Reimplementation/Upgrade Flowchart

---

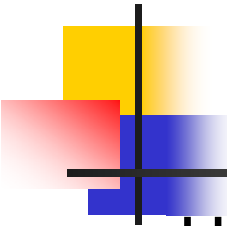
- Do you need the data in your Siebel database?
  - If yes then let the Siebel upgrade migrate your data to the new schema.
    - Using EIM instead of upgrade scripts to move data is also considered an antipattern based on past customer experiences.
- Do you need the new data elements you added to the Siebel schema such as extension tables and columns?
  - If yes then let the Siebel upgrade (Tools merge) migrate your schema extensions to the new Siebel version.
- Do you need these data elements shown on the Siebel UI?
  - If yes then use the Siebel upgrade (Tools merge) to migrate customized tools objects such as applets and buscomps to the new Siebel version.
- Do you need all of the custom eScript code you have written?
  - If no then delete all scripts after the upgrade. Are you kidding? We need that business logic!
  - If yes then keep custom code.
    - Can look for non-scripting options to replace code but beware of the cost and time implications of doing so.
    - Elimination of custom code is good but sometimes this can be difficult and expensive.



# Going Back To OOTB vs. New Free Features

---

- Going back to OOTB can carry a high price tag so be careful when you say you want to go back to OOTB as part of your upgrade. We have to pick and choose wisely when to return to OOTB.
- Actual examples where OOTB features replaced customizations:
  - Replaced eScript Audit trail with Siebel audit trail for about \$10K.
  - Replaced complex eScript assignment manager with Siebel 7.8 Assignment Manager for about \$175K (20% of upgrade budget).
- Compare to features like 7.8/8.0 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 many places where you can actually remove your customizations.
- E.g. what is the cost/benefit of replacing eScript business rules with the 7.8 business rules engine or the new 8.0 Haley rules engine?
  - What benefit would your business and end users see?
  - What if the rules changed a lot vs. were static?
- E.g. You added X\_ADDR\_LINE\_2 and now Siebel has ADDR\_LINE\_2 OOTB. What should you do?



# Achieve Balance – Refactored Upgrade

---

- Use database and tools upgrade.
- Selectively replace customizations with OOTB where cost/benefit warrants.
- Consider refactoring other parts of configuration with new and better configuration where cost/benefit warrants and new features allow for things like code removal.
  - E.g. Replace certain eScript code with workflow processes now that wfp run so much faster than before.
  - Could always do refactoring as part of Phase II after upgrade.



# Siebel Upgrade Overview – Two 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 merged but all objects including new totally custom objects can be changed in the upgrade process.
- Note: Siebel upgrade runs in all environments including dev, test and production.
  - Question: When is it ok to not run the Siebel upgrade in certain environments?



# 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 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 is not always correct in all situations.
  - Advanced upgrade best practices sometimes require slight changes to the steps.



# 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)

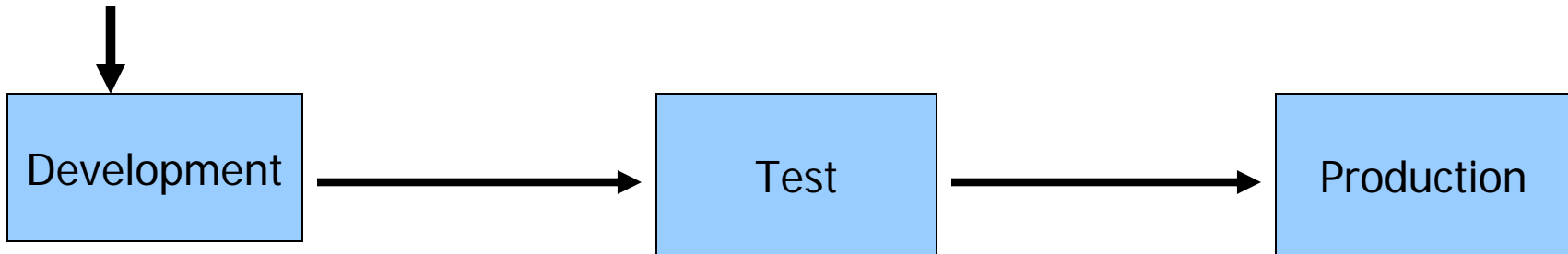


# Automated Siebel Upgrade Processes

Step	Development	Production Test (QA)	Production
Prepare for Production Upgrade		Only runs here	
Upgrade Siebel Database Schema (upgrep)	Yes	Yes	Yes
Tools Merge, Post Merge Utilities, Generate EIM Processing Columns, Resolve Merge Conflicts	Only runs here		
Upgrade Custom Database Schema (upgphys)	Yes	Yes	Yes

# Upgrade Data Flows Between Environments

New Siebel schema (ddl.ctl)  
Old and new Siebel repositories (rxxx.dat, mastrep.dat)



Customized Siebel schema (schema.ddl)  
Customized Siebel repository (custrep.dat)

Deduplication files (dedup1.sql, ..., dedup4.sql)

Non repository CM changes (LOV changes, state model, assignment rules, workflow policies, web service definitions, symbolic url's, etc.)



# Things You Might Not Have Known About The Siebel Upgrade

- Most Siebel upgrades take longer and cost more than they should.
- System performance is not always better with the new version of Siebel (until you work on it).
  - No doubt UI performance is much better with 7.8/8.0 than 7.0/7.5.
  - Sometimes new features inadvertently hurt performance.
- Removing customization and returning to OOTB Siebel can carry a high price tag.
- 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.
- Even using non scripting alternatives can lead to some work on Siebel upgrades.



## Live Demo – Running the Upgrade

---

- Database server configuration upgrep
- Logparse and the upgrep output
- Repository merge
- Database Server configuration upgphys
- Siebel Upgrade Guide
- SupportWeb and the Siebel Upgrade



# How Long Will My Upgrade Take?

- Based on historical evidence, it is difficult to perform a Siebel upgrade in less than four months.
  - World record highly complex Siebel 6.x -> 7 upgrade was performed in just four months instead of normal 12 months.
  - Not many 7 -> 7 Siebel upgrades that include production users have come in at 4 months or less.
- Basic formula for upgrade timeline based on project critical path. Could you do any of these in less than 1 month each?
  - Planning and prep time.
  - Development upgrade time.
  - QA testing time.
  - Production upgrade and rollout time.
- Question: Can any of these be overlapped and if so how?

# Two Nearly Identical Upgrades – Two Vastly Different Timelines

Attribute	Upgrade 1	Upgrade 2
Timeline (plan 2 prod)	4 Months	9 Months
Cost	\$250K	\$750K
Version	6 -> 7.7	6 -> 7.8
Customization	Highly complex, eScript, interfaces, reports, etc.	Highly complex, eScript, interfaces, reports, etc.
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



# Factors That Influence Your Time

---

- People
- Organizational Characteristics
- Size of Siebel implementation
- Complexity of implementation
- Criticality of Siebel to business
- Limited downtime or large database size
- New/changed Siebel modules
- New features added as part of the upgrade
- Upgrade approach and return to OOTB
- How much time your staff must spend on other things including production support
- From/To Siebel version



## Did You Configure Siebel Correctly?

- Were OOTB applets extensively modified or were copies created?
- Were unwanted items inactivated or were they deleted?
- Was scripting minimized?
- Prefixing/suffixing new/copied objects can sometimes help with the upgrade.
- External programs access Siebel via XML or object interfaces only and no direct SQL?



# 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 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.



# Development Upgrade Expectations

- Application probably won't launch after upgrade due to one or more configuration errors that will have to be fixed.
- 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 will require changes even on Siebel 7 -> 7.8/8.0 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 7.7/7.8 removal of page tabs from main screen list views (aggregate view require drilldowns to see tabs).



# QA Testing Time

---

- Largest component on 7.x 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 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.
  - Unfortunately “big bang” is the best practice for Siebel upgrades.
  - Better to have a “cushioned big bang” with immediate fallback in place.
- Always best to 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.



# Other Tasks To Consider

- Upgrade tuning sometimes requires months for large databases with limited downtime. Only about 10% of Siebel customers need to be concerned.
  - Question: How to know if you are one of them?
- Implementing new modules like 7.8 Order Management can be a very large project in itself.
- Migration from very old Siebel versions 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?



# Planning Tips

- 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



# Upgrep Tips

- 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 Tips

- 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.



# Tools Merge Tips

---

- 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 MSTSC or similar.
  - If Windows app server consider installing Tools here.
- Delete old repositories especially if there are a lot of them.
  - Now documented in Upgrade Guide.
- 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 Tips

- 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 Tips

- 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 Tips

- Perform several practice runs exactly the same way as the real production upgrade will be performed.
  - Question: What does exactly mean?
- 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.



# Questions and Answers

---

Respond at [ponderproserve.com](http://ponderproserve.com)

770.490.2767