



ORACLE®

Siebel Product Migration Best Practices

David Fan
Oracle Communications

Robert Wing
aMind Solutions



Siebel Product Migration Best Practices

Agenda

- Introduction
- Business Requirements
- Migration Options & Sequencing
- Best Practices
- Migration Improvement Opportunities
- Q&A



About aMind Solutions

Founded by ex-Siebel experts & Architects who designed and developed the Siebel Order Capture and solutions

Company was founded to address the market need for end to end Order Management solutions

Have successfully engaged with Fortune 1000 companies in North America and Europe

Recognized globally as the “Best in Class” organization for Order Management expertise

Member of Oracle/Siebel Advisory Board for Order Management

Offer consulting services and solutions for end to end Order Management deployments with Siebel as the center piece

Functional Areas of expertise

- Order Capture

- Product and Price modeling

- Deal Management

- Supply Chain integration of Order Fulfillment

- Web Self Service

Product Expertise

- Siebel - Product and Price Master

- Siebel - Order Capture Processes

- Siebel Deal Management

- Web Self Service enablement



Sample Customers

SIEMENS

ORACLE



NOKIA
Connecting People

Level 3
COMMUNICATIONS

Honeywell



Rabobank

PitneyBowes

NetApp

PHILIPS



IR **Ingersoll Rand**
Inspiring Progress™

“Engaging aMind has been the best decision we made in deploying our networks order entry solution”

Director, Major Telecom Company

“We have never engaged with a team as competent as aMind. Their abilities have allowed us to design and implement a world class order capture solution”

Program Manager, Fortune 1000 Hi-Tech Manufacturing Company



What We Will and Wont' Cover

Will Cover:

- **Migration of Siebel entities and data between different environments, such as promoting changes from Development → Test → Production or setting up a new test environment**

Will Not Cover:

- **Loading legacy asset data into Siebel (Asset Migration)**
- **Updating existing Asset, Quote, or Order data to significantly different new product models (Bulk Orders)**
- **Cache Management to ensure that newly migrated content takes effect**



Business Requirements Drive the Migration Approach

There is no “One Size Fits All” Solution Out of the Box

- **Nobody uses all entities (and their migrations)**
 - Ex/ not everyone uses product compatibility
- **Everyone adds their own extensions and new entities**
 - Ex/ new discount matrices or extending eligibility matrices

Migrating Products includes much more than just the projects themselves

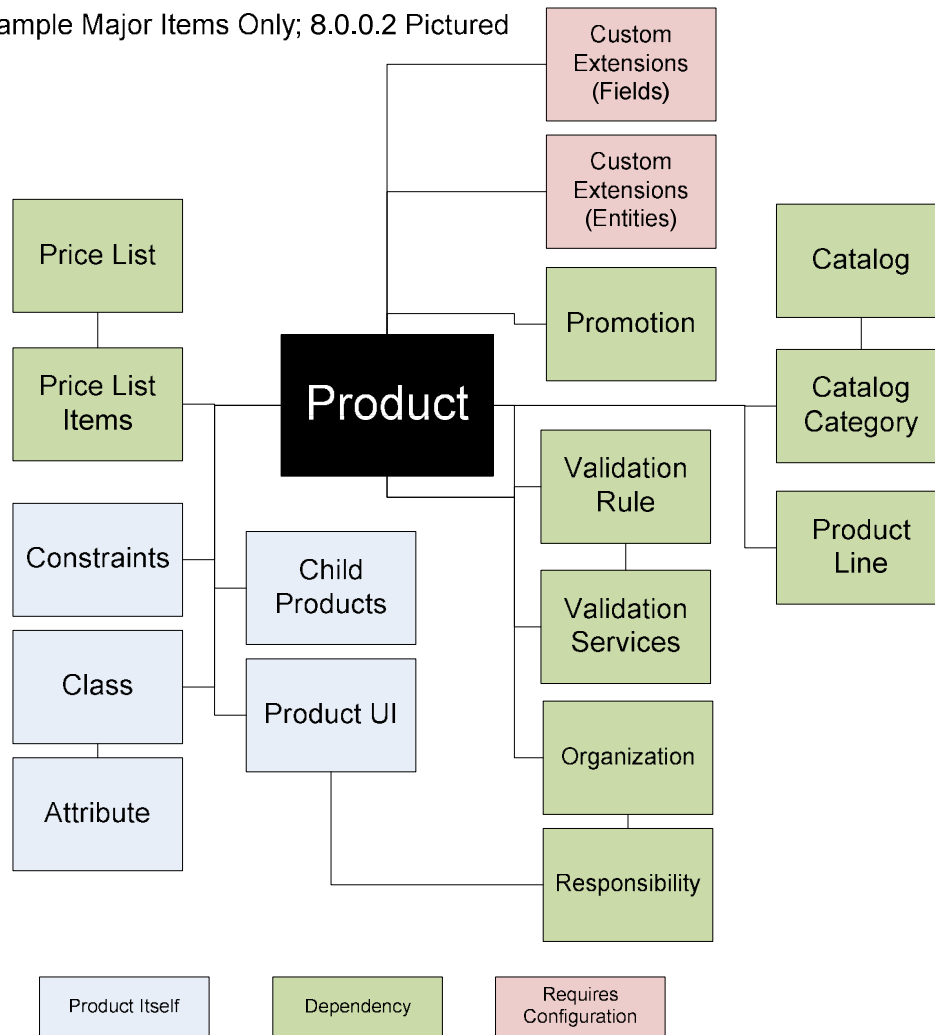
There are many building blocks to start from to assemble your solution:

- **Individual Import/Exports**
- **ADM Content Objects**
- **Tools SIF**
- **Workspace**
- **EIM**

A Product's Dependencies Require Care in Migration Design

Customizable Products and Typical Dependencies

* Sample Major Items Only; 8.0.0.2 Pictured



Careful attention to the sequence of migration is critical to getting all product dependencies covered:

1. Organizations
2. Responsibilities
3. Access Groups (and Party dependencies)
4. Product Web Templates & Images
5. Product Lines
6. Simple Products and Promotions
7. Full Product Structure (Workspace)
8. Validation Rules (ADM Content)
9. Catalogs and Categories
10. Eligibility & Compatibility Matrices
11. Price Lists/Cost Lists and Pricing Matrices
12. Other Pricing Adjustments

This list of common major entities should be validated against what is actually being used + additional configuration implications.



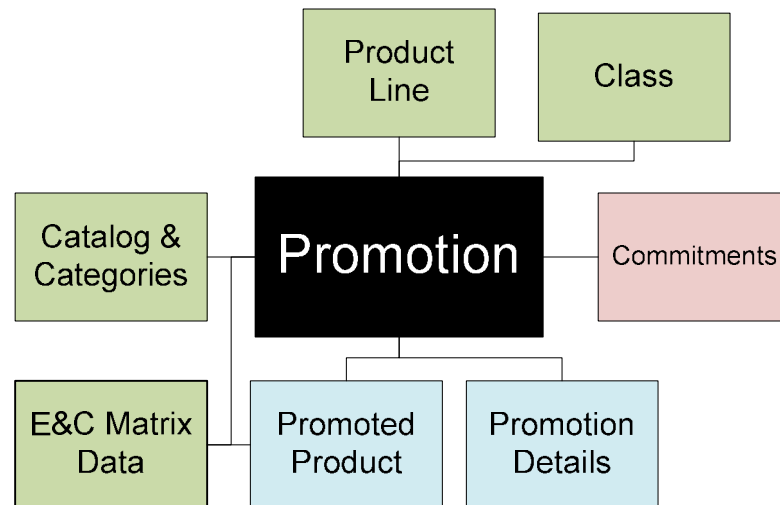
Tuning Existing Product Imports and Exports

Product Workspace Imports provides some special options that are not well known; using these features can help ensure smooth migrations.

- **Report Only**: identify all problems with migration without actually performing it, permitting problems to be addressed in a more orderly manner
- **Auto Proceed**: allows automatic conflict resolution vs. immediate abort when problem is encountered. It should be used in combination with report only mode so that report is complete with all problems.
- **Use Names As Keys (Name Based Import)**: useful while fixing name collision conflicts or while doing an upgrade to 7.8 or beyond. It simulates pre 7.8 behavior of the import mechanism by relaxing rules around object id conflicts during import. This is particular useful for project recovery where the “There Shall Be One And Only One Siebel Product Entry Point” rule has been violated.

Please see Tech Note 658 for additional information

A Promotion's Dependencies Require Care in Migration Design



Careful attention to the sequence of migration is critical to getting all product dependencies covered:

1. Access Groups (and Party dependencies)
2. Simple Products
3. Classes
4. Product Lines
5. Conditional Charge Plans
6. Full Product Structure (Workspace)
7. Promotions
8. Catalogs and Categories
9. Eligibility & Compatibility Matrices

This list of common major entities should be validated against what is actually being used + additional configuration implications.

* The ISS Promotion Integration Object is missing a few components to fully migrate Promotions. EAI Configuration will discuss this.



EAI Is Your Friend Migrated Content is Configurable

The Imports and Exports are just EAI operations

The Product Workspace is just a bunch of EAI operations

The Integration Objects can be configured to address unique challenges

- **Straightforward:**

- Add Fields
- Add Child Entities (be careful of adding additional dependencies!) Ex/ include product validation rules in the Workspace vs. a separate ADM migration
- Remove dependencies that you aren't using (improve import performance) Ex/ removing constraint editor graphical nodes usually reduces the product export size by ~75%

- **Approach with more caution:**

- Change Integration Component Keys
- Script the Exp / Imp Business Service

NOTE: Because an EAI Operation is a single transaction, it is not an ideal tool to use with large bulk data (1,000's to 10,000's or more records). In these cases alternate mechanisms to load the data such as EIM should be examined



Extending Siebel To Facilitate Migration

Objective: reduce the operational cost by automating manual migration tasks and vastly reducing migration errors; reduce production system down-time

Options:

- Create new “Workspaces” – controlled grouping of the export / import such that dependencies are accounted for and all needed entities are migrated
- Script and/or Workflow to coordinate the import process with the cache clearance process

Consider:

- Cost of development vs. cost of manually performing work
- Remember to consider all the times manual operations are needed, not just pushing forward from Development → Test → Production
- Remember to consider the cost of time spent resolving migration errors
- Consider the cost of delays in the project, not just the work hours



Approaches Targeted to The Entity

Entity	How Migrated
Class, Attribute	Own Imp/Exp, Workspace, ADM
Complex Product (CP)	Own Imp/Exp, Workspace* ¹ , ADM
Price List and Price List Items	ADM, EIM
Compatibility, Eligibility & Pricing matrices	ADM (8.0)
Promotion	Own Imp/Exp, Workspace* ² , ADM(8.0)
Catalog, Category and related Objects	ADM* ³
Product Validation Rule	ADM (8.0)

*1 Entities like Product UI Controls, Product Literature require additional configuration

*2 Promotions themselves are migrated, some related objects are not handled (ex/ commitments).

*3 OOTB Catalog ADM is provided as an example of how it could be configured, but it doesn't cover full range of data structure and hierarchy (ex/ categories migrate as a flat list, not as a hierarchy).



Approaches Targeted to The Entity (cont.)

Entity	How Migrated
Signal* ¹	Own Imp/Exp, Workspace
Variable Map* ¹	Own Imp/Exp, Workspace
Data Map* ¹	ADM
Data Validation Rule* ¹	Own Imp/Exp
Symbolic URL	ADM (8.0)* ²
Web Service Definition	Own Imp/Exp* ²
Workflow	Own Imp/Exp, Repository

*1 Dependency on underlying repository objects.

*2 Example of something that needs to be changed between environments to point to the appropriate ESB or other system



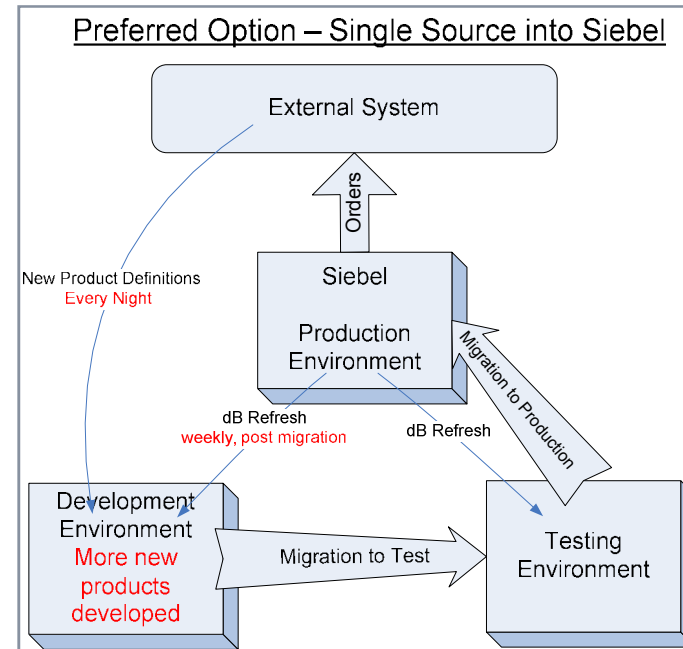
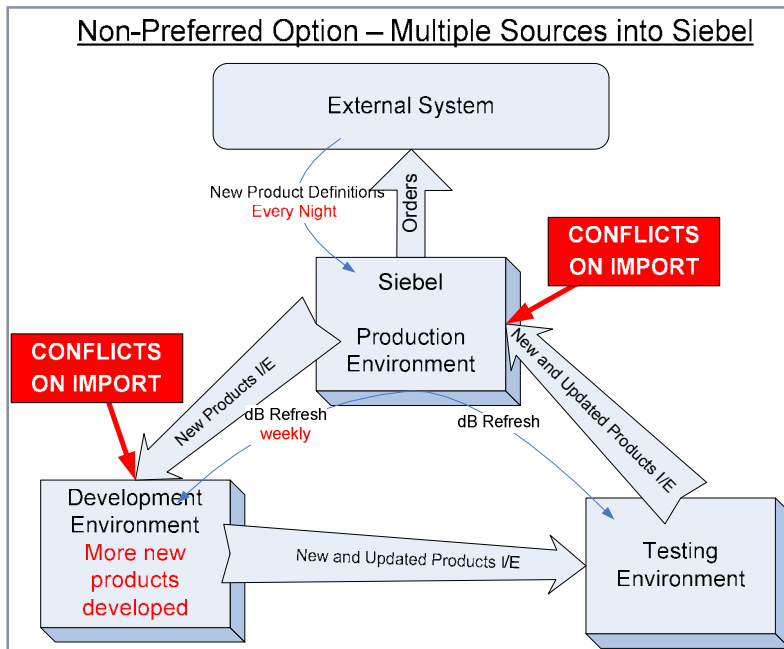
Common Scenario: Products Sourced from Back Office; Structure Defined in Configurator

Many companies integrate their product list from a back office system, such as Billing or ERP, while also defining sales configuration models in Siebel. This becomes an OOTB process with AIA for Communications.

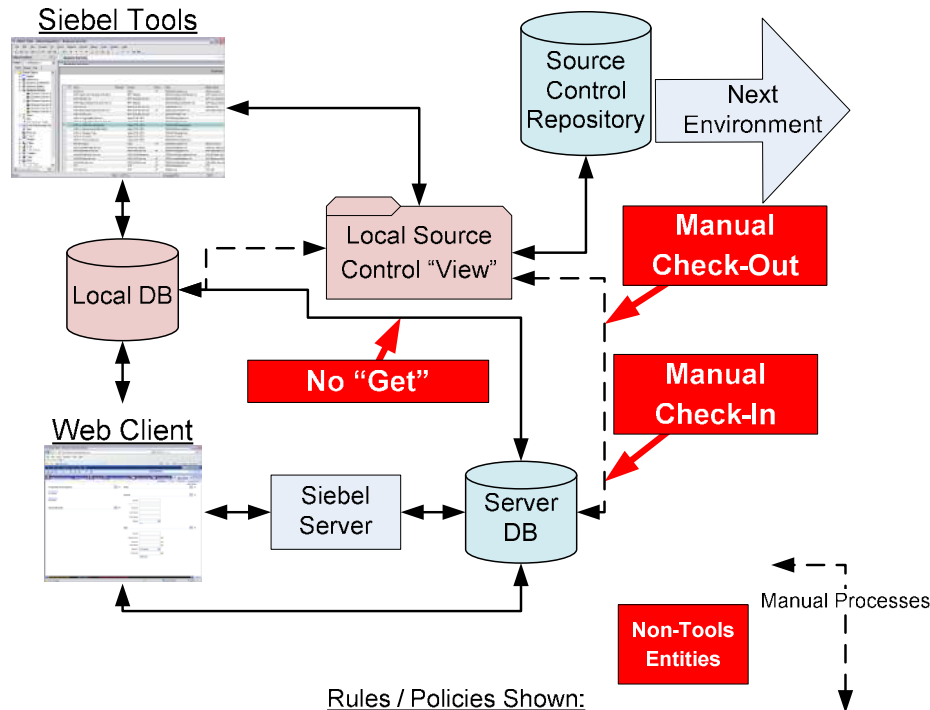
Whenever there are multiple integrations involved for a single entity (migrations are integrations, too) it is critical that:

- The integration keys for identifying a record match (technically or logically)
- Those integration keys satisfy the User Key for the table

Best Practice: have only one point of introducing products into the Siebel ecosystem



Source Control Integration



* Everything that promotes between environments MUST come from Source Control

For those companies that use a Source Control (VSS, Clear Case, etc.) system with their projects, there are additional considerations to plan for:

- Most entities used on OM projects (products, catalogs, signals, variable maps, copy maps, etc.) are manually integrated with Source Control.
- These same OM entities must be manually 'Got' from the server to each developer's local environment
- ✓ Clear procedures, notifications, and disciplined execution are needed to avoid breaking local environments, keep source control current with the Server DB, avoid promoting outdated configurations, and developing to outdated dependencies



Assembling a Solution

Incremental Improvements

- **Controlled Administration in Production**
- **Extending Existing Exports / Imports**
- **Using High Efficiency Exp / Imp Options**
- **Improved Troubleshooting of Migration Errors by Siebel Admin Personnel**
- **“Procedural Workspaces” – systematically manually perform the steps to identify and include dependencies**

Larger Improvements

- **Creating Additional “Workspaces”**
- **Cache Clear & Re-Load Scripts**
- **Systematic Migration Validation Tools**



Why Not Administer in Production? Some Companies Do...

Objective: permit certain changes to take effect immediately (w/o migration delay) and reduce migration costs for items that are very frequently changed

Carefully identify the places where changes:

- **Are intuitive enough that they can be made without material testing.**
- **Are tightly contained and have no “ripple” effect.**
- **Are relatively frequently incrementally changed.** Avoiding the migration is a major cost savings
- **Have acceptably minor consequences in the event of an error**

Example Change	Assessment
Discount Matrix % Change	Appropriate; frequently changed, low risk, and no ripple effect
Introducing New Products	Not Appropriate; impacts back office
Signal Change – Additional Functionality	Not Appropriate; development work should be tested
Currency Exchange Rates	Appropriate; frequent, necessary, no ripple



Fixing Migration Errors (Basic Option)

Problem: when migrations go wrong, it can be difficult to figure out why. “It works in one environment, but not in the other”

Typical Reasons (we’re all human):

- Migration procedure mistakes
- Not all dependent objects needed to implement a change were identified
- Something was missed when manually integrating with Source Control

Solution: Once a small set of possible problems have been identified, export the XML for those entities (products, variable maps, workflows, etc.) from both environments and compare them with a commercially available Different Tool (WinMerge, UltraCompare, etc.)

Provides: A detailed itemization of the differences between the environments

Limitations:

- Comparing everything this way takes too long – you need to already have a good idea where the problem is
- This will compare everything, not just what you care about. Expected and proper differences in Row Ids produces many ‘false positive’ differences that take time and expertise to parse through

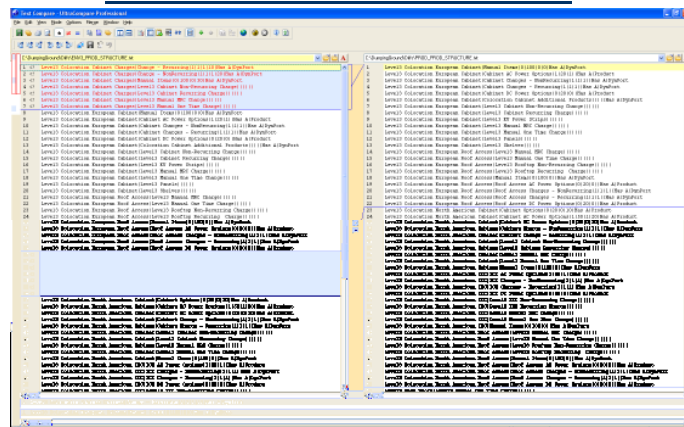
Finding and Fixing Migration Errors – Advanced

Enhancement: execute a bulk comparison for all entities, examining only the fields that drive functionality

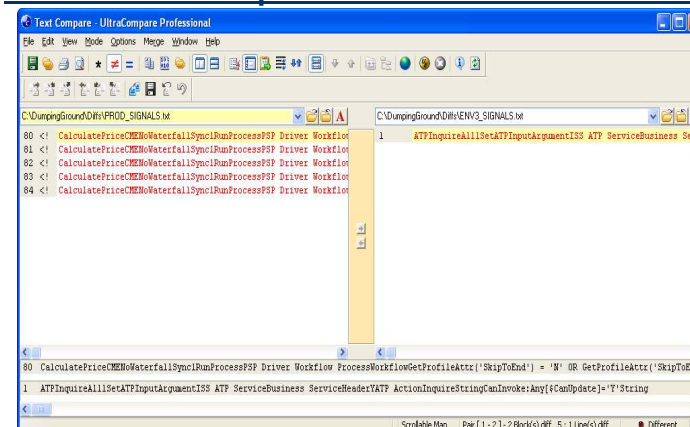
- Avoids the need to manually debug the errors down to a small handful of options, reducing the knowledge level and skill set needed to find and fix the error
- All differences are bad differences, making the test much more conclusive

These comparison tools are tailored for each use to (a) include custom extensions and (b) only compare the vanilla entities being used

Product Structure - All



Variable Maps – Differences Only



Best Practices – Summary

Include Migration Implications in Design and Scope Decisions – consider the time and cost of migration enhancements (software and/or personnel time) when making functional enhancements or adding integrations

Take the time to figure out what you need – the cost of execution here is frequently much less than the total cost of failing to execute; itemizing all the dependencies and developing procedures and/or tools to handle them takes time

Cost-justify and develop needed automation tools – these tools can considerably reduce both the final cost and risk of a deployment

Plan Appropriately for Parallel Development – there are parts of the application that merge more smoothly (Tools objects) than others (products, variable maps, etc.)

Arm Migration Personnel With Basic Debugging Capabilities – to reduce the cost of migration errors by finding and fixing them quickly using minimum staff



Improving Migrations – Best Practices Review & Techniques

aMind Solutions can improve your migration times and reduce their costs by applying and tailoring our pre-built tools and processes to your environment. This engagement would target:

- **Migration process automation**
- **Automated migration validations**
- **Streamlined debugging and correction of migration errors**
- **Best practices policies and procedures**
- **Server memory cache management with a migration**
- **Designing for migrations in integrated environments**



Migration Automation Tool Option

Migration Automation tool set is being developed that would address many of the fundamental challenges:

- Automate the migration of major OM and standard entities, eliminating many labor-intensive and time-consuming manual processes
- Coordination with Source Control for everything, not just Tools
- Establishing cross-entity dependencies and handling them with improved import processes
- Simplify auditing the success of a migration and debugging any errors



Moving Forward

Would you be interested in a future webinar on the following topics?

- Loading legacy asset data into Siebel (Asset Migration)
- Updating existing Asset, Quote, or Order data to significantly different new product models (Bulk Orders)
- Cache Management to ensure that newly migrated content takes effect



ORACLE®