

Siebel OS/DB Migration

Migrating your Database to SQL Server 2000

Published: April, 2004

For the latest information, please see:

<http://www.siebelonmicrosoft.com/mssiebel/Technology/OSDB.aspx>



Table of Contents

Chapter 1: Introduction to Migration.....	1
Target Audience.....	2
The Microsoft-Siebel Global Alliance.....	3
Frequently Asked Questions.....	4
Why Migrate to SQL Server?	4
What are These Types of Migrations Referred to as?	4
Who Can Perform the Migration?	5
Can I Perform the Migration on My Own?	5
What Role, if Any, Does Microsoft Play in a Siebel Migration?	5
Is There a Certification Program for Partners who Perform these Migrations?	6
How Do We Ensure a Migrated System will be supported by Siebel?	6
What is Typically Involved in a Migration?	6
How Long Will the Migration Take?	6
Can I Perform an Upgrade in Conjunction With a Migration?	6
Chapter 2: Data Migration and Loading	9
Data Movement Using EIM and DATAIMP/DATAEXP	10
Using EIM.....	10
Using DATAIMP/DATAEXP	10
Methods for the Conversion.....	12
Siebel 7 to Siebel 7: Just Changing Databases	12
From Siebel 6 to Siebel 7: Changing Databases & A Siebel Upgrade	12
Validation of Data.....	15
Performance Optimization of EIM for SQL Server.....	16
Migrating Using DATAEXP/DATAIMP	18
Migrating a Siebel Enterprise from Oracle to SQL Server.....	19
Customization Considerations	21
Migration Considerations and General Performance Characteristics	22
Summary.....	23
Glossary	25
Chapter 1 Abbreviations	26
Chapter 2 Terms	27
Appendix A: Siebel OS/DB Migration Example Project Plan	28
Appendix B: Websites for Additional Information	31
General Microsoft and Siebel Resources	32
Microsoft Classes.....	33
SQL Server Resources	34
System Integrator Resources	35
Appendix C: Sample Code.....	37
Chapter 2	37

Chapter 1: Introduction to Migration

Companies that choose to migrate their Siebel database platform do so for a variety of reasons. Whatever the reason the benefits are substantial. To gain a clear understanding of these benefits and provide our customers with an idea of what they can expect, Microsoft commissioned a study to be conducted by NerveWire, Inc., an integrated management consulting and systems integration firm. The NerveWire study listed and verified the benefits of migrating Siebel CRM (Customer Relationship Management) to Microsoft SQL Server 2000. The study is available at the following URL:

http://www.siebelonmicrosoft.com/mssiebel/Downloads/PDF/Nervewire_CRM_Migration.pdf

To help other customers realize the cost savings and benefits of running Siebel on the Microsoft .NET platform, Microsoft is putting together a comprehensive solution for the migration process while working closely with Siebel and other partners. This paper is part of that comprehensive solution.

The goal of this paper is to discuss the major parts of a Siebel migration to Microsoft SQL Server, who is involved, and the planning aspects of the migration. Figure 1-1 provides a high level road map of the migration process.

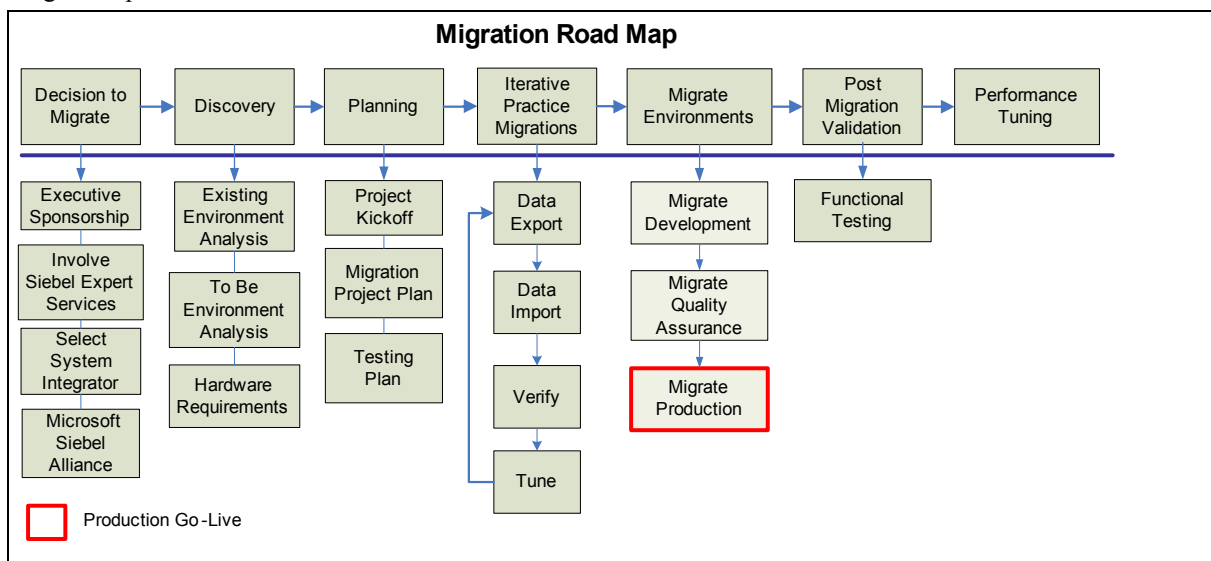


Figure 1-1: Migration Road Map

In the course of this paper, we examine a few key areas involved in a migration, while providing in depth discussion on the migration and loading of data as well as the performance tools used during the migration process.

Moving the data from the source platform to Microsoft SQL Server is actually the fastest and easiest part of the migration. Because the bulk of work in a migration is spent planning, a sample template for a project plan is provided in Appendix A. Additionally, links to other Siebel non-specific reference materials and supporting documentation are provided in Appendix B. To allow better flow and readability of the document any code referenced is placed in Appendix C.

Target Audience

This whitepaper is intended to help technical professionals who are migrating the Siebel database for the first time or for IT Managers wishing to understand the migration process at a deeper level in order to make well informed decisions and plans. The information in this publication is not intended as a specification of any programming interfaces that are provided by Siebel or Microsoft. Refer to Microsoft SQL Server Books Online and Siebel documentation for more information about what publications are considered to be product documentation.

The Microsoft-Siebel Global Alliance

The Microsoft-Siebel Global Alliance is a strategic relationship formed between Microsoft and Siebel in 2002 to optimize and grow the technologies between the two leaders in each market. Siebel providing a superior Customer Relationship Management (CRM) solution and Microsoft leading in the platform area.

It is much more than a marketing push. It has over 30 people engaged in the effort, most of whom are involved in engineering. The support to the customer ranges from new engineering product initiatives, scalability testing, early adopter integration of technologies, and pre/post sales support, to extended support queues between the two organizations.

Beyond support roles and consulting, the Microsoft-Siebel Global Alliance provides numerous tangible benefits that no other platform offers. These include:

- Microsoft's Siebel Technical Champions program. This program was founded to grow the end user and partner technical community and keep them abreast of the latest technologies available on the Microsoft/Siebel platforms.
- Websites that provide technical papers in the form of notes that are fixes to common issues that customers have. For example, index optimization on SQL Server, high performance, and **Enterprise Information Manager (EIM)** loading with SQL Server.
- Proactive insights into Microsoft's engineering initiatives with Siebel. Current development projects include Outlook integration, mobile device development and many other topics.
- Executive escalation points within Microsoft. Often on mission critical applications and migrations, it can be more than just an engineering issue. Microsoft can provide executive level sponsorship to ensure the success of your program. These executives will get briefings to make sure that your project is meeting its goals. Additionally, it gives a company a single person to call for action.
- Free technical workshops covering a wide variety of topics (High Performance EIM loading, Siebel **High Availability (HA)** on the Microsoft Platform, and many others.). These talks are given by leading industry experts from Siebel-Microsoft and are available to all Microsoft and Siebel customers as well as system integrators and hardware vendors.
- Microsoft-Siebel Joint Support Queue. This ensures that both companies are working together to resolve your issues rather than assuming the problem lies with the other's product. Microsoft and Siebel monitor joint customers to ensure clear resolution to issues.
- Using the **Microsoft Technology Centers (MTC)**, customers can prototype and make proof of concepts using Microsoft's hardware and software. This cuts ramp up and conversion times by allowing immediate starts to projects without having to wait for hardware delivery.

For more information on this alliance and what it means to you, please visit the official Alliance web site:

www.siebelonmicrosoft.com

Frequently Asked Questions

To ensure a better perspective, the remainder of this chapter provides answers to some of the more frequently asked questions involved during your Siebel migration to Microsoft SQL Server.

For the best solution possible, please work closely with your Siebel Technical Account Manager (TAM) and coordinate with your Microsoft support team. If there are any questions, the Alliance can be reached via e-mail at: seblhelp@microsoft.com or <http://www.siebelonmicrosoft.com/>

Why Migrate to SQL Server?

Microsoft and Siebel work very closely to optimize performance on SQL Server. The Microsoft platform is the first choice for development, and then the code is ported to the UNIX variants.

From a business choice perspective, SQL Server provides many benefits for a CRM deployment, such as :

- A faster return on investment than it's competitors
- Faster deployment to the business users during initial deployments, future upgrades, and rollouts
- Lower operational costs due to self tuning parameters
- Lower hardware prices on commodity machines instead of expensive Unix servers
- Lower software and maintenance prices

All these points mean a quicker and leaner business which leads to beating your competitors in the marketplace.

From a technical point of view, the SQL Server database technology offers a vast array of benefits:

- Performance. Top rankings in both Siebel and the **Transaction Processing Performance Council (TPC)** benchmarking at a much lower cost per transaction than any competing platform.
- Tools. SQL Server bundles many out-of-the-box tools that others charge thousands of dollars for.
- Operations. SQL Server allows your DBA's to be out of the "knob turning" business. DBA's don't need to focus on optimizing the kernel. They spend time solving larger architectural issues.

The byline for SQL Server's design is "simple but not simplistic." This means that the database tunes and optimizes itself without the constant need of a Database Administrator (DBA). This is not to say that it's a "give and go" system that runs itself completely. It's not. Care and feeding are still needed, but various components of the system are self optimizing with the underlying operating system. For example, data and procedure cache that self balance with the needs of the Central Processing Unit (CPU) of other applications or the database optimizer which generates superior plans without the need of hard coding hints into the query. Both of these are required on some competing platforms for optimum results.

The aforementioned is just a brief list of why SQL Server provides a cost efficient, reliable, and scalable platform on which to migrate your Siebel deployment.

What are These Types of Migrations Referred to as?

There are several names people use to refer to migrations. Each has certain nuances about them. For example, operating system/database (OS/DB) migration, DB migration, or heterogeneous system copy. All have slightly different meanings. Table 1-1 provides a high level vocabulary view of the various types of migrations and what they typically entail.

Table 1-1: Types of Migration

Migration Type/Name	Description
OS/DB	Changing platform (operating system) and database. Example: migrating from SUN-Solaris/Oracle to Windows/SQL Server
OS	Platform (OS) only. Example: migrating from HP-UX/Sybase to Windows/Sybase.
DB	Database only. Example: migrating from Windows/DB2 to Windows/SQL Server.
Heterogeneous System Copy	A less used term but encapsulates either of the previous situations.

Who Can Perform the Migration?

Due to the complexities of multiple environments, multiple operating systems, testing, rollouts, operational planning, high availability strategies, etc., it is generally encouraged to work with an experienced systems integrator (SI) who has the resources at hand and can coordinate with Siebel Expert Services (ES) to perform the migration.

The SI can provide a wealth of experience around planning, testing, and rollouts. Several SI's have migration service offerings. Additionally, Microsoft Consulting Services can help, too. It is imperative that Siebel Expert Services is involved. ES is the only body that can certify the migration itself thus ensuring supportability of the final product.

Please look at the following website for some of the service offering that SI's have in this area:

<http://www.microsoft-isvpartners.com/siebel/resources.htm>

Can I Perform the Migration on My Own?

The migration process itself is documented, but it is highly discouraged to try and perform one on your own. Expertise is required in both the source and target databases as well as Siebel. Additionally, expertise would be needed in advising on a support strategy as well as other activities. Finally, you will still need to certify the system, which can only be done by Siebel Expert Services. Otherwise, your software warrantee may be rendered void.

What Role, if Any, Does Microsoft Play in a Siebel Migration?

Depending on the project and the level of expertise of the SI partner and the customer in Microsoft technologies, Microsoft Consulting Services (MCS) may be used to mitigate technology risks and possibly to assist in optimization, designing administration processes or integration tasks. However, the actual migration generally does not require MCS involvement.

The following is a brief list of just some of the services that MCS can provide that complement an SI and ES during a migration:

- Optimization of SQL Server. For example, making sure that the database is optimally configured for the underlying **Storage Area Network (SAN)** or hard drives.
- Operational best practices review. This will ensure that the customer has their support team properly trained, backups in place, and disaster recovery well planned, etc. Additionally, MCS can help work with the customer to configure tools like Visual Source Safe and **Microsoft Operations Manager (MOM)** for build control and operational support management in a Siebel environment.

- EIM data loading. Microsoft has extensive experience with Siebel data loading tools. We have worked at most major Siebel/SQL Server customers world wide on their data loading. Papers, workshops and consulting are all available on this topic.
- Troubleshooting. MCS has extensive experience in debugging issues and handling Siebel escalations. MCS and the Microsoft Siebel Alliance are able to work closely with Microsoft's **Premier Support Services (PSS)**, a deeply technical support offering from Microsoft, as well as Microsoft and Siebel engineering to solve the problems at hand.
- Automation and Monitoring. MCS can help automate routine maintenance jobs such as backups and index defragmentation. In addition, they can also help configure alarms for when a failure occurs. This could be used for Windows Server and SQL Server using the native tools and/or MOM.

Is There a Certification Program for Partners who Perform these Migrations?

No. While some software vendors provide an OS/DB migration certification course and a standardized process, Siebel currently does not. It is up to you to work with an experienced SI to put together a migration strategy with Siebel Expert Services. Each situation is unique. Several SI's have experienced personnel in migrations and have supporting plans and material around this area, but there are no Siebel official migration certifications independent of Siebel Expert Services.

How Do We Ensure a Migrated System will be supported by Siebel?

The only way to have a supported migration is to make sure Siebel Expert Services is involved right from the start. They are the only party that can certify the "migrated system" is compliant with Siebel requirements and is a supported system.

Failure to engage Siebel Expert Services can mean your system is non-supportable.

What is Typically Involved in a Migration?

The basic concept is that you need to export your data from the source database and load it into SQL Server. There are of course many details in planning around this activity. Moving the data is only one part of the process. For example, the following elements all need to be coordinated: planning the new hardware, stress testing the application, assuring the deployment of new drivers to connect to the database, new high availability strategies for the target database, operational reviews for the new environments, migration of the data in all environments (development, quality assurance, production, and training). In short, a migration is a comprehensive event that needs to be addressed in its entirety. Movement of the data is only part of the solution. The rest of this paper will address the major moving parts and their considerations during a migration process.

How Long Will the Migration Take?

Moving the data is actually the fastest part. It can usually be done in a weekend of down time for most Siebel databases. The planning and execution of the migration in its entirety should be from 1 to 3 months depending on the complexity of the customer's environment. Further considerations may need to be addressed if a Siebel upgrade is also being contemplated. Work with your Siebel TAM to determine the proper path.

Can I Perform an Upgrade in Conjunction With a Migration?

Yes and no. Migrations are typically considered at the time of an upgrade because there are changes being planned already. Due to the quantity of "moving parts" in an upgrade alone, it's generally encouraged to think of a migration as separate project from the upgrade. For example, there may be certain compatibility issues

that will not allow you to migrate prior to the upgrade. As such, you should consider them two separate projects. Either migrate first, then upgrade, or the other way around.

A common solution that many companies take when moving from Siebel 6 to Siebel 7 is to install a clean SQL Server-Siebel instance, put the new configuration data in the SQL Server instance, then EIM the data over from the source system as the final part of the upgrade.

Chapter 2: Data Migration and Loading

After the decision has been made to migrate your Siebel database, you need to know how to move the data between the platforms. DBAs typically think of using bulk tools such as BCP, DTS, SQL Loader, etc. However, these tools are not always applicable in a Siebel environment due to the constraints of the Siebel application. For these reasons, custom made solutions to data loading in a Siebel environment are not supported and you need to use Siebel provided tools. Generally speaking, there are two different supported methods for converting Siebel data between different platforms: Siebel's EIM or Siebel's DATAIMP/DATAEXP.

EIM is a batch-operated utility that is a Siebel mainstay for moving most data. DATAIMP/DATAEXP are less known and are bulk loading tools. Each method has its nuances and uses. In this chapter we will explore the pros and cons of each method and how to optimize the data movement processes.

When migrating platforms, there are other considerations beyond moving data. The following is a short list of some:

- Porting proprietary stored procedures between platforms
- Moving data between different data structures and types
- Customization of the Siebel application
- Change in optimizers, data types, and other dependent structures
- Version specific issues of Siebel

Examples these would be conversion of PL/SQL stored procedures to Transact-SQL stored procedures, converting from Oracle sequences to SQL Server identity columns, extended columns to Siebel tables, and ported builds that may be incompatible from older versions of Siebel 6.

Data Movement Using EIM and DATAIMP/DATAEXP

As just mentioned, EIM and DATAIMP/DATAEXP are two Siebel-provided methods for moving data between platforms. The choice depends on what you are trying to accomplish given a certain set of criteria. If you are migrating as part of an upgrade, then a batch-oriented tool like EIM is probably the way to go. If your company is just moving between platforms, then bulk tools like DATAIMP/DATAEXP are arguably a better choice. The following sections explore these choices in more detail.

Using EIM

EIM is designed to move data from legacy systems into Siebel or exporting the data from a Siebel database. It's commonly used for the initial data loading process. Table 6-1 lists its pros and cons.

Table 6-1: EIM Pros and Cons

Pros	Cons
<ul style="list-style-type: none"> • After basic data mappings have been done and an IFB file created, EIM does the rest • Handles errors, duplicate data and orphaned data • Makes schema changes between Siebel versions much less significant to the migration efforts. For example, the “Party Model” in Siebel 7 that didn’t exist in Siebel 6 • No interim flat file storage system is needed. The data can be unloaded into a database table, then linked to the target server and re-imported. Database caching efficiencies help over a flat file. 	<ul style="list-style-type: none"> • Can only process as many records as in one batch. Batches tend to be 5,000 to 20,000 rows in size. • Data loader tools to populate and archive the EIM interface tables need to be made • EIM needs to be watched. It’s not typical to run large volumes in a “lights off” environment • Can cause a lot of extra overhead that is generally not needed when copying data between two Siebel systems. For example, the Siebel generated ROW_ID’s are already in the source system • Time to load large quantities of data can be prohibitive in a “Go Live” weekend. • Defragmenting the database on large loads may be needed

Using DATAIMP/DATAEXP

DATAIMP/DATAEXP are bulk loading tools provided by Siebel to move data between systems. They are actually two distinct tools. DATAEXP exports the Siebel source database to a flat file. DATAIMP imports the flat file into the Siebel database. Table 6-2 lists the two’s pros and cons.

Table 6-2: DATAIMP/DATAEXP Pros and Cons

Pros	Cons
<ul style="list-style-type: none"> • Moves data extremely fast • Extra processing overhead does not occur. For example, ROW_IDs are not generated. • Optimizations can be made at the database level 	<ul style="list-style-type: none"> • Does not check referential integrity between systems. Quality of data is not validate. • Duplicated and orphans are possible. • Does not work well between large Siebel versions. For example, significant schema

<p>level</p> <ul style="list-style-type: none">• Less prone to bad execution plans due to fewer SQL Server operations.	<p>changes took place between Siebel 6 and Siebel 7.</p> <ul style="list-style-type: none">• Not very forgiving where customers have modified schema. For example, adding columns.• Extra disk space is needed for the interim flat file
--	---

Methods for the Conversion

This section explores the different reasons to choose one method of conversion over the other given the previous advantages and disadvantages of each tool. Two common scenarios are presented:

- Keeping the same version of Siebel, and only changing database platform
- Upgrading versions of Siebel and changing database platform

As usual, mileage varies in each situation and set of circumstances. Carefully review your needs with your Siebel TAM and Siebel Expert Services for the best solution that fits your site.

Siebel 7 to Siebel 7: Just Changing Databases

If your goal is to switch database platforms, and there is no underlying application change (i.e. upgrade), the easiest and fastest route is to use the Siebel bulk loading tools DATAIMP/DATAEXP. The main reason is pure speed for the down time production conversion window and “the fewer moving parts, the better” strategy. In systems design, a main goal is simplicity. This is because the more complex something is, the greater the chance something is that will break. Data loading is no different. DATAEXP/DATAIMP offer a compelling story for simplicity. A DBA dumps the data to a flat file and re-imports it into the target system using the utilities. The speed efficiency comes from using database specific bulk operations, and doing the conversion in mass operation, as opposed to many individual batches (which is what EIM would involve).

When planning your downtime on the cut over weekend, a lot of data needs to be moved over in the shortest possible time. This is a priority from a management perspective, because it ensures that there is ample time to bring the system up and validate the results of the conversion prior to handing it off to the end users.

Additionally, somewhere before the migration is actually finished, management approval is needed.

For example, if a conversion is supposed to be done at 3PM on Sunday, typically status e-mails are sent out at regular intervals to management letting them know the state of the conversion. If the conversion does not go as planned, you may end up scrubbing the migration and redoing on another weekend. Hence, getting the operation done as soon as possible ensures that validation and management approval occur as soon as possible. The biggest caveats to these utilities lies in the fact that the source and target schemas need to be in sync. That means that any customization on the source system needs to be in place prior to the load on the target system. To do this, make sure the repository is in place and a DDL sync has been run. Indexes are not as much of a concern due to the fact that you probably will drop them prior to the actual loading of the data and put on after the load is complete.

From Siebel 6 to Siebel 7: Changing Databases & A Siebel Upgrade

The most common scenario the Microsoft Siebel Global Alliance is seeing now days is a conversion in tandem with a Siebel upgrade. This is due to budgeting and time line considerations by customers. The drawback to the double operation is the level of complexity this adds to project management and the technical considerations of the project.

There are several tactics one can take when migrating data platforms during an upgrade:

- Lateral conversion of the database then upgrade the Siebel application.
- Upgrade the Siebel application then laterally move the data to the new database.
- Make a clean install on the target database platform. Redo your Siebel configuration from scratch on it. Then EIM the data over from the source system.

Figure 2-1 illustrates the first two upgrade paths.

Two Phase: Upgrade and Migrate

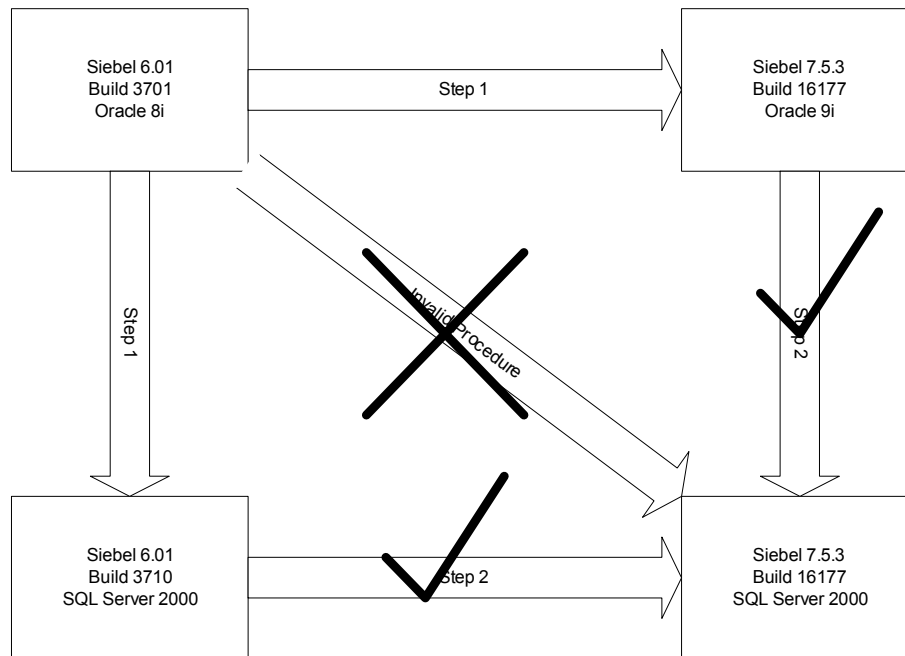


Figure 2-1: Upgrade paths

Option 1: Lateral Conversion, then Upgrade

There can be technical issues with this option. The first and foremost is whether or not Siebel supports the current version on the target database. For example, if you are on an older version of Siebel, it may not be supported on the latest version of SQL Server. Hence, you would also need to consider upgrading the database after the migration. Usually, not a good option.

Option 2: Upgrade Application, then Migrate Database

This option is a viable one (Figure 2-2). It is just the reverse of option 1. Upgrade the application, then port the database. The benefit behind this model is that the target database version is not as critical.

Clean Install and Migrate Data

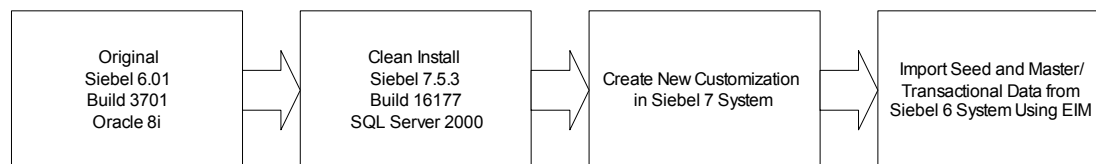


Figure 2-2: Clean Install and Migrate Data

Option 3: Clean Install, then Migrate Database

The third option is probably the most common one considered. The reason for this is that old Siebel sites have extra configuration that is no longer needed or that in changing from a two tier system to a web based application there is significant reconfiguration needed anyway. A new clean install of Siebel provides a fresh beginning, and the data—which is the only thing of value in the old system, is just migrated forward. The negatives behind this model are the analysis involved and the speed of moving the data. When using EIM, the IFB file needs to be defined. The IFB file defines the source and target columns between the Siebel interface tables (EIM_*) and the Siebel base tables (S_*). This is not a trivial task and can be time consuming.

The benefits are that Siebel makes sure there are no discrepancies in the PK/FK logic and comprehensive knowledge of the schema is not needed.

Validation of Data

A database is only as good as the information in it. As such, a post load validation of it's contents is very important after a migration. Typically, customers will define out a suite of results that they expect back. For example, quantity of customers. This suite of scripts is run both from the application level and database level. These results on the source database are in turn compared to the results on the target database.

An example of this would be to go into a contact screen as a Siebel Administrator, run a query and see how many contacts come up. Then run a comparable query at the database, for example:

```
select count(*) from SIEBELDB..S_CONTACT (nolock)
```

This simple test would be run on the source and target systems. Each query should return the same quantity of information. If they do not, then a root cause analysis needs to take place as to why they are not. For example, inspecting the migration logs for duplicate records or other errors.

The Siebel Repository should also be checked. This would help confirm that the Siebel meta data is in sync with the SQL Server meta data. Examples of objects that may show up are indexes. Siebel has certain predefined indexes on base tables that it expects for performance reasons. During a mass data load, certain indexes may have been dropped. This post load validation test shows any indexes that are not present. Your Siebel TAM and Expert Services can verify this type of information with Siebel Tools and command line scripts.

Other post migration validation would be ensuring that work flow triggers are in place. It is common for work flow to be disabled during a migration. Database triggers typically do not need to be firing when just moving data via INSERTS and DELETES. This causes excessive CPU consumption and space utilization in the transaction log that is not needed.

Lastly, a quick post migration performance test of key screens and metrics to ensure that the migrated database reacts as it did during the testing phases. By running through these metrics, you are also "warming up" the database cache so performance is optimized already for the users when they first log into the system. This saves valuable time by caching pre-compiled execution plans for common queries and response time for fetching the data via logical reads from RAM instead of physical reads from the disk.

Performance Optimization of EIM for SQL Server

This section provides an overview of how to optimize Siebel's EIM data loading tool and process for Microsoft's SQL Server database. Please refer to the Microsoft Siebel Tech Champs website listed in the appendix for a more comprehensive discussion on this topic.

The basic premise for optimizing EIM is to find which queries are generating the most IO and/or taking up the most CPU. To do this, the DBA needs to look at the problem in a two fold manner: from the application down and from the database up.

The breadth of tools available to the DBA are numerous. Siebel provides valuable input from a logging capacity. SQL Server provides a rich set of debugging tools from the database. From the Siebel perspective, there are loading logs to review. Depending on the granularity of logging, the DBA can see the query and the timing results from it. This is helpful for locating obvious issues like long running queries, but also for more indirect information (i.e.: network latency that Profiler will not show). Looking at the issues from SQL Server, you have Profiler that shows the DBA query plans, IO, CPU ticks, etc., which aids in tracking down the problem. Then using Query Analyzer, debug and resolve them. The Figure 2-3 illustrates this concept.

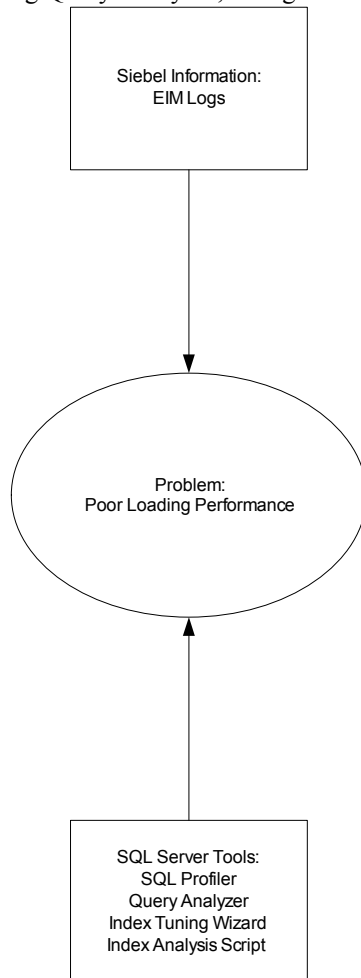


Figure 2-3: Caption TBS by author

A successful EIM load is greatly dependent on several variables. Papers written by both Microsoft and Siebel illustrate how to accomplish optimization of these processes.

When trying to optimize EIM it's often an iterative process. There usually isn't one problem to fix, but a variety. Often the problems are small in size, but the aggregate of each fix can be the difference between a successful load and migration, and one that is dogged by problems and delays.

The following is a breakdown of the broad areas for optimization of EIM:

- Hints. Hints are methods for hard coding a path to the data. In general, SQL Server's cost based optimizer provides superior results without the need for hints, yet EIM by default puts them in. Hints can cause extremely poor performance, and should be removed initially and only put back in as a last resort and proved with testing that they help. There are two parameters in the IFB file that control hints. The following are the parameters and the settings they should be set at:

```
USE INDEX HINTS = FALSE
```

```
USE ESSENTIAL HINTS = FALSE
```

- Loading too much meta data. By default, Siebel will load the whole repository even for the smallest EIM jobs. The symptom of this is that the initial startup of the EIM job (reading the repository) takes up more time than the actual run time of validating and inserting the data. Evidence of this can be seen by watching Profiler. At the front end of the job, S_COLUMNS and other Siebel base repository tables are read. Typically, a large cursor will be opened and run through prior to any EIM interface table queries. The remedy to this is to configure the IFB file to only load the minimum number of tables needed. Please work with your Siebel TAM or appropriate resource for this.
- Too many indexes on the interface tables. Siebel commonly puts a wide variety of indexes on the EIM interface tables. This is to accommodate the many types of EIM jobs that could possibly be run. Reality is that many indexes are not populated or used and need to be pruned off during the load. To figure out which indexes are not being used, load several thousand rows and process them. Then run the "Poor Index" script in the appendix on the interface table. It will show the DBA candidates to prune off.
- Incorrect indexes on both the base and interface tables. Just as there are too many indexes as in the previous bullet point, there are often incorrect indexes. Siebel's position is that they do not want you to drop indexes off the base tables without Expert Services involvement, but they do not have problems with the DBA adding indexes via Siebel Tools. Finding poor running queries is often as easy as turning on SQL Profiler and filtering on anything taking more than 10,000 reads. Look at the query in the text window, then match up the columns to the existing indexes. This can also be facilitated by cutting and pasting the query into Query Analyzer and looking at the graphical estimated query plan. Make a baseline using "set statistics io" (see BOL for more information). Add/change indexes on a test database and measure if the reads get better or worse. Then, after careful review and analysis, add the proper index to the base/interface table(s). In addition to putting more indexes on (note that adding indexes can also hurt performance, too) consider re-sequencing the columns of the index for greater selectivity (but remember to keep the batch number as the first column of the clustered index on the EIM interface table).
- Blocking. Blocking can occur through lock escalation while loading data in parallel with EIM. To isolate and monitor blocking in SQL Server, refer to KB article 271509 (<http://support.microsoft.com/default.aspx?scid=kb:en-us:271509>). To help eliminate blocking and lock escalation during the loading process, try using the system stored procedure "sp_index_option" (please read BOL for information on this stored procedure). Disable page level locking and enable only row level locking.
- Batch Size. Performance in EIM can also be controlled by the input batch size. Experiment with batch sizes from 5,000 to 20,000 rows. Upwards of 50,000 have been successful, too. Performance varies depending on level of complexity for the specific job type, disk configuration, and cpu of the database (among many other variables).

Migrating Using DATAEXP/DATAIMP

DATAEXP and DATAIMP provide a mechanism for bulk loading data in and out of Siebel in an expedited way. The usefulness for these tools are for moving lateral between platforms when the Siebel version is not changing (i.e. the customer is not doing an upgrade).

The following is an overview of the migration process when using the DATAEXP and DATAIMP tools to move from Oracle to SQL Server.

Migrating a Siebel Enterprise from Oracle to SQL Server

Assumptions

- Moving between the identical versions of the Siebel product (e.g. 7.5.2 on Oracle to 7.5.2 on SQL Server).
- Same version of the Siebel product supports both Oracle and SQL Server.
- Adequate hardware/disk on the target environment to support the database, concurrent workload for test purposes, etc.
- Not doing full-integration testing in the SQL Server based Siebel Enterprise. Any steps or requirements for this are in addition to what is identified here.

Export from Oracle

- Export the customized Repository using EXPREP.KSH. This will produce a single, binary file.
- Assess data volumes in Oracle. Understand the largest Tables - either by number of records or "size" (number of records * average record length). This is VERY important since it will drive how many DATAEXP processes are required.
- Build a list of non-Repository Tables using a SQL statement similar to the following (one Table per line, distinct function used there are multiple Repositories in the Oracle database, save the result set to a text file):

```
select distinct name
from siebel.s_table
where type <>'Repository'
order by 1 asc
```

- From this list, build multiple files/groupings of tables to support the desired number of DATAEXP streams. You want to save each "grouping" to a separate text file. The majority of the small Tables can reside in a single file (and hence a single DATAEXP stream). It is critical to not miss a Table from the "union" of all groupings, and to group accordingly. Do not include any of the S_DOCK* tables in any of the files. These are for Siebel remote and cannot be transferred to the SQL Server database.
- Copy the text files to the <siebsrvr>\bin folder.
- Start a DOS/command window. Navigate to <siebsrvr>\bin. Run SIEBENV.bat first to set environmental variables. Run DATAEXP to see the command-line switches.
- Use the /i switch along with the name of the desired text file (containing the set of Tables for that run of DATAEXP). I'm looking at a Siebel 7 environment while composing this email message, but expect to use the switches /u /p /c /d /f /i /l. Other switches are not necessary. Each DATAEXP process should write to a distinct "dump" file and a distinct log file.
- Check the log files for errors. Shouldn't be any, but...

Import into SQL Server

- Verify SQL Server installs patch levels (from the Siebel supported platform guide).
- Create the Siebel Enterprise. Install the File System, Gateway Server, Siebel Server, and Database Server (SQL Server database).
- Size the SQL Server so you have sufficient capacity.
- IMPREP.KSH to import the customized Repository.

- DDLSYNCH.KSH to synchronize the logical schema (Repository) with the physical schema (SQL Server Tables).
- Backup SQL Server database.
- Full compile of the Repository to produce a SRF with the customizations. Push this SRF to the Siebel Server and to any Client that will be used for testing.
- Delete the "Seed Data" inserted during INSTALL.KSH. A SQL statement similar to the following can be used to build the DELETE SQL statements:

```
select 'delete from siebel.' || name || ';'
from siebel.s_table
where type <> 'Repository'
order by 1 asc
```

- DATAIMP for each data file that you produced using DATAEXP. You do not need to specify a list of Tables for DATAIMP - it will know what Tables were unloaded into the binary file. It should be possible to run parallel DATAIMP processes (run each in a separate DOS/command window). Again, start a DOS/command window, navigate to <siebsrvr>\bin, run SIEBENV.bat, then run DATAIMP to see the command-line switches.
- Expect to use the DATAIMP switches /u /p /c /d /f /l. For performance, consider using /n with a larger value (say 500 or 1000).
- RUNSTATS on all Siebel Tables and all System Catalog Tables.
- Backup database.
- Check for fragmentation. Defragment accordingly.
- Backup of database.
- Test the functionality and customizations, test the referential integrity of the data, etc.

Possible failure points:

- Insufficient space.
- Column defined in Oracle and not defined in the Repository. In this situation DATAEXP will extract the data but DATAIMP will fail since the column will not exist in the corresponding Table in SQL Server. Resolution is to define the column in the Repository, run DDLSYNC.KSH again, and then run DATAIMP.

Customization Considerations

Additional considerations when migrating include how much customization has been done to the source system. This can impact how much time the project will take as well as the amount of effort involved.

Generally speaking, the less customization, the faster/easier the process will be.

Any customizations made to the repository (extension tables and columns, custom indexes, etc.) are handled via the Siebel-provided repository export/import scripts (such as exprep/impexp). However anything that affects the physical data model is an area that should be focused on during any database migration. The following are some general areas that require investigation and documentation before migrating:

- Docking Rules for Extension tables. Document the custom extension tables created in your repository. Ensure that these rules are still consistent before starting the migration process. In general these have been created by Siebel Expert Services Non Standard Change Request process.
- Row lengths for custom tables. Check the row lengths for custom tables prior to starting the migration. Refer to SQL Server documentation before migrating vendor specific datatypes (binary objects etc.).
- Object Names. SQL Server has a 128 character limit on object names. This may be an opportunity to rename objects when migrating from DB2/390 v6 or 7 (18 character limit on database objects including schemas (these are limited to eight characters), indexes, index specifications, user-defined functions (UDFs), user-defined types (UDTs), triggers, table spaces, stored procedures, node groups, and bufferpools)

Please carefully review these criteria prior to the migration. Success of the project is very dependent on careful planning and execution on the amount of customization involved. If in doubt, please engage with your Siebel TAM.

Migration Considerations and General Performance Characteristics

The following is a list of items to be considered during a migration:

- Migrate both the seed and customer data as one. Don't assume that the seed data from a fresh install will be the same as the seed data from the original.
- For performance reasons, make sure that Siebel logging, docking transaction, audit trail, workflow and triggers at the database level are disabled. Failure to do this can mean a degradation in the migration effort.
- As a general feeling for how long it will take to move data, the following numbers may provide some guidance:
- DATAIMP/DATAEXP took 31 hours for 120G of data moved in serial (non-parallel operations).
- INSERTS/SELECT SQL took 13 hours for 80GB in a serial migration.
- Consider the implications of proprietary data types and structures when moving the data. For example, Oracle sequences to Microsoft SQL Server identity columns.
- Acceptable date ranges in Oracle and SQL Server are different. Customers sometimes use date fields to store information other than dates. As a result date fields can contain dates prior to 01/01/1753. This will certainly cause Siebel tools to error out during DATAIMP into SQL Server. All dates prior to 01/01/1753 must be reset to 01/01/1753 or later.

To keep the database running efficiently in a post migration effort, remember to update the statistics and defragment the tables prior to going live.

Summary

The amount of effort required for a migration is greatly dependent on several factors: quantity of data, amount of customization, method of moving the data, performance and optimization of the loads, and if an upgrade is also involved. That said, the cost savings in the near term with licensing and hardware, combined with the long term savings on operational efficiencies and costs, more than offset the effort involved.

Microsoft and its partners are committed to the success of your migration and have many service offerings to help. Please feel free to reach out to the Microsoft-Siebel Global Alliance for any questions that you may have regarding the technical details involved.

The following stored procedure interrogates a Siebel database looking for indexes that are poor. For example, indexes that have 100% NULL values. Please use this procedure as a starting place and investigate fully before dropping the indexes. Work with Siebel Expert Services and your Siebel TAM prior to dropping indexes on base tables.

Additionally, the DBA can modify the script to look at specific tables, for example EIM tables only.

Glossary

Chapter 1 Abbreviations

CPU -Central Processing Unit

CRM -Customer Relationship Management

DBA - Database Administrator

EIM - Enterprise Information Manager

ES - Siebel Expert Services

HA -High Availability

MCS - Microsoft Consulting Services

MOM - Microsoft Operations Management

MTC - Microsoft Technology Centers

OS/DB - Operating System/Database

PSS - Microsoft Premier Support Services

SAN - Storage Area Network

SI - Systems Integrator

TAM - Technical Account Manager

TPC - Transaction Processing Performance Council

Chapter 2 Terms

EIM: A batch operated utility

PK/FK: Primary Key to Foreign Key relationship. Ensures integrity of the data in the database.

Post Load Validation: The process of confirming that the correct data has been moved from one location to another.

Siebel Repository: A Siebel repository is a collection of meta data that defines how the Siebel application will run. Initially it is a collection of base seed data. For example, definitions of columns, indexes and attributes, but will also contain information about customization, too.

Sp_indexoption: **SQL Server stored procedure to control row level locking.**

SQL Loader: Oracle's data loading tool.

Triggers: Pieces of code stored in the database that react to inserts, updates and deletes to a database table. The Siebel application uses database level triggers for workflow.

Appendix A: Siebel OS/DB Migration Example Project Plan

The following list of tasks represents a high level project plan that can be used as a template to begin developing your own migration project plan. The plan was developed in Microsoft Project and follows the MS Project conventional indentation for tasks and sub-tasks. The plan can be found on the publicly available Microsoft Technical Champs for Siebel:

<http://www.siebelonmicrosoft.com/mssiebel/Downloads/DOC/generic%20migration%20plan.mpp>

Table A-1: Sample Project Plan

Task ID	Task Name	Duration	Start Date	End Date	Resources
1	Siebel OS/DB Migration	48 days	1/5/2004	2/11/2004 17:00	
2	Decision to Migrate	1 day	1/5/2004	1/5/2004 17:00	
3	Establish Executive Sponsorship	1 day	1/5/2004 8:00	1/5/2004 17:00	
4	Select a System Integrator	1 day	1/5/2004 8:00	1/5/2004 17:00	
5	Notify and Involve Siebel Expert	1 day	1/5/2004 8:00	1/5/2004 17:00	
6	Include the Microsoft Siebel Alliance	1 day	1/5/2004 8:00	1/5/2004 17:00	
7	Discovery	11 days	1/6/2004	1/20/2004 17:00	2
8	Project team introductions and	1 day	1/6/2004 8:00	1/6/2004 17:00	
9	Existing System Analysis and	1 day	1/7/2004 8:00	1/7/2004 17:00	8
10	Timeline and migration/upgrade	2 days	1/8/2004 8:00	1/9/2004 17:00	9
11	Target Landscape planning	3 days	1/12/2004 8:00	1/14/2004 17:00	10
12	Hardware Sizing	3 days	1/15/2004 8:00	1/19/2004 17:00	11
13	Hardware approval and Order	1 day	1/20/2004 8:00	1/20/2004 17:00	12
14	Project planning	6 days	1/21/2004	1/28/2004 17:00	7
15	Siebel ES approval of	1 day	1/21/2004 8:00	1/21/2004 17:00	
16	Presentation of timeline/cost	0.5 days	1/22/2004 8:00	1/22/2004 12:00	15
17	Management approval	1 day	1/22/2004 13:00	1/23/2004 12:00	16
18	Project Kickoff	1 day	1/21/2004 8:00	1/21/2004 17:00	
19	Fine Tune Migration Project Plan	1 day	1/21/2004 8:00	1/21/2004 17:00	
20	Functional testing team	6 days	1/21/2004	1/28/2004 17:00	19SS
21	Establish Testing Team	2 days	1/21/2004 8:00	1/22/2004 17:00	
22	Define Testing Methodology	1 day	1/23/2004 8:00	1/23/2004 17:00	21
23	Testing Application approved and	1 day	1/26/2004 8:00	1/26/2004 17:00	22
24	Business and Systems	3 days	1/26/2004	1/27/2004 17:00	
25	Establish Business Objectives	1 day	1/26/2004 8:00	1/26/2004 17:00	22
26	Fine tune or reuse existing	1 day	1/27/2004 8:00	1/27/2004 17:00	25
27	Test Plan Development	1 day	1/28/2004 8:00	1/28/2004 17:00	24

28			Iterative Practice Migrations	20 days	1/5/2004	2/24/2004 15:40	
29			Receive New Hardware	1 day	1/5/2004 8:00	1/5/2004 17:00	
30			Backup Customer Production DB	0.33 days	1/6/2004 8:00	1/6/2004 10:40	29
31			Configure SQL Server in Test	1 day	1/6/2004 8:00	1/6/2004 17:00	29
32			Export Source DB	7 days	1/6/2004	1/15/2004 10:40	
33			Export customized repository	1 day	1/6/2004 10:40	1/7/2004 10:40	30
34			Assess data volumes in Source	1 day	1/7/2004 10:40	1/8/2004 10:40	33
35			Build a list of non-repository	1 day	1/8/2004 10:40	1/9/2004 10:40	34
36			build multiple files/groupings of	1 day	1/9/2004 10:40	1/12/2004 10:40	35
37			convert the text files to the	1 day	1/12/2004 10:40	1/13/2004 10:40	36
38			start cmd window run siebenv.bat	1 day	1/13/2004 10:40	1/14/2004 10:40	37
39			check the log files for errors	1 day	1/14/2004 10:40	1/15/2004 10:40	38
40			Import to SQL Server	13 days	1/15/2004	2/3/2004 10:40	39-51
41			Verify SQL Server installation	1 day	1/15/2004 10:40	1/16/2004 10:40	31
42			create the Siebel Enterprise. file	1 day	1/16/2004 10:40	1/19/2004 10:40	41
43			verify the disk layout and sizes of	1 day	1/19/2004 10:40	1/20/2004 10:40	42
44			IMPREP.KSH to import the	1 day	1/20/2004 10:40	1/21/2004 10:40	43
45			DDLSYNCH.KSH to synchronize	1 day	1/21/2004 10:40	1/22/2004 10:40	44
46			Backup the SQL DB	1 day	1/22/2004 10:40	1/23/2004 10:40	45
47			Full compile of repository to	1 day	1/23/2004 10:40	1/26/2004 10:40	46
48			delete the "seed data" inserted	1 day	1/26/2004 10:40	1/27/2004 10:40	47
49			DATAIMP for each data file that	1 day	1/27/2004 10:40	1/28/2004 10:40	48
50			RUNSTATS on all Siebel Tables	1 day	1/28/2004 10:40	1/29/2004 10:40	49
51			backup database	1 day	1/29/2004 10:40	1/30/2004 10:40	50
52			check for fragmentation	1 day	1/30/2004 10:40	2/2/2004 10:40	51
53			check basic functionality	1 day	2/2/2004 10:40	2/3/2004 10:40	52
54			Iterate To Optimize Development	5 days	2/3/2004	2/10/2004 10:40	53-57
55			Defect Testing	4 days	2/3/2004	2/9/2004 10:40	
56			Run Load Runner Scripts to	1 day	2/3/2004 10:40	2/4/2004 10:40	27
57			Run Load Runner Scripts to	1 day	2/4/2004 10:40	2/5/2004 10:40	27.53.56
58			Compare performance on	1 day	2/5/2004 10:40	2/6/2004 10:40	57
59			Compare size of Source DB	1 day	2/5/2004 10:40	2/6/2004 10:40	57
60			Isolate issues and resolve.	1 day	2/6/2004 10:40	2/9/2004 10:40	59
61			Optimization of the	1 day	2/9/2004	2/10/2004 10:40	55
62			Try different approaches to	1 day	2/9/2004 10:40	2/10/2004 10:40	55
63			DTS	1 day	2/9/2004 10:40	2/10/2004 10:40	55
64			siebimp/exp	1 day	2/9/2004 10:40	2/10/2004 10:40	55
65			Analyze whether a "tool" can	1 day	2/9/2004 10:40	2/10/2004 10:40	55

66				Finalize the absolute best	1 day	2/9/2004 10:40	2/10/2004 10:40	55
67				Unicode Testing	4 days	2/2/2004	2/9/2004 10:40	
68				Try migrating to a SQL Server	1 day	2/3/2004 10:40	2/4/2004 10:40	39.31
69				Measure the differences in	1 day	2/4/2004 10:40	2/5/2004 10:40	68
70				Re-Run base lines to see	1 day	2/5/2004 10:40	2/6/2004 10:40	69
71				Source DB -> SQL Server	1 day	2/6/2004 10:40	2/9/2004 10:40	70
72				Document the procedure	1 day	2/10/2004 10:40	2/11/2004 10:40	54
73				Adjust the Project Plan Based on	1 day	2/10/2004 10:40	2/11/2004 10:40	54
74				Present new project plans and	1 day	2/11/2004 10:40	2/12/2004 10:40	73
75				Customer Site Migration	4 days	2/12/2004	2/18/2004 10:40	74
76				First Migration test run based on	3 days	2/12/2004 10:40	2/17/2004 10:40	74
77				Functional Testing of Migrated	1 day	2/17/2004 10:40	2/18/2004 10:40	27.76
78				Customer Site Migration	4.5 days	2/18/2004	2/24/2004 15:40	75
79				Restore SQL Server Clean install	0.5 days	2/18/2004 10:40	2/18/2004 15:40	75
80				Migrate Source DB	1 day	2/18/2004 15:40	2/19/2004 15:40	79
81				Functional Testing of Migrated	2 days	2/19/2004 15:40	2/23/2004 15:40	80.27
82				Documentation and procedural	1 day	2/23/2004 15:40	2/24/2004 15:40	81
83				Migrate Environments	8-17 days	2/24/2004	2/26/2004 7:40	82
84				Migrate Development System	1 day	2/24/2004 15:40	2/25/2004 15:40	82
85				Migrate Quality Assurance System	1 day	2/25/2004 15:40	2/26/2004 15:40	84
86				Production Migration Contingency	1 day	2/26/2004 15:40	2/27/2004 15:40	85
87				Production Migration testing with	2 days	2/27/2004 15:40	3/2/2004 15:40	86.82
88				Final planning and go live strategy	1 day	3/2/2004 15:40	3/3/2004 15:40	87
89				Production Readiness	1 day	2/2/2004	2/1/2004 15:40	88
90				Operations Review	1 day	3/3/2004 15:40	3/4/2004 15:40	88
91				High Availability Review	1 day	3/3/2004 15:40	3/4/2004 15:40	88
92				Customer Production	4-17 days	2/1/2004	2/26/2004 7:40	89
93				Prepare new SQL Server	1 day	3/4/2004 15:40	3/5/2004 15:40	91
94				shutdown Siebel current	1 day	3/5/2004 15:40	3/5/2004 23:40	93.89
95				Migrate production	1 day	3/5/2004 23:40	3/6/2004 7:40	94
96				Post Migration Validation	2 days	2/26/2004	2/10/2004 17:00	95
97				Functional Test	1 day	3/6/2004 7:40	3/6/2004 15:40	95
98				Customer support team sign off	1 day	3/8/2004 8:00	3/8/2004 17:00	97
99				Customer Management Sign Off	1 day	3/9/2004 8:00	3/9/2004 17:00	98
100				Release Production to Users	1 day	3/10/2004 8:00	3/10/2004 17:00	99
101				Performance Tuning (ongoing)	1 day	3/11/2004 8:00	3/11/2004 17:00	100

Appendix B: Websites for Additional Information

The links in this appendix provide a wealth of additional information to assist with your OS/DB migration project. From SQL Server training to finding System Integrators, these links should compliment the data gathering phase of your project and insure that you have the information you need to proceed with confidence.

General Microsoft and Siebel Resources

Microsoft Technical Champs for Siebel

<http://siebelonmicrosoft.com>

Microsoft | Siebel Alliance Information

<http://www.siebel-microsoft.com/>

Siebel Support Web

<https://ebusiness.siebel.com/supportweb/>

Microsoft Press Pass Siebel eBusiness Applications and Microsoft .Net Technology

<http://www.microsoft.com/presspass/press/2003/oct03/10-07MSFTSiebelPR.asp>

Companies Switching to SQL Server: Evaluate and Compare

<http://www.microsoft.com/sql/evaluation/compare/switchers.asp>

Microsoft | Siebel Global Alliance:

<http://www.siebel-microsoft.com>

Siebel Systems, Inc.

<http://www.siebel.com/>

Microsoft Classes

SQL Server Training and Events Web Site

<http://www.microsoft.com/sql/techinfo/training/default.asp>

SQL Server Resources

Microsoft Press Pass Siebel eBusiness Applications and Microsoft .Net Technology

<http://www.microsoft.com/presspass/press/2003/oct03/10-07MSFTSiebelPR.asp>

Technical Champs SQL Server Technology Section

<http://www.siebelonmicrosoft.com/mssiebel/Technology/SqlServer.aspx>

Technical Champs OS/DB Migration Technology Section

<http://www.siebelonmicrosoft.com/mssiebel/Technology/OSDB.aspx>

Migrating Oracle Databases to SQL Server 2000

<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/sql/deploy/upgrdmigrate/sqlorcle.asp>

<http://www.microsoft.com/resources/documentation/sql/2000/all/reskit/en-us/part2/c0761.msp>

System Integrator Resources

Microsoft Technical Champs for Siebel Resources

<http://www.siebelonmicrosoft.com/mssiebel/Partners.aspx>

Accenture Siebel Alliance

http://www.accenture.com/xd/xd.asp?it=enweb&xd=services%5Ccrm%5Ccrm_alliances_siebel.xml

Accenture Unix to Windows Migration and Unified Desktop Tools

http://www.accenture.com/xd/xd.asp?it=enweb&xd=services%5Cmicrosoft%5Cmicr_unix_migr.xml

Appendix C: Sample Code

Chapter 2

The following stored procedure interrogates a Siebel database looking for indexes that are poor. For example, indexes that have 100% NULL values. Please use this procedure as a starting place and investigate fully before dropping the indexes. Work with Siebel Expert Services and your Siebel TAM prior to dropping indexes on base tables.

Additionally, the DBA can modify the script to look at specific tables, for example EIM tables only.

```
-- to run, type:
-- exec poor_index
-- go

create proc poor_index
as
begin
set nocount on

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[#temp_tbl_ind]') and OBJECTPROPERTY(id, N'IsTable') = 1)
drop table #temp_tbl_ind

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[
#temp_idx]') and OBJECTPROPERTY(id, N'IsTable') = 1)
drop table #temp_idx

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[#temp_col]')
and OBJECTPROPERTY(id, N'IsTable') = 1)
drop table #temp_col

-- get the top 10 tables and indexes from the largest tables

select
top 100
id, indid, dpages, rows, name
into #temp_tbl_ind
from sysindexes (nolock)
where
-- indid = 0 or
-- indid = 1
(status & 64=0) and
indid < 255 and
id > 99 -- and
--rows > 1000
order by dpages desc, id desc, indid
```

```

-- next, get just the tables

select id, indid, name
into #temp_idx
from #temp_tbl_ind
where indid > 0
order by id, indid

-- then roll through the indexes and get the column name for the first column

SELECT id, indid, index_col(object_name(id),indid,1) 'col_name'
into #temp_col
from #temp_idx

if exists(select 1 from sysobjects where name like 'dist_vals')
begin
    drop table dist_vals
end

create table dist_vals(tbl sysname, indid int, first_col sysname, uniq bigint,
rows bigint)

if exists(select 1 from sysobjects where name like 'qry')
begin
    drop table qry
end

create table qry(sql_txt varchar(500))

-- build the query to get the data on the usefulness of the index first column

insert into qry
select 'insert into dist_vals select '+char(39)+object_name(id)+char(39)+', '
+char(39)+cast(indid as varchar(3))+char(39)+
', '+char(39)+col_name+char(39)+
', count(distinct(isnull('+col_name+',1))), '+
'count(isnull('+col_name+',1)) from [' +object_name(id)+'] (nolock)'-- +char(13)+'
'+char(13)
from #temp_col

-- select * from qry

truncate table dist_vals

```

Appendix C: Sample Code

```
-- run the queries and log the data

declare @sql_txt nvarchar(500)

DECLARE sql_cursor CURSOR FOR
SELECT sql_txt
FROM qry

OPEN sql_cursor

FETCH NEXT FROM sql_cursor
INTO @sql_txt

WHILE @@FETCH_STATUS = 0
BEGIN

--                                     select @sql_txt
                                     exec sp_executesql @sql_txt

                                     FETCH NEXT FROM sql_cursor
                                     INTO @sql_txt

END

CLOSE sql_cursor
DEALLOCATE sql_cursor

-- analyze the results

select 'poor index',tbl 'table_name',name 'index_name', first_col, uniq
'unique_values_in_column',a.rows
'rows_in_table',convert(numeric,dpages*8*1024)'bytes_used'
from dist_vals a, sysindexes b
where
(a.uniq=1 or a.uniq=2) and
object_id(tbl)=b.id and
a.indid=b.indid and
a.rows > 0

end
```

Blocking Script from KB article 271509

<http://support.microsoft.com/default.aspx?scid=kb;en-us;271509>

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Appendix C: Sample Code

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2001 Microsoft Corporation. All rights reserved.

Microsoft, Windows, and SQL Server 2000 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.