PLANNING AND EXECUTING A SUCCESSFUL SQL SERVER DATABASE MIGRATION
Presentation Topics

➤ Typical SQL Server Migration Plan

➤ Key Deliverables

➤ Database Migration Steps

➤ Brief Overview of SQL Server Migration Assistant Tool (SSMA)

➤ Common Problems in Database Migrations
What is NOT Covered

➤ Detailed SSMA Tutorial

➤ DBMS Specific Migration Information

➤ How to Build Target SQL Server System
Goals

By the end of the presentation you will:

➤ Understand at a high level how to migrate a database to SQL Server

➤ Know how to avoid surprises by better planning

➤ Leverage SSMA as both a planning and a migration tool
 Presenter Info

➤ 20 years of experience in database development, administration, and project management

➤ B.S. in Management of Information Systems – USF

➤ Dobler Consulting lead of DBA Managed Services practice
SQL Server Migration Challenges

➤ All database migration projects are unique
  ➤ Differences in DBMS, configurations, SQL code, front-end applications, third-party tools, etc.

➤ Not all database system components can be migrated
  ➤ In most cases, logins cannot be migrated
  ➤ Third-party applications may not be compatible with SQL Server
  ➤ Embedded and dynamic SQL code has to be migrated manually
  ➤ Applications may require new drivers/connectors

➤ Scope of database migration project can be quite large
  ➤ Migration effort can include thousands of database objects and lines of code
  ➤ Not enough time to perform full regression/performance testing
Steps for Overcoming Challenges

➤ Thoroughly analyze existing database system
  ➤ Identify all components that must exist in target SQL Server environment

➤ Develop an approach for each component that will require manual migration
  ➤ Consider grouping components by categories that will implement the same approach (i.e. replace current transactional replication and failover solutions with AlwaysOn).
  ➤ Evaluate risks and contingency options

➤ Include adequate time for migration testing
  ➤ Plan to execute your database migration as many times as possible in your test environment
  ➤ Develop test approach to minimize risks
  ➤ Practice makes perfect!
Typical SQL Server Migration Plan

I. Analysis
II. Planning
III. Build
IV. Migrate, Test, Fix, Repeat
V. Deployment
VI. Support
Analysis Deliverables

➤ System Architecture Diagrams
  ➤ Map all components of AS-IS and TO-BE systems

➤ System Configurations
  ➤ Operating System, DBMS, Applications and Third Party tool versions
  ➤ For database, look for Collation, ANSI defaults, Quoted Identifier settings

➤ Hardware Specifications
  ➤ Identify server and storage requirements for target system
  ➤ Unless you are migrating to an existing SQL Server, you will need to order new hardware early in the project
Planning Deliverables

➤ Environment Plan
  ➤ Outline of environments to be used in migration project
  ➤ States what components from AS-IS and TO-BE systems will be included
  ➤ Document who will use the environment and when

➤ Project Schedule
  ➤ Leverage SSMA Migration Reports for estimating effort
Planning Deliverables

➤ Change/Source Code Version Management Plan
  ➤ Change happens, and you better be ready for it
  ➤ Develop a strategy to incorporate changes to source system without disrupting migration schedule
  ➤ Create new source code repository for migrated database objects

➤ Production Contingency/Back out Plan
  ➤ Get management buy-in early in the project
  ➤ Agree on checkpoints and who makes GO/NO-GO decisions
  ➤ Agree on criteria to implement contingency plan
Intro to SSMA Tool

➤ SQL Server Migration Assistant Overview
  ➤ Provides detailed migration assessment report with effort estimates
  ➤ Automates schema and data migration
  ➤ Available for five DBMS: Oracle, Sybase, MySQL, Access, and DB2
  ➤ Capable of migrating source DBMS to SQL Server 2005 or higher and to SQL Azure
SSMA Configuration
SSMA Configuration

Please select a migration project type in the 'Migration Target Version' drop down box.
# SSMA Configuration

### Default Project Settings

**Migration Target Version:** SQL Server 2014

<table>
<thead>
<tr>
<th>Mode</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conversion</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Migration</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Collecting Data</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Conversion

- @@ERROR global variable: Convert and mark with warning
- Dynamic SQL: Simple conversion
- Equality check conversion: Simple conversion
- Insert an explicit value into a timestamp column: Exclude column
- Proxy table conversion: Mark with error
- RAISERROR base message number: 30001
- Store temporary objects defined in procedures: Yes
- System objects: Convert and mark with warning
- Unresolved identifiers: Convert and mark with warning

### String expressions

- Concatenation of NULL: Keep current syntax
- Conversion of Empty Strings: Keep current syntax
- Conversion of LIKE operator: Keep current syntax
- CONVERT and CAST binary string conversion: Convert and mark with warning
- CONVERT or CAST empty strings to numeric types: Simple conversion
- Format strings: Create new string

### System functions

- CHARINDEX function: Keep current syntax
- DATALENGTH function: Keep current syntax
- INDEX function: Keep current syntax

### Misc

- OK
- Apply
- Cancel
SSMA Configuration

Default Project Settings

- Migration Target Version: SQL Server 2014

Mode: Default

- Conversion
  - Migration
  - Collecting Data

**Misc**
- @@ERROR global variable: Convert and mark with warning
- Dynamic SQL: Convert and mark with warning
- Equality check conversion: Simple conversion
- Insert an explicit value into a timestamp column: Exclude column
- Proxy table conversion: Mark with error
- RAISERROR base message number: 30001
- Store temporary objects defined in procedures: Yes
- System objects: Convert and mark with warning
- Unresolved identifiers: Convert and mark with warning

**String expressions**
- Concatenation of NULL: Keep current syntax
- Conversion of Empty Strings: Keep current syntax
- Conversion of LIKE operator: Simple conversion
- CONVERT and CAST binary string conversion: Cast to fixed length
- CONVERT or CAST empty strings to numeric types: Simple conversion
- Format strings: Create new string

**System functions**
- CHARINDEX function: Keep current syntax
- DATALength function: Keep current syntax
- IND$IC$ function: Keep current syntax

**Conversion of LIKE operator**
Specifies whether to convert LIKE operands to match Sybase ASE behavior. If the option is "Cast to fixed length" and LIKE pattern is \\
"%" with leading spaces, the left hand operand will be cast to the char or nchar data type.
# SSMA Configuration

## Default Project Settings

**Migration Target Version:** SQL Server 2014

### Mode: Default

#### Dates Correction
- **Replace unsupported dates**: Do nothing

#### Migration Engine
- **Migration Engine**: Client Side Data Migration Engine

#### Misc
- **Batch size**: 10000
- **Check constraints**: False
- **Data migration timeout**: 15
- **Extended data migration options**: Hide
- **Fire triggers**: False
- **Keep identity**: True
- **Keep nulls**: True
- **On Error**: Proceed to next batch
- **Round fractional part of numbers**: No
- **Sybase Unicode Endian**: False
- **Table lock**: True
- **Use cursors**: False

### Parallel Data Migration
- **Parallel data migration mode**: Auto

### Migration Engine
- **Migration Engine used during data migration**
SSMA Configuration

Default Project Settings

Migration Target Version: SQL Server 2014

Reset to Default

Synchronization

General

Synchronization

GUI

Type Mapping

Misc

Attempts

Synchronization for SQL Server
- Refresh local object on local and remote object change: Write to Database
- Refresh local object on local object change: Write to Database
- Refresh local object on remote object change: Refresh from Database
- Refresh when local object metadata is missing: Refresh from Database

Misc
### SSMA Configuration

**Type Mapping**

<table>
<thead>
<tr>
<th>Source Type</th>
<th>Target Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real[16..]</td>
<td>Real[50]</td>
</tr>
<tr>
<td>image</td>
<td>image</td>
</tr>
<tr>
<td>integer</td>
<td>int</td>
</tr>
<tr>
<td>longysename</td>
<td>nvarchar[255]</td>
</tr>
<tr>
<td>money</td>
<td>money</td>
</tr>
<tr>
<td>national char</td>
<td>nchar</td>
</tr>
<tr>
<td>national char varying</td>
<td>nvarchar</td>
</tr>
<tr>
<td>national char varying[.4000]</td>
<td>nvarchar[*]</td>
</tr>
<tr>
<td>national char varying[4001..]</td>
<td>nvarchar(max)</td>
</tr>
<tr>
<td>national char*[.4000]</td>
<td>nchar[*]</td>
</tr>
<tr>
<td>national char[4001..]</td>
<td>nvarchar(max)</td>
</tr>
<tr>
<td>national character</td>
<td>nchar</td>
</tr>
<tr>
<td>national character varying</td>
<td>nvarchar</td>
</tr>
<tr>
<td>national character varying[.4000]</td>
<td>nvarchar[*]</td>
</tr>
<tr>
<td>national character varying[4001..]</td>
<td>nvarchar(max)</td>
</tr>
<tr>
<td>national character[.4000]</td>
<td>nchar[*]</td>
</tr>
</tbody>
</table>

*Deprecated datatype*
Verify Drivers

SSMA for Oracle Setup

Required Component(s) Missing
Oracle client software not found.

Setup is unable to find Oracle client software. You can install it from Oracle product media or download it from Oracle web site.
Click on the Download button to navigate to Oracle website to install Oracle client software.
Click on the Next button to install later.
Verify Drivers

The selected provider is incompatible with the installed version of SSMA for MySQL. This warning may be a result of running SSMA as 64-bit application while having only 32-bit connectivity components installed or vice versa. You can run 32-bit SSMA application if you have 32-bit connectivity components or 64-bit SSMA application if you have 64-bit connectivity components. Shortcut to both 32-bit and 64-bit SSMA can be found under the Programs menu.
Verify Drivers

Access Object Collector error: Database

Retrieving the COM class factory for component with CLSID {CD7791B-49FD-11C0-81E1-8D05F11F0415} failed due to the following error: 80040154 Class not registered (Exception from HRESULT: 0x80040154 (REGDB_E_CLASSNOTREG)). This error may be a result of running $SMA$ as 64-bit application while having only 32-bit connectivity components installed or vice versa. You can run 32-bit $SMA$ application if you have 32-bit connectivity components or 64-bit $SMA$ application if you have 64-bit connectivity components. Shortcut to both 32-bit and 64-bit $SMA$ can be found under the Programs menu. You can also consider updating your connectivity components from http://go.microsoft.com/fwlink/?LinkId=197802.

An error occurred while loading database content.
SSMA as a Planning Tool

➤ SSMA features a Database Migration Assessment Report
   ➤ The report lists all database objects that will require manual effort for conversion
   ➤ Indicates the reason objects cannot be migrated automatically
   ➤ Provides rough estimates on manual effort

➤ Perform a test conversion to uncover additional issues
   ➤ SSMA may not catch all possible migration issues
   ➤ Errors may appear when loading migrated objects to SQL Server (i.e. improper use of quoted identifiers, computed columns, etc.)
   ➤ Do a test conversion to get a full picture of migration assessment
Right click on database
SSMA Migration Assessment Report

Microsoft SQL Server Migration Assistant 6.0

Click error to show code
How to Access Saved Report

Open with Excel
How to Access Saved Report

![Excel Image]

- Click the first red arrow to access saved report.
- Click the second red arrow to open the report.
- Click the third red arrow to customize the report settings.
How to Access Saved Report

Develop approach for each error
Develop Migration Scripts

- Develop scripts to execute schema migration
  - Complete manual object migration in SQL Server
  - Generate full set of scripts after SQL code is complete

- Recommended scripts:
  - Create database scripts
  - Create tables, views, functions, stored procedures
  - Create triggers
  - Create clustered indexes, primary key constraints
  - Create non-clustered indexes
  - Create foreign key constraints
  - Create logins, roles, permissions
  - Post migration scripts (i.e. computed columns)
SSMA Schema Migration

Right click on database
SSMA Schema Migration

Check error list
SSMA Schema Migration

Apply changes to SQL Server
SSMA Schema Migration

Synchronize with the Database

- Database
  - [Not Found]
  - [Not Found]

- Action
  - [Not Found]
  - [Not Found]
  - [Not Found]
  - [Not Found]

- Local Metadata
  - sakila
  - Schemas
  - dbo
    - Datatypes
    - Functions
    - Procedures
    - Sequences
    - Tables
    - Views

41 Object(s) Changed Locally | 22 Object(s) Changed in Database | 0 Object(s) Not Changed | 0 Object(s) Changed in Database and Locally
SSMA Schema Migration

Check error list
Data Migration Options

➤ Easiest option is to use SSMA to perform data migration
  ➤ Test performance to determine best drivers to use

➤ For some DBMS source systems, SSMA offers Client-Side or Server-Side migration
  ➤ Client-Side migration typically offers more driver options, data migration occurs on computer running SSMA
  ➤ Client-Side migration not recommended if running SSMA remotely
  ➤ Server-Side migration runs data migration on target database server, but typically is more restrictive on driver options
  ➤ Test data migration the same way you would run it on Production!
SSMA Data Migration
SSMA Data Migration

### Data Migration Report

<table>
<thead>
<tr>
<th>Status</th>
<th>From</th>
<th>To</th>
<th>Total Rows</th>
<th>Migrated Rows</th>
<th>Success Rate</th>
<th>Duration (DD:HH:MM:SS:MS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sakila.actor</td>
<td>sakila.dbo.actor</td>
<td>200</td>
<td>200</td>
<td>100.00 %</td>
<td>00:00:00:02:852</td>
</tr>
<tr>
<td></td>
<td>sakila.address</td>
<td>sakila.dbo.address</td>
<td>603</td>
<td>0</td>
<td>0.00 %</td>
<td>00:00:00:09:521</td>
</tr>
<tr>
<td></td>
<td>sakila.category</td>
<td>sakila.dbo.category</td>
<td>16</td>
<td>16</td>
<td>100.00 %</td>
<td>00:00:00:02:755</td>
</tr>
<tr>
<td></td>
<td>sakila.city</td>
<td>sakila.dbo.city</td>
<td>600</td>
<td>600</td>
<td>100.00 %</td>
<td>00:00:00:02:872</td>
</tr>
<tr>
<td></td>
<td>sakila.country</td>
<td>sakila.dbo.country</td>
<td>109</td>
<td>109</td>
<td>100.00 %</td>
<td>00:00:00:02:752</td>
</tr>
<tr>
<td></td>
<td>sakila.customer</td>
<td>sakila.dbo.customer</td>
<td>599</td>
<td>599</td>
<td>100.00 %</td>
<td>00:00:00:02:773</td>
</tr>
<tr>
<td></td>
<td>sakila.film</td>
<td>sakila.dbo.film</td>
<td>1000</td>
<td>1000</td>
<td>100.00 %</td>
<td>00:00:00:03:337</td>
</tr>
<tr>
<td></td>
<td>sakila.film_actor</td>
<td>sakila.dbo.film_actor</td>
<td>5462</td>
<td>5462</td>
<td>100.00 %</td>
<td>00:00:00:03:388</td>
</tr>
<tr>
<td></td>
<td>sakila.film_category</td>
<td>sakila.dbo.film_category</td>
<td>1000</td>
<td>1000</td>
<td>100.00 %</td>
<td>00:00:00:03:188</td>
</tr>
<tr>
<td></td>
<td>sakila.film_text</td>
<td>sakila.dbo.film_text</td>
<td>1000</td>
<td>1000</td>
<td>100.00 %</td>
<td>00:00:00:02:945</td>
</tr>
<tr>
<td></td>
<td>sakila.inventory</td>
<td>sakila.dbo.inventory</td>
<td>4581</td>
<td>4581</td>
<td>100.00 %</td>
<td>00:00:00:01:111</td>
</tr>
<tr>
<td></td>
<td>sakila.language</td>
<td>sakila.dbo.language</td>
<td>6</td>
<td>6</td>
<td>100.00 %</td>
<td>00:00:00:03:393</td>
</tr>
<tr>
<td></td>
<td>sakila.payment</td>
<td>sakila.dbo.payment</td>
<td>16049</td>
<td>16049</td>
<td>100.00 %</td>
<td>00:00:00:03:504</td>
</tr>
<tr>
<td></td>
<td>sakila.rental</td>
<td>sakila.dbo.rental</td>
<td>16044</td>
<td>16044</td>
<td>100.00 %</td>
<td></td>
</tr>
</tbody>
</table>

**Check error list**
SSMA Data Migration

Check error list
Testing Converted Database Objects

➤ Develop a test approach focused on minimizing risk
  ➤ Full regression testing may be impractical on large systems
  ➤ Recommended approach is to unit test each database object
  ➤ Run each object on source and target system and compare output
  ➤ For objects where output does not match, perform additional testing and fix as needed
  ➤ Consider production parallel testing if possible

➤ Schedule UAT and Performance testing on target Production hardware
  ➤ Baseline performance of source system for critical functions and compare timings to target system
  ➤ Before final deployment, perform user acceptance on target Production hardware
Test Database Migration Cycles

➤ Schedule time to run database migration steps multiple time
  ➤ At a minimum, run through the database migration steps when setting up the test, QA test, and UAT environments
  ➤ Additional test migrations can be scheduled when refreshing test environments and when doing code drops
  ➤ Executing the migration steps multiple times provides more opportunities to uncover issues and to fine tune the process
Final Deployment

➤ Schedule a change moratorium prior to the final deployment
  ➤ Allows for migration scripts to be in a stable state prior to deployment
  ➤ Schedule one final test migration with the finalized scripts
  ➤ Any emergency fixes to source system can be applied to target system after final deployment
  ➤ All enhancements and non-critical fixes should be worked on new system

➤ Develop deployment plan
  ➤ Include contact information and hours when people need to be available
  ➤ Migration steps can take a long time, so schedule checkpoint meetings where people can get status information
  ➤ Consider having a dedicated person to coordinate communications
Contingency Plans

➤ Have a detailed plan ready to enable source system
  ➤ Preserve the state of source system to ensure back out plan is successful
  ➤ Do not make changes on SQL Server that will make back out plan harder
  ➤ Determine in advance the criteria for back out and who is responsible for making the decision
  ➤ Plan your final deployment with a back out plan in mind (do not make changes that will make backing out more difficult)

➤ Plan for an extended post-production support
  ➤ Some issues may arise days after the final deployment
  ➤ Plan to set aside time for support team to address any emergency fixes
Decommission Source System

➤ Before decommissioning source system consider:
  ➤ Script all database objects and archive in source code repository
  ➤ Create backups and determine retention plan
  ➤ Consider having a system available where source system can be restored if needed
QUESTIONS?
THANK YOU!