Loading a Data Warehouse Using SSIS

- Why SSIS for data warehouse loads?
- SSIS DW design principles
- Data warehouse load design patterns
- Change detection in SSIS
- Automating development

About the Presenter

Tim Mitchell

- Data architect
- Consultant
- Microsoft Data Platform MVP
- Dallas, Texas area
- TimMitchell.net
- @Tim_Mitchell

Tyleris Data Solutions

- Data architecture
- Business intelligence
- Data warehousing
- ETL
- Analytics
- Tyleris.com

WHY SSIS FOR DATA WAREHOUSE LOADS?

Why SSIS for Data Warehouse Loads?

Ease of use

- Easy-to-use UI
- Simple tasks are simple
- Difficult tasks are possible
- Built-in tools such as logging and error recovery

Why SSIS for Data Warehouse Loads?

Capability

- Handles most any ETL demand, not just DW workloads
- Well-performing

Why SSIS for Data Warehouse Loads?

Flexibility

- Built-in connectors for most common data endpoints
- Customizable through scripting
- Orchestrate T-SQL (ELT) or native SSIS data flows (ETL)

Why SSIS for Data Warehouse Loads?

Modularity

- Supports write-once, use-many
- Don't rewrite code!
- SSIS parent-child patterns

SSIS DATA WAREHOUSE DESIGN PRINCIPLES

SSIS DW Design Principles

Limit package size

- Smaller, single-operation packages
- Easier reusability
- Simpler to edit, test
- My rule of thumb: one package per table load

SSIS DW Design Principles

Use an orchestrator package

- Coordinate dependendencies and parallelism
- More options than configuring as SQL Agent job steps

SSIS DW Design Principles

Externalize changeable values

- If it can change, it will
  - Connection strings
  - File paths
- Easier to reconfigure
- Easier to move from one environment to the next
- Easier to automate tests

SSIS DW Design Principles

Consider where the data resides

- Same server = more direct T-SQL options
- Different servers = SSIS data flow, or stage
- This should be a key consideration in any package design

SSIS DW Design Principles

Consider what happens upon failure

- Do nothing
- Roll back the entire operation
- Handle errant data on a row-by-row basis
- Triage

SSIS DW Design Principles

Tracking data lineage

- "*Where did this data come from?*"
- Row-level lineage ID helps answer this
- Use SSIS execution ID as lineage ID
- Ties in with SSIS logging tables

SSIS DW Design Principles

Remember that these are business decisions

- Granularity
- Frequency
- Historical tracking type
- Lineage
- Row-level error handling (if any)

DATA WAREHOUSE LOAD DESIGN PATTERNS

Data Warehouse Load Design Patterns

Fact loads

- Typically insert only (no update)
- Sometimes, updates are required
  - Slowly changing fact table

Data Warehouse Load Design Patterns

Fact loads

- SSIS data flow
  - When source and destination are (or may someday be) on different instances
  - When moving between nonrelational and relational
  - When no updates are required (insert only)

Data Warehouse Load Design Patterns

Dimension loads

- Type 0: insert only
- Type 1: inserts and in-place updates
- Type 2: inserts and historical changes through date/time versioning
- Other variations on these

Data Warehouse Load Design Patterns

Dimension loads

- Direct data flow load
- Slowly changing dimension wizard (!!)

Data Warehouse Load Design Patterns

Dimension loads

- Staged load (ELT)
  - Load to staging server on destination
  - T-SQL to perform insert, update

CHANGE DETECTION

Change Detection

- Find new, updated, and deleted rows in the source
- Change detection is a critical part of any data warehouse load

Change Detection

Why change detection?

- Performance
- Historical record

Change Detection

Source change detection

- Filters out the rows brought into ETL pipeline
- Useful when there is a solid method for detecting new, added, and deleted rows on the source

Change Detection

Source change detection

- Usually the better option when available
- Technical assumptions

Change Detection

Destination-side change detection

- Use ETL logic to compare source rows to destination rows
- Can have performance issues
- Fewest technical assumptions

Change Detection

Change detection in SSIS

- Use native tools in database engine
  - CDC
  - Change tracking
  - Temporal tables (2016+)

Change Detection

Change detection in SSIS

- Destination comparision via T-SQL
  - MERGE statement
  - Row hashing
  - Brute force (column-by-column)

Change Detection

Change detection in SSIS

- Data flow change detection
  - Lookup transformation
  - Merge Join transformation
  - Row hashing
  - Brute force detection

Change Detection

Whichever change detection method is used, SSIS is flexible enough to support it.

AUTOMATING DEVELOPMENT

Automating Development

- Often, SSIS data warehouse ETL development is repetitive
- Similar patterns used over and over

Automating Development

For pattern-based development, Biml can speed up the process

- Business Intelligence Markup Language
- Free to install and use
- Creates 1:n SSIS packages with minimal code

Automating Development

Benefits

- Speed up development
- Makes changes faster, more consistent
- Eliminate human error on mundane work


DEMO
LIVE! 360