

# Testing SSIS Packages

*Tim Mitchell*

1

## Agenda

- Why test?
- Design patterns for test-ready packages
- Design time vs. runtime testing
- Performance testing
- Testing error logic

## About Tim Mitchell

- Business intelligence consultant
- Principal Data Architect, Tyleris Data Solutions
- Microsoft Data Platform MVP
- TimMitchell.net
- Twitter.com/Tim\_Mitchell

## Housekeeping

- Questions
- Feedback
- Contact Info

## WHY TEST?

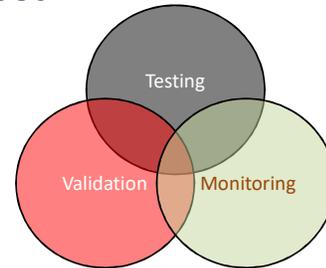
## Why Test?

- Better, more reliable processes
- Avoid the 3am wake-up call!
- Know what to expect
  - Performance
  - Error prevention and/or response

## Why Test?

- Keep the surprises to a minimum
- Avoid discovering new execution patterns in production

## Why Test?



## Why Test?

- Remember – testing never ends!

## DESIGN PATTERNS FOR TEST-READY PACKAGES

## Design Patterns for Test-Ready Packages

- Keep packages small
  - Recommended practice: One package per operation
  - Test a single package at a time
  - Also easier to share work with multiple developers
  - Reusability

## Design Patterns for Test-Ready Packages

- Externalize changeable values
  - Connection strings
  - File paths
  - Business logic
- Makes easier the task of manual and automated testing

## Design Patterns for Test-Ready Packages

- Documentation
  - Naming conventions for tasks, components
  - Application Name property on DB connections

13

## DESIGN TIME VS. RUNTIME TESTING

14

## Design Time vs. Runtime Testing

- Design time: pre-deployment
- Not just Visual Studio!

15

## Design Time vs. Runtime Testing

- Design time testing activities
  - Intentionally bad data
  - Error prevention/handling

16

## Design Time vs. Runtime Testing

- Runtime testing: Because we don't trust what we can't control

17

## Design Time vs. Runtime Testing

- Runtime test: Package metadata validation
  - Fail early rather than late
  - Eliminate the need for rollback or cleanup
  - Notify early

18

## Design Time vs. Runtime Testing

- Runtime test: Data validation
  - Handle non-failure data anomaly
  - Row count match
  - Aggregate (dollar amount total, etc.) validation
  - Other custom metrics

18

## Design Time vs. Runtime Testing

- Both testing realms are needed
- Like visiting the physician

20

## PERFORMANCE TESTING

21

## Performance Testing

- Objectives of perf testing
- Capacity planning
- Make sure you're not beating down your server
- See if you can benefit from parallelism

22

## Performance Testing

- Data flow performance testing
  - Throughput testing (end-to-end)
  - Isolation testing

23

## Performance Testing

- Control flow performance testing
  - Identify bottleneck tasks
  - Parallel process testing
  - External processes

24

## Performance Testing

- Data flow performance testing
  - More invasive than control flow testing
  - May require package modification for exhaustive testing
  - Use logging as much as possible

26

## Performance Testing

- Logging is critical!
  - Runtime for tasks
  - Component phases
  - Compare relative run times with row counts

26

## TESTING ERROR LOGIC

27

## Failure Testing

- Check error prevention and handling logic
- What happens if the error handler fails?
- What happens if the error notification fails?

28

## Failure Testing

- Testing should include error conditions
- Force errors
- Force non-failure anomalies (row count, etc.)
- End-to-end tests

29

## DEMOS

30

**Thank you!**

- [Tim@TimMitchell.net](mailto:Tim@TimMitchell.net)
- [Twitter.com/Tim\\_Mitchell](https://twitter.com/Tim_Mitchell)
- [www.TimMitchell.net](http://www.TimMitchell.net)