

Do's and don'ts of PDI

Best Practices

May 17th 2014

Matt Casters

Chief Architect Data Integration at Pentaho

Schedule

- Introduction
- Naming conventions
- Parameters
- Logging
- ETL re-use
- Creating loops
- Building queues
- Load balancing
- Documentation
- Performance best practices
- Basic lifecycle management
- Q&A



Naming conventions

- Naming conventions
 - Transformations / steps
 - big_file.txt
 - \${FILENAME}
 - CRM.cust_table
 - Jobs / job-entries
 - Database connections
 - No host/platform/Environment specific names
 - Database tables and fields
 - D_CUST, DIM_CUSTOMER, D_CUSTOMERS
 - Directories / Folders
 - Server names
 - Notes and descriptions



Parameters & Variables

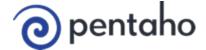
- Parameters
 - Make variable usage explicit
 - Allow others to list occurences
 - Cleaner / easier
 - Easier testing with default values
- Variables in shared objects
 - kettle.properties file
 - \${SOLUTION_HOME}
 - \${INPUT_FOLDER}
 - \${CRM_SERVER}
 - \${LARGE_LOAD_STEP_COPIES}



Logging

- Transformation
- Jobs
- Step performance
- Job entries
- Log channels
- Global setup with variables

•



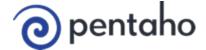
ETL re-use

- Mappings
- Simple mappings
- This is a "macro"
- Use different field names
- Avoid "Select Values" / Rename / Remove
- Parent/child lookup example



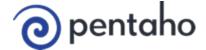
Loops

- Creating loops
- Easier in versions >=5.0
- Job Executor step
- "New" button
- Example: Process small files



Queues

- What is a queue?
- Building and using queues
- Explicit process logging
- Transparent



Load balancing

- Interrogate carte webservice
- Using carte and queues
- Documented example



Documentation

- Cookbook project
 - https://code.google.com/p/kettle-cookbook/
- Automatic Documentation step
- Impact analyses extraction
- API: specific metadata extraction:
 - Load transformations
 - Extract for all steps:
 - Type
 - type description
 - Name
 - Store in database



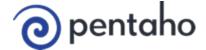
Transformation performance

- Slowest link in the chain
- Parallelisation
- Pipe-lining
- Data partitioning
- Doing work where it is fastest



Lifecycle management

- Import/export for repositories
 - Automated export
 - Export rules
- Version control for file based setups



Thank You

JOIN THE CONVERSATION. YOU CAN FIND US ON:









