

Pentaho Data Integration (PDI) Techniques - Naming Standards



Change log (if you want to use it):

Date	Version	Author	Changes
10/11/2017	1.0	Matthew Casper	
1/5/2018	1.1	Sandra Wagner	edits

Contents

Overview	1
Before You Begin.....	1
Other Prerequisites	1
Use Cases	1
PDI Organization and Naming Conventions.....	2
Job Naming.....	2
Transformation Naming.....	3
Job Entry Naming	4
Transformation Step Naming.....	5
Related Information.....	7
Finalization Checklist.....	7
Appendix 1: Job Entry Naming Conventions.....	8
Appendix 2: Step Naming Conventions.....	11

This page intentionally left blank.

Overview

This document covers some best practices on Pentaho Data Integration (PDI). In it, you will learn PDI step naming standards and tips about how to choose appropriate steps in certain situations.

Our intended audience is PDI users or anyone with a background in ETL development who is interested in learning PDI development patterns.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

Software	Version(s)
Pentaho	7.x, 8.0

The [Components Reference](#) in Pentaho Documentation has a complete list of supported software and hardware.

Before You Begin

Before beginning, use the following information to prepare for the procedures described in the main section of the document.

Other Prerequisites

This document assumes that you have knowledge about PDI and that you have already installed Pentaho software.

Use Cases

Use cases employed in this document include the following:

-
- A company new to PDI needs to set forth naming standards across the team of developers to deliver consistent lines of code which could be managed by any team member and not just the person who did the development.*
-

PDI Organization and Naming Conventions

Consistent and uniform organization, coding, and naming are important in the implementation of PDI, so that everyone using the system can find what they are looking for and understand where the information comes from and what it indicates.

You can find details on these topics in the following sections:

- [Job Naming](#)
- [Transformation Naming](#)
- [Job Entry Naming](#)
- [Transformation Step Naming](#)

Job Naming

A job serves as a wrapper for one or several transformations or jobs, and provides the ability to perform database operations such as truncate, file operations such as file transfer protocol (FTP), and notifications through email. Jobs graphically represent the order in which a table or set of tables should be loaded, and any pre-processing or post-processing that should occur.

To distinguish job names from names of other objects, and to emphasize what the job is intended to accomplish, create your own naming convention to apply to all your job names wherever possible.



For example, if a directory contains more than one type of object, you could try a prefix of `j_` on all job names, and create the rest of the name according to what the job does, like this:

`j_<Purpose/Action>_<Target Identifier>`.

Try using examples such as those in Table 1 for your job names. In Table 1, the first part of the job name, the `<Purpose/Action>`, is the verb of what is being done in the job, such as load, perform, build, move, delete, and so on. This piece of the name lets you know why the job exists.

The second part of the job name, the `<Target Identifier>`, is the name of the main subject of the job, such as `dim_tables` or `fact_tables`. It can also represent the names of other objects such as a database schema (`staging` or `bi` or any other), a file (`flat_file` or something similar), or a single table name (such as `fact_sales`).

Table 1: Example Job Names

Job Description	Job Name Example
Loads all of a database's dimension tables	load_dims
Loads all of a database's fact tables	load_facts
Handles load processes for a single database table	load_sales_fact
Main job in creating a flat file for FTP delivery	build_ftp_transaction_file
Validates files by comparing checksum values	compare_checksum
Pulls data from one area to another	pull_oltp_to_staging
Handles error notification	update_on_error send_error_notification

There are other alternatives you can consider to shorten job names, such as removing vowels and using `dim_tbls` instead of `dim_tables`.

Transformation Naming

A transformation contains the business rules that the source data must follow as it populates the target destination table(s) or file(s). Transformations, together with jobs, create the executable code that can be called using Kitchen, Pan, or an external scheduling tool. Transformations graphically represent a form of data flow diagram that connects the source table(s) with the target table(s).

One or more transformations may be included within a job. Examples include data warehouse table loads, fact table loads, dimension table loads, and aggregate table calculations.

To name transformations (or mappings, which use the `map_` prefix but otherwise follow these rules):



Use whichever naming standard (uppercase, lowercase) is right for your organization, but be consistent. In our example, we will use lowercase.

- Create transformation names in all lowercase.
- Use this convention for transformation or mapping naming:

```
<Purpose/Action>_<Subject/Target Identifier>
map_<Purpose/Action>_<Subject/Target Identifier>
```

- Fill in the <Purpose/Action> and <Subject/Target Identifier> parts of the name depending on what the transformation does. For example, if the transformation loads the `dim_equipment` table, try naming the transformation `load_dim_equipment`. If the transformation truncates all the dimension tables, it makes more sense to name the transformation based on that action and subject: `truncate_dim_tables`.

Table 2: Example Transformation Names

Transformation Description	Transformation Name Example
Loads the <code>dim_equipment</code> table	<code>load_dim_equipment</code>
Loads the <code>fact_sales</code> table	<code>load_fact_sales</code>
Truncates all dimension tables	<code>truncate_dim_tables</code>
Sets filename received from previous results as a variable	<code>set_file_name_var</code>
Handles inserts to the dimension tables for the Not Applicable or zero record only	<code>initial_insert_na_record</code>
Sets variables for a specific database area	<code>set_staging_vars</code> <code>set_dwh_vars</code>
Mapping that obtains different metadata properties from a text file	<code>map_file_properties</code>

As with the job naming, one way to make transformation names shorter is to remove vowels.

Job Entry Naming

A job entry name should reflect the function of the object based on the following criteria:

- An entry based on a target object (like a transformation) should inherit the target's name (like `dim_date_hour`).
- Name an entry based on an action to reflect the purpose of the action (like `delete_closure_records`).
- Always name entries in lowercase letters (or according to the consistent upper/lowercase usage of your organization).

Use the standard prefix for entries, and name them with descriptions of the operations or intent of the entry. The full list of job entry naming conventions is available in [Appendix 1](#).

Table 3: Example Job Entry Names

Job Entry	Naming Convention	Description/Example
Transformation	Transformation name	This entry should reflect the transformation name, <code>load_dim_equipment</code> .
Simple evaluation	<code>se-+ purpose/attribute</code>	If the simple evaluation is being performed on a variable determining whether the run is the initial load, the job entry name should be <code>se-initial_load</code> .
Structured Query Language (SQL)	<code>sql-+ action/purpose/filename</code>	If the SQL script contains code to drop indexes on a table, then the name can simply be <code>sql-drop_indexes</code> . You can include the table name as well, but consider the overall job entry name length.
Write to log	<code>wtl-+ purpose/job name</code>	If the <code>load_dims</code> job is making note in the log of what parameter/variable values are being used at execution time, the name should be <code>wtl-load_dim_vars</code> .
Job	Job name	This should be the same name of the job being called by the job entry step, <code>load_dims</code> .
Set variables	<code>svar-+ purpose</code>	If you are setting variables coming from a properties file named <code>staging.properties</code> , then the job entry name should be <code>svar-staging_properties</code> .

Transformation Step Naming

There are more than 200 different transformation steps available in developing a transformation, and the basic naming convention is a two- to seven-letter abbreviation of the step name followed by a colon and a short description identifying the step's purpose.

A step name should reflect the function of the object based on the following criteria:

- A step based on a source object (like a table or file) should inherit the source's name.
- A step based on a target object should inherit the target's name.
- Always name steps in lowercase letters (or according to the consistent upper/lowercase usage of your organization).

Use the standard prefix for steps, and name them with descriptions of the operations or intent of the entry. The full list of step naming conventions is available in [Appendix 2](#).

Table 4: Example Step Names

Step Type	Naming Convention	Description/Example
Add constants	ac-+ attribute/subject name	Name this step as ac-+ the name of the attribute being generated, or, if there are many attributes, a subject area name would be sufficient.
Add sequence	aseq-+ attribute name for which the sequence is being generated	If the sequence is being generated for <code>column_product_key</code> , then the step should be named <code>aseq-product_key</code> .
Database lookup	dblkp-+ attribute name for which lookup is being performed	If the lookup is being performed on <code>product_key</code> using the table <code>DIM_PRODUCT</code> , then name the step <code>dblkp-product_key</code> .
Filter rows	fr-+ attribute name for which the filter is being applied	If the filter step is based on a filter condition for the state attribute, name the step <code>fr-state</code> .
Get system info	gsi-+ attribute name for which information is being selected	Name this step as <code>gsi-+</code> the name of the attribute being generated. If the <code>system_date</code> is being pulled, name the step <code>gsi-system_date</code> .
Insert/Update	iu-+ target/entity name for which the update strategy is being defined	If the insert/update strategy is being defined for the table <code>customer</code> , name the step <code>iu-customer</code> .
Select values	sv-+ purpose + attribute	Use this step to clean up a data stream so that only the needed columns are included or so the columns have the correct name or data type. If you are renaming attribute <code>reason_1</code> to <code>reason</code> , name the step <code>sv-rename_reason</code> .
Table input	ti-+ source/subject name	If the source data is defined on table <code>dim_product</code> , then name the step <code>ti-dim_product</code> . If many tables are used, provide a subject name that represents the data being pulled, like <code>ti-transaction_data</code> .
Table output	to-+ the name of the target	These step names should represent the target table. If you are inserting records into the <code>customer</code> table, name the step <code>to-customer</code> .

Related Information

Here are some links to information that you may find helpful while using this best practices document:

- [Components Reference](#)

Finalization Checklist

This checklist is designed to be added to any implemented project that uses this collection of best practices, to verify that all items have been considered and reviews have been performed. (Compose specific questions about the topics in the document and put them in the table.)

Name of the Project: _____

Date of the Review: _____

Name of the Reviewer: _____

Item	Response	Comments
Did you set up a clearly named folder structure?	YES_____ NO_____	
Did you systematically name folders, jobs, transformations, and steps?	YES_____ NO_____	

Appendix 1: Job Entry Naming Conventions

Here is the complete list of job entry naming conventions:

Table 5: Job Entry Naming Conventions

Job Entry Name	Naming Convention
Abort job	'abrt-' + purpose/reason
Add filenames to result	'afntr-' + file name/purpose
Amazon EMR Job Executor	'aemrje-' + job name
Amazon Hive Job Executor	'ahje-' + job name
BulkLoad from Mysql into file	'blmysqlf-' + file name/purpose
BulkLoad into MSSQL	'blmssql-' + target
BulkLoad into Mysql	'blmysql-' + target
Check Db connections	'chkdbc-' + connection name/purpose
Check files locked	'chkfl-' + file/subject area/purpose'
Check if a folder is empty	'chkfldr-' + folder name/purpose
Check if connected to repository	'chkrepo-' + repo name/purpose
Check if XML file is well formed	'chkxml-' + xml name/purpose
Check webservice availability	'chkws-' + webservice name/purpose
Checks if files exist	'chkfe-' + file name/subject area
Columns exist in a table	'colex-' + column name/table name/subject area
Compare folders	'cfldrs-' + purpose/subject area
Convert file between Windows and Unix	'cnvrtf-' + file name/purpose
Copy Files	'cf-' + file name/subject area/purpose
Copy or Move result filenames	'cmf-' + file name/subject area/purpose
Create a folder	'cflldr-' + folder name
Create file	'cf-' + file name
Decrypt files with PGP	'dpgp-' + file name/subject area/purpose
Delete file	'df-' + file name
Delete filenames from result	'dffr-' + file name/subject area/purpose
Delete files	'dfls-' + file name/subject area/purpose
Delete folders	'dfldrs-' folder name/subject area/purpose
Display MsgBox Info	'msgbox-' + purpose
DTD Validator	'dtdval-' + dtd file name/xml file name
Dummy	'dmmy-' + purpose

Job Entry Name	Naming Convention
Encrypt files with PGP	'epgp-' + file name/subject area/purpose
Evaluate files metrics	'evalfm-' + file name/subject area/purpose
Evaluate rows number in a table	'evalrnt-' + table name/purpose
Export repository to XML file	'erepo-' + xml file name
File Compare	'fc-' + files/subject area/purpose
File Exists	'fe-' + file name/subject area/purpose
FTP Delete	'ftpd-' + ftp server name/file name/subject area/purpose
Get a file with FTP	'gftp-' + file name
Get a file with FTPS	'gftps-' + file name
Get a file with SFTP	'gsftp-' + file name
Get mails (POP3/IMAP)	'gmail-' + purpose/subject area
Hadoop Copy Files	'hadcp-' + file name/purpose/subject area
Hadoop job executor	'hadje-' + job name
HL7 MLLP Acknowledge	'hl7ack-' + server name/message/purpose
HL7 MLLP Input	'hl7i-' + source
HTTP	'http-' + purpose/webpage name
JavaScript	'js-' + purpose
Job	Job Name
Mail	'mail-' + purpose
Mail validator	'mailv-' + purpose
Move Files	'mf-' + file name/purpose/subject area
MS Access Bulk Load	'msabl-' + target
Oozie Job Executor	'oozje-' + job name
Palo Cube Create	'pcc-' + cube name
Palo Cube Delete	'pcd-' + cube name
Pentaho MapReduce	'pmr-' + purpose
Pig Script Executor	'pse-' + script name/purpose
Ping a host	'ping-' + host name/purpose
Put a file with FTP	'pftp-' + file name
Process result filenames	'prfn-' + file name/purpose/subject area
Put a file with SFTP	'psftp-' + file name
Send information using Syslog	'syslog-' + server name/message purpose
Send Nagios passive check	'nagios-' + host name/message purpose

Job Entry Name	Naming Convention
Send SNMP trap	'snmp-' + server name/message purpose
Set variables	'svar-' + purpose
Shell	'shll-' + script name/purpose
Simple evaluation	'se-' + purpose/attribute
SQL	'sql-' + action/purpose/file name
Sqoop Export	'sqpex-' + target
Sqoop Import	'sqpim-' + source
SSH2 Get	'ssh2g-' + source
SSH2 Put	'ssh2p-' + target
Start	Start
Start a PDI Cluster on YARN	'strtyarn-' + cluster schema/purpose
Stop a PDI Cluster on YARN	'stpyarn-' + cluster schema/purpose
Success	Success
Table exists	'tble-' + table name
Talend Job Execution	'talje-' + job name
Transformation	Transformation name
Truncate tables	'trunc-' + purpose/schema name
Unzip file	'unzip-' + file name/purpose
Upload files to FTPS	'uftp-' + file name/purpose/subject area
Verify file signature with PGP	'vpgp-' + file name/purpose
Wait for	'wait-' + purpose
Wait for file	'waitf-' + file name
Wait for SQL	'waits-' + purpose
Write to file	'wtf-' + file name
Write To Log	'wtl-' + purpose/job name
XSD Validator	'xsdval-' + xml file name
XSL Transformation	'xlst-' + xml file name/output file name'
Zip file	'zip-' + file name

Appendix 2: Step Naming Conventions

Here is the complete list of step naming conventions:

Table 6: Step Naming Conventions

Step Name	Naming Convention
Abort	'abrt-' + purpose/reason
Add a checksum	'chksum-' + subject name/purpose
Add Constants	'ac-' + attribute/subject name
Add Sequence	'aseq-' + attribute name for which sequence is being generated
Add value fields changing sequence	'chgseq-' + attribute/purpose
Add XML	'axml-' + subject name/reason
Analytic Query	'aq-' + subject name/purpose
Append Streams	'as-' + subject name/purpose
ARFF Output	'arffo-' + target
Automatic Documentation Output	'ado-' + target
Avro input	'avroi-' + source/subject name
Block this step until steps finish	'bsusf-' + purpose/subject name
Blocking Step	'bs-' + purpose/subject name
Calculator	'calc-' + purpose
Call DB Procedure	'dbproc-' + procedure name
Call Endpoint	'endpt-' + purpose/subject name
Cassandra input	'cassi-' + source
Cassandra output	'casso-' + target
Change file encoding	'chgfe-' + purpose/subject name
Check if a column exists	'chkce-' + column name
Check if file is locked	'chkfl-' + file name
Check if webservice is available	'chkwsa-' + webservice name
Clone Row	'cr-' + purpose
Closure Generator	'cg-' + Hierarchy
Combination lookup/update	'clkpu-' + table name
Concatenate Fields	'cf-' + name of target/purpose
Copy Rows to Result	'crtr-' + purpose/subject name
CouchDB Input	'cdbi-' + source name

Step Name	Naming Convention
Credit card validator	'ccval-' + field name/card type
CSV file input	'csvi-' + source name
Data Grid	'dg-' + source/subject name
Data Validator	'dval-' + purpose/subject name
Database Join	'dbjn-' + purpose
Database Lookup	'dblkp-' + attribute name for which lookup is being performed
Delay row	'delr-' + purpose/subject name
Delete	'del-' + target name
De-serialize From File	'dsff-' + name of object/purpose
Detect Empty Stream	'des-' + subject name/purpose
Dimension Lookup/Update	'dimlu-' + dimension to be updated
Dummy (do nothing)	'dmmy-' + purpose/subject name
Dynamic SQL row	'dsqrlr-' + purpose/table name
Edi to XML	'edixml-' + purpose/target
ElasticSearch Bulk Insert	'elsbi-' + target name
Email messages input	'emsgi-' + source/subject name
ESRI Shapefile Reader	'esri-' + source/subject name
ETL Metadata Injection	'emi-' + purpose
Execute a process	'execp-' + process name
Execute row SQL script	'exrsql-' + purpose/subject name
Execute SQL script	'exsqls-' + script name
File Exists	'fe-' + filename
Filter Rows	'fr-' + attribute name for which filter is being applied
Fixed file input	'fxdfi-' + file name
Formula	'frmla-' + attribute/subject name
Fuzzy match	'fzzym-' + purpose/subject name
Generate random credit card numbers	'grcc-' + purpose/subject name
Generate random value	'grval-' + attribute/subject name
Generate Rows	'gr-' + attribute/subject name
Get data from XML	'gdxml-' + source
Get File Names	'gfn-' + file/purpose
Get files from result	'gffr-' + files/purpose

Step Name	Naming Convention
Get Files Rows Count	'gfr-' + file/purpose
Get ID from slave server	'gidfss-' + purpose/slave server name
Get repository names	'grnms-' + repo name/purpose
Get rows from result	'grfr-' + purpose/subject area
Get Session Variables	'gsvars-' + purpose/variable name
Get SubFolder names	'gsfnms-' + purpose/directory name
Get System Info	'gsi-' + attribute name for which information is being selected
Get table names	'gtbls-' + schema name/purpose
Get Variables	'gv-' + purpose/target table name
Google Analytics	'gglea-' + purpose
Google Docs Input	'ggledi-' + source/purpose
Greenplum Bulk Loader	'gpbl-' + target
Greenplum Load	'gpl-' + target
Group By	'grp-' + attribute/subject name
GZIP CSV Input	'gzcsvi-' + source
Hadoop File Input	'hdpfi-' + source
Hadoop File Output	'hdpfo-' + target
HBase input	'hbsi-' + source
HBase output	'hbso-' + target
HBase Row Decoder	'hbsrd-' + purpose/subject area/table name
HL7 Input	'hl7i-' + source
HTTP client	'httpc-' + purpose
HTTP Post	'http-' + purpose/web page
IBM Websphere MQ Consumer	'ibmc-' + source
IBM Websphere MQ Producer	'ibmp-' + target
Identify last row in a stream	'idlr-' + purpose/field name
If field value is null/Replace Null Value	'rnv-' + subject name
Infobright Loader	'ibldr-' + target
Ingres VectorWise Bulk Loader	'vwblldr-' + target
Injector/Row Injector	'rinj-' + purpose
Insert/Update	'iu-' + target/entity name for which update strategy is being defined
Java Filter	'javaf-' + purpose/field name

Step Name	Naming Convention
JMS Consumer	'jmsc-' + source
JMS Producer	'jmsp-' + target
Job Executor	'je-' + target job name
Join Rows (cartesian product)	'jr-' + purpose
JSON Input	'jsoni-' + source
JSON output	'jsono-' + target
Knowledge Flow	'kfl-' + source/purpose
LDAP Input	'ldapi-' + source
LDAP Output	'ldapo-' + target
LDIF Input	'ldifi-' + source
Load file content in memory	'lfcml-' + source
LucidDB Bulk Loader	'ldbbl-' + target
LucidDB Streaming Loader	'ldbsl-' + target
Mail	'mail-' + purpose/subject name
Mail Validator	'mailv-' + purpose
Mapping (sub-transformation)	'm-' + purpose
Mapping input specification	'mis-' + purpose/source
Mapping output specification	'mos-' + purpose/target
MapReduce Input	'mapri-' + source
MapReduce Output	'mapro-' + target
MaxMind GeoIP Lookup	'mmndlkp-' + source
Memory Group by	'mgb-' + purpose/field name
Merge Join	'mj-' + purpose
Merge Rows (diff)	'mr-' + purpose
Metadata structure of stream	'mdss-' + purpose
Microsoft Access Input	'msai-' + source
Microsoft Access Output	'msao-' + target
Microsoft Excel Input	'msxlsi-' + source
Microsoft Excel Output	'msxlso-' + target
Microsoft Excel Writer	'msxlsw-' + target
Modified Java Script Value	'mjstv-' + attribute name being generated
Mondrian Input	'mndrni-' + source
MonetDB Agile Mart	'mdbam-' + target
MonetDB Bulk Loader	'mdbbl-' + target

Step Name	Naming Convention
MongoDB Input	'mdbi-' + source
MongoDB Output	'mdbo-' + target
Multiway Merge Join	'mwmj-' + purpose
MySQL Bulk Loader	'mysqlbl-' + target
Null if...	'null-' + field name/purpose
Number range	'nbrr-' + field name/purpose
OLAP Input	'olapi-' + source
OpenERP Object Delete	'erpdel-' + target
OpenERP Object Input	'erpinp-' + source
OpenERP Object Output	'erpout-' + target
Oracle Bulk Loader	'orabl-' + target
Palo Cell Input	'paloi-' + source
Palo Cell Output	'paloo-' + target
Palo Dimension Input	'palodi-' + source
Palo Dimension Output	'palodo-' + target
Pentaho Reporting Output	'pro-' + target
PostgreSQL Bulk Loader	'psqlbl-' + target
Prioritize Streams	'ps-' + name of key field/purpose
Process files	'pfiles-' + source/directory/subject name
Property Input	'pi-' + source
Properties Output	'po-' + name of target file/purpose
R script executor	'rsrpt-' + script name
Regex Evaluation	're-' + purpose
Replace in String	'ris-' + file/purpose
Reservoir Sampling	'rs-' + purpose
REST Client	'rest-' + source/purpose
Output steps metrics	'osm-' + subject name
Row De-Normalizer	'rdnml-' + source/subject name
Row flattener	'rf-' + purpose/source field
Row Normalizer	'rnml-' + source/subject name
RSS Input	'rssi-' + source
RSS Output	'rsso-' + target
Rules Accumulator	'racc-' + source/purpose
Rule Executor	'rexe-' + source/purpose

Step Name	Naming Convention
Run SSH commands	'rssh-' + source/purpose
S3 CSV Input	's3csvi-' + source
S3 File Output	's3fo-' + target
Salesforce Delete	'slsfd-' + target
Salesforce Input	'slsfi-' + source
Salesforce Insert	'slsfinsrt-' + target
Salesforce Update	'slsfu-' + target
Salesforce Upsert	'slsfupsrt-' + target
Sample rows	'smplr-' + purpose
SAP HANA Bulk Loader	'sapbl-' + target
SAP Input	'sapi-' + source
SAS Input	'sasi-' + source
Script	'scrpt-' + purpose
Secret key generator	'skgen-' + purpose/key field
Select Values	'sv-' + purpose
Send message to Syslog	'smsgsl-' + target/purpose
Serialize to File	'stf-' + name of target/purpose
Set field value	'sfv-' + field name/purpose
Set field value to a constant	'sfvc-' + field name/purpose
Set files in result	'sfnr-' + filename field/purpose
Set Session Variables	'ssvar-' + purpose/variable name
Set Variables	'svar-' + purpose
SFTP Put	'sftpp-' + target/purpose
Simple Evaluation	'se-' + subject name
Simple Mapping	'sm-' + target/purpose
Single Threader	'sthr-' + target/purpose
Socket reader	'scktr-' + source
Socket writer	'scktw-' + target'
Sort Rows	'sr-' + purpose
Sorted Merge	'sm-' + purpose
Split Fields	'sf-' + purpose
Split Fields to rows	'sftr-' + purpose
Splunk Input	'splnki-' + source
Splunk Output	'splnko-' + target

Step Name	Naming Convention
SQL File Output	'sqlfo-' + target
SSTable Output	'sstblo-' + target
Stream Lookup	'slkp-' + attribute name/purpose
Streaming XML Input	'sxmli-' + source
String Cut	'sc-' + purpose
String Operation	'so-' + name of key field/purpose
Switch/Case	'swcs-' + process being evaluated
Symmetric Cryptography	'symc-' + result field name/purpose
Synchronize after merge	'syncham-' + target
Table Agile Mart	'tblam-' + purpose
Table Compare	'tblc-' + target/purpose
Table exists	'tble-' + target
Table Input	'ti-' + source/subject name
Table Output	'to-' + name of the target
Teradata Fastload Bulk Loader	'tdbl-' + target
Teradata TPT Insert Upsert Bulk Loader	'tdiubl-' + target
Text File Input	'tfi-' + file name/purpose
Text File Output	'tfo-' + file name
Transformation Executor	'te-' + transformation name/purpose
Unique Rows	'ur-' + field name/purpose
Unique rows (HashSet)	'urh-' + field name/purpose
Univariate Statistics	'unvstats-' + field name/purpose
Update	'updt-' + name of target
User Defined Java Class	'udjc-' + purpose/field name
User Defined Java Expression	'udje-' + purpose
Value Mapper	'vm-' + subject name/purpose
Vertica Bulk Loader	'vbl-' + target
Web services lookup	'wslkp-' + target/purpose
Write to log	'wlog-' + purpose
XBase input	'xbsi-' + source
XML Input	'xmli-' + source
XML Input Stream (StAX)	'xmlis-' + source
XML Join	'xmlj-' + target/purpose

Step Name	Naming Convention
XML Output	'xmlo-' + name of the target
XSD Validator	'xsdv-' + purpose/source file
XSL Transformation	'xslt-' + source/purpose
Yaml Input	'yamli-' + source