Treasure Workflow Terms and Concepts

Treasure Workflow offers best practices for workflow processes and uses standard industry terms. Treasure Workflow extends and enhances the capabilities of the open-source workflow program, Digdag, and therefore the workflow concepts are the same. You can use the following list of terms and concepts as a reference.

Term	Description
Task	An action to be taken, as specified in a workflow definition file. Task syntax consists of Task Name and Operator, and can also contain Variables and Parameters. A task can group tasks, in which case it is a grouping-only task. A task can generate additional tasks, in which case it is a task generating task. When a task has finished generating tasks, it turns into a grouping task because it now has children. Tasks can have dependencies on other tasks. If a task is independent, it is a task.
Task Name	A descriptive name specified by the user that labels each task in a workflow definition file. In Treasure Data workflow syntax, each task name is preceded by the + (plus) symbol.
Operator	An instruction that is part of a workflow task. In Treasure data workflow syntax, each operator is followed the > (greater than) symbol. Types of pre-defined operators include workflow control operators from Digdag (such as call>, loop>, echo>), Treasure Data operators (such as td>, td_run>, td_load>, td_table_export), and operators for databases, and networks. Operators can contain parameters and scripts (such as queries or other calls) and are designed to act like plugins that can be reused in multiple workflows.
Parameter	A constant or a variable that further specifies an operator. In Treasure data workflow syntax, there are three kinds of parameters: local, export, or store. • Local parameters are set directly to the task. • Export parameters are used for a parent task to pass values to children. • Store parameters are used for a task to pass values to all subsequent tasks, including children. Parameters are merged into one object when a task runs. Local parameters have the highest priority. Export and store parameters override each other and thus parameters set at later tasks have higher priority. Store parameters are not global variables. When two tasks run in parallel, they will use different store parameters. This makes the workflow behavior consistent regardless of actual execution timing.
Variable	A keyword for a class or set of objects. In Treasure data workflow syntax, each variable is preceded by the $ (dollar) symbol with values contained in {} brackets. Some variables already built-in Treasure workflow include: timezone, session_id, task_name. You can also define your own variables. You can define variables in 3 ways: using the _export parameter in YAML, programmably using API, or as you start a session using -p KEY=VALUE. You can use basic JavaScript scripts in ${...} syntax to calculate variables.
Project	A container for workflows and files used by a set of workflows. The files can be almost any script. For example, SQL, Python, Ruby, Shell scripts and configuration files. A project is used to group related workflows, for example, workflows that complete a specific action or that have dependencies on each other. All workflows in a project are updated together when you upload a new revision.
Revision	A version of a project. When you edit a workflow in a project or files that are part of a project a new revision is created. Project revision history is found in Workflows area in the TD Console. Earlier revisions may be selected and restored.
Session	A plan to run a workflow In the Workflow UI. A session specifies the date of the data, identifying the data set that the workflow acts upon. A session can be unscheduled or scheduled but must be unique. In a workflow, you cannot have two sessions specifying the same date for a data set. In Treasure Data, the default session value is the current timestamp; unscheduled workflows use the default session value, which is the current timestamp.
Session Time	A timestamp called session_time, for which a session is to run. The session_time is unique in history of a workflow. If you submit two sessions with the same session_time, the later request will be rejected. This prevents accidental submission of a session that ran before for the same time. If you need to run a workflow for the same time, you should retry the past session instead of submitting a new session.
Execution Time	A timestamp that captures when a workflow ran, whether successfully completed or not. Example: You might have a workflow that is scheduled every day, the session_time is 00:00:00 of a day such as 2017-01-01 00:00:00. Actual execution time may not be the same time. You might want to delay execution for 2 hours because some data need 1 hour to be prepared.
Attempt	An actual execution of a session. A session has multiple attempts if you retry a failed workflow. If a session fails, you can check attempts of it, and debugs the problem from the logs. You may upload a new revision to fix the issue, then start a new attempt.