Talend and HP Vertica Tips and Techniques

HP Vertica Analytic Database

Document Release Date: 12/15/14

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 2006- 2014 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation. UNIX® is a registered trademark of The Open Group.

Talend and HP Vertica Tips and Techniques

12/15/2014

Contents

About HP Vertica Tips and Techniques

HP Vertica develops Tips and Techniques documents to provide you with the information you need to use HP Vertica with third-party products. This document provides guidance using one specific version of HP Vertica and one version of the vendor's software. While other combinations are likely to work, Hewlett-Packard may not have tested these.

Overview

This document provides guidance for configuring Talend Open Studio to connect to HP Vertica. This document covers Talend Data Integration; it does not specifically cover any other products in the Talend Unified Platform. However, connectivity options for other Talend products should be similar to the options covered here.

Compatibility Matrix

Examples and recommendations in this guide were tested with Talend DI 5.5.1 and HP Vertica 7.0.x . The guidance provided in this document should work with earlier versions of both products.

The following table lists and describes connection testing success across various Talend and HP Vertica versions, using the native HP Vertica connector provided by Talend. Recommendations in this document have been tested using the versions in this table.

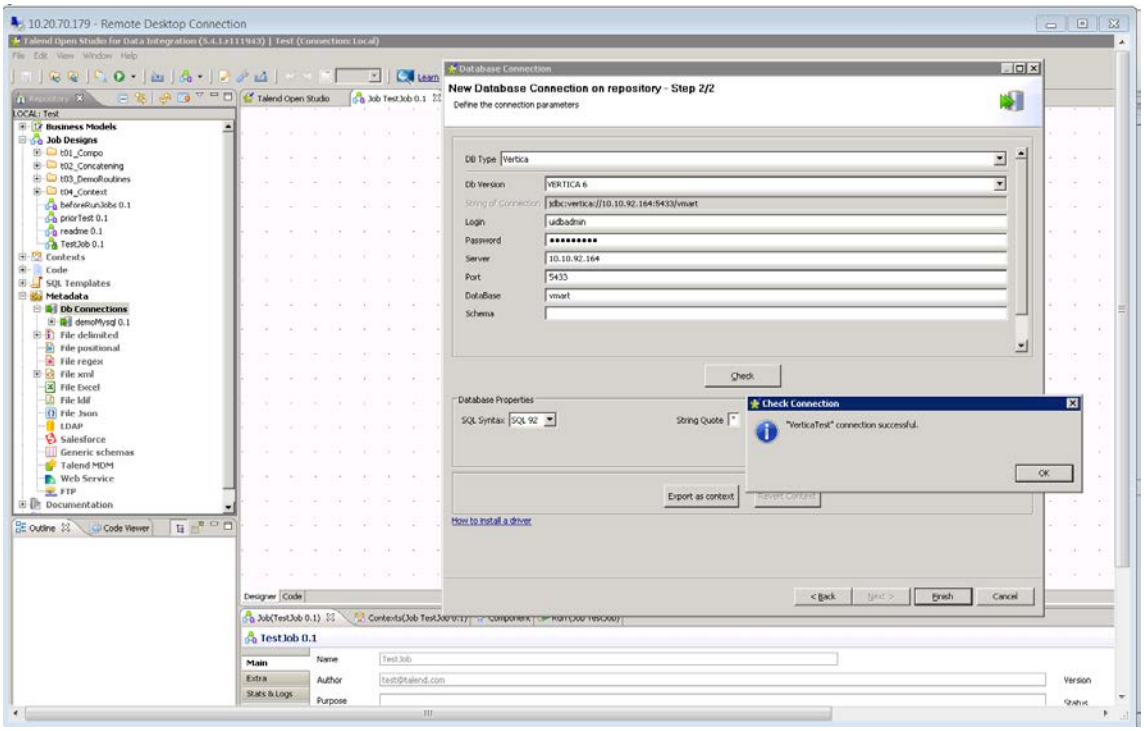
Talend Data Integration	HP Vertica JDBC Driver	HP Vertica Server
5.2 MR4	5.1.1	5.1.1
5.2 MR4	6.0.1	6.0.1
5.2 MR4	6.1.1	6.1.1
5.4.1	7.0.2	7.0.2
5.4.1	7.1.1	7.1.1
5.5.1	7.1.1	7.1.1
5.6 MR2	7.1.1	7.1.1

Connecting Talend and HP Vertica

While you can use a generic JDBC driver for a Talend/HP Vertica connection, the best way to connect is using the Vertica JDBC driver. When you create a connection, Talend automatically downloads the version of the HP Vertica JDBC driver appropriate for your version of Vertica. After the download, you can find the driver here: `<Talend_Install_Dir>/lib/java`

Setting Up a Connection

The following image shows how to create a connection using the Vertica 6.x JDBC driver.



Upgrading Your JDBC Driver

On occasion, you may need to upgrade the JDBC driver version you're using to take advantage of fixes in a later version. In these instances, you can download a more current driver from my.vertica.com and replace your driver. The example below assumes you need to replace the default HP Vertica driver provided by Talend with the version 6.1.3 driver, downloaded from my.vertica.com.

1. Exit from Talend.
2. Locate the.jar file containing your JDBC driver by searching for `vertica*jar` in the directory where Talend is installed.
3. Move the jar file (vertica-jdk5-6.0.0-0.jar) to another location outside of your Talend directory. Be sure to move all instances of the .jar file.
4. Copy the new jar file (vertica-jdk5-6.1.3-0.jar) to all of the locations where you found the original vertica-jdk5-6.0.0-0.jar .
5. Rename the new jar file (vertica-jdk5-6.1.3-0.jar) you just downloaded from the Vertica web site to vertica-jdk5-6.0.0-0.jar.

When you start Talend, you'll be using the new version of the driver.

HP Vertica Client Driver/Server Compatibility

Usually, each version of the HP Vertica server is compatible with the previous version of the client drivers. This compatibility lets you upgrade your HP Vertica server without having to immediately upgrade your client software. However, some new features of the new server version may not be available through the old drivers.

The following table summarizes the compatibility of each recent version of the client drivers with the HP Vertica server versions.

HP Vertica Client Driver Version	Compatible Vertica Server Versions
6.1.x	6.1.x, 7.0.x, 7.1.x
7.0.x	7.0.x, 7.1.x
7.1.x	7.1.x

Only the first 2 digits in the version string matter for client/server compatibility. For example, and 7.0.x client driver can talk to any 7.0.x server. For more information about client driver/server compatibility, see the Connecting to Vertica guide on vertica.com.

Installing Service Packs

Always install the latest service pack available when upgrading your driver. Service packs contain important fixes and updates. (The third digit in the release number represents the service pack.) Check the my.vertica.com downloads page to see release notes for the latest service pack.

Talend Tips and Techniques

This section provides guidance to consider when setting up Talend to work with HP Vertica. It contains these sections:

- Using Talend Components for HP Vertica
- Using the Talend SQL Builder
- Enabling Parallelization in Talend

Using Talend Components for HP Vertica

Talend provides a number of HP Vertica-specific components you can use to manage your ETL jobs. This section describes ETL flow and loading guidelines that will help you determine which components to use. It then describes some of the most commonly used components and provides recommendations for using them. For complete information on all of the Vertica components, see these Talend resources:

- Talend Open Studio 5.5.1 Components Guide
- Integrating Talend with HP Vertica (Video)

Understanding ETL Flow and Loading Guidelines

Understanding these three basic ETL flows will help you decide which Vertica component and options to use. This section addresses the following load types:

Type of Load	Use this COPY option	Results
Small Bulk Load COPY (<100MB)	AUTO	• Writes to WOS • Spills to ROS when WOS overflows
Large Bulk Load COPY	DIRECT	• Writes to ROS • Each commit becomes a new ROS container
Incremental Load COPY	TRICKLE	• Writes to WOS • Errors when WOS overflows

Follow these guidelines when loading data into HP Vertica:

- You should use COPY to load large amounts of data. Using COPY helps to avoid fragmenting the WOS, as well as the overhead of INSERT.
- If your row count is small (fewer than 1000 rows), you can use INSERT.
- You should load multiple streams on different nodes.

If you are using `INSERT into SELECT from` syntax in ETL jobs with large volumes, be sure to use the following syntax.

```
INSERT /+*direct*/ INTO table SELECT…
```

Commonly-Used HP Vertica-Specific Talend Components

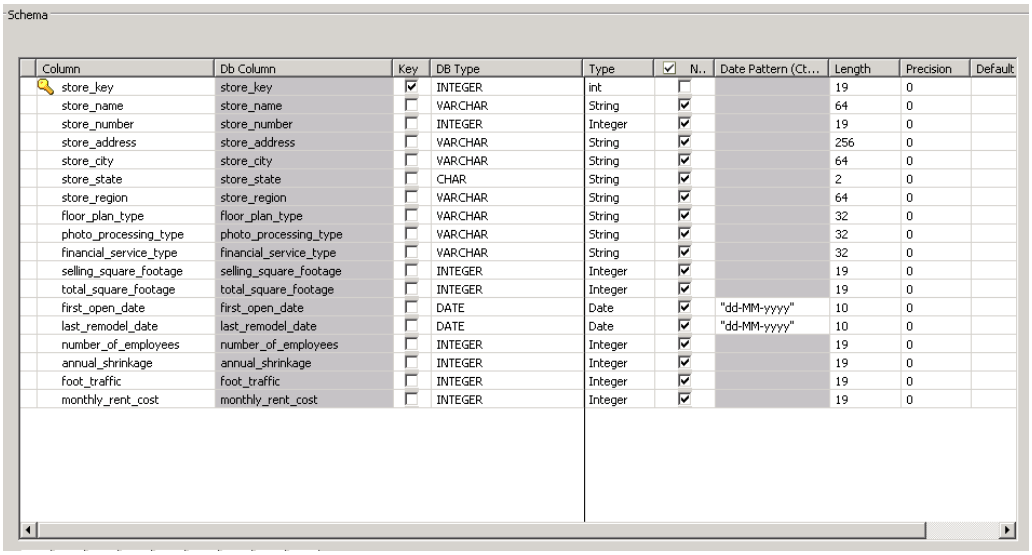
This document discusses the Talend 5.5.1 HP Vertica components listed in the following table:

Component	Description
tVerticaInput	Extracts data from HP Vertica .
tVerticaBulkExec	Loads from a file.
tVerticaOutputBulk	Writes to a file.
tVerticaOutputBulkExec	Writes to a file and loads the data.
tVerticaOutput	Inserts or updates rows into a Vertica table.
tVerticaRow	Executes the SQL query stated against the Vertica database.

tVerticaInput

The tVerticaInput component allows you extract data from HP Vertica. Give special attention to DATE and VARCHAR fields, as follows:

- Date fields-- Note that all DATE fields must be in MM-dd-yyyy format. Be sure to check your DATE fields and correct the format wherever necessary. (See the image below.)



- VARCHAR fields--Talend examines a sampling of the data when determining size of the VARCHAR. If the table contains large VARCHAR values, you may need to increase the default column.

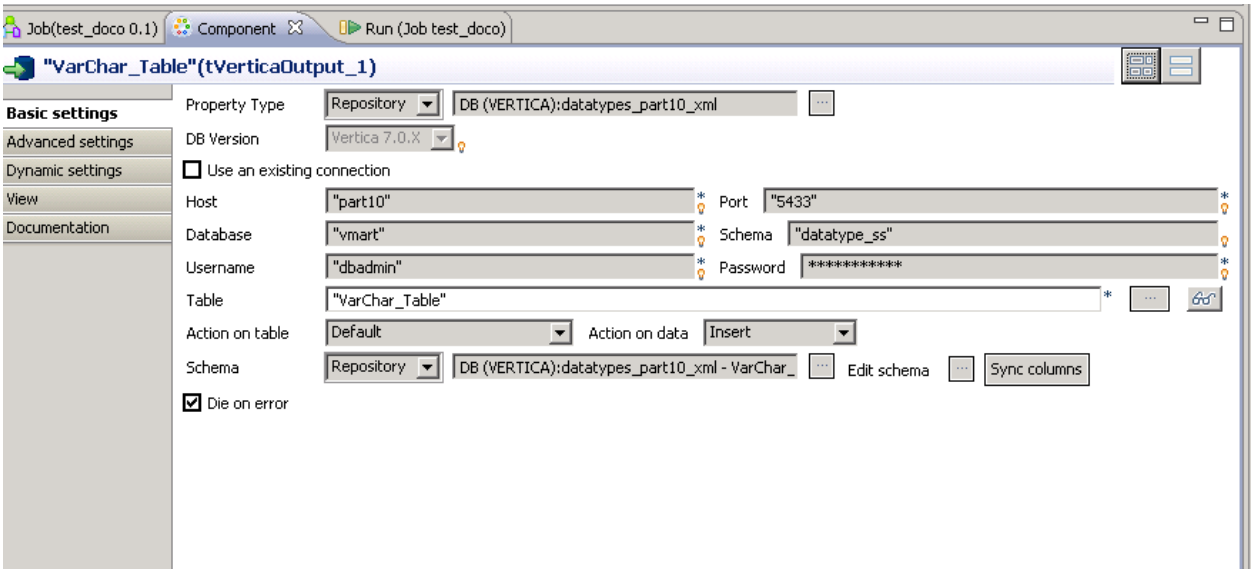
Note: You can also use the tMap component to transform or remap the data type.

tVerticaOutput

tVerticaOutput provides options you can and should change depending on the size and type of load. Specifically, the Action on Data and Action on Table options need to be changed as follows:

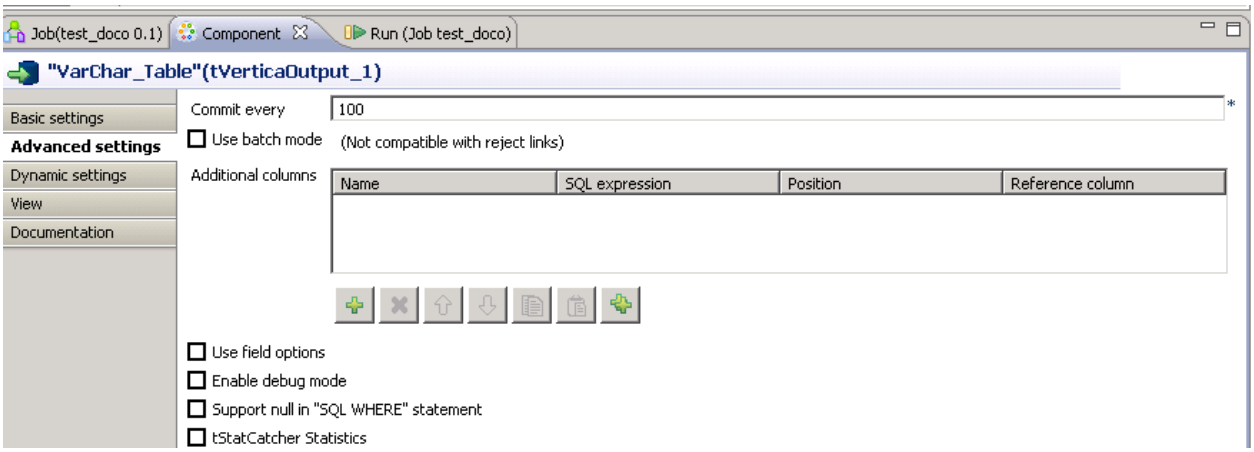
Basic Settings for Trickle or Incremental Loads (Writing to WOS)

Set the **Action on Table** and **Action on Data** options according to your use case. For Trickle or Incremental Loads, **Action on Data** should be set to INSERT.



Advanced Settings for Trickle or Incremental Loads (Writing to WOS)

When writing to WOS, use the defaults for Advanced Settings as follows:



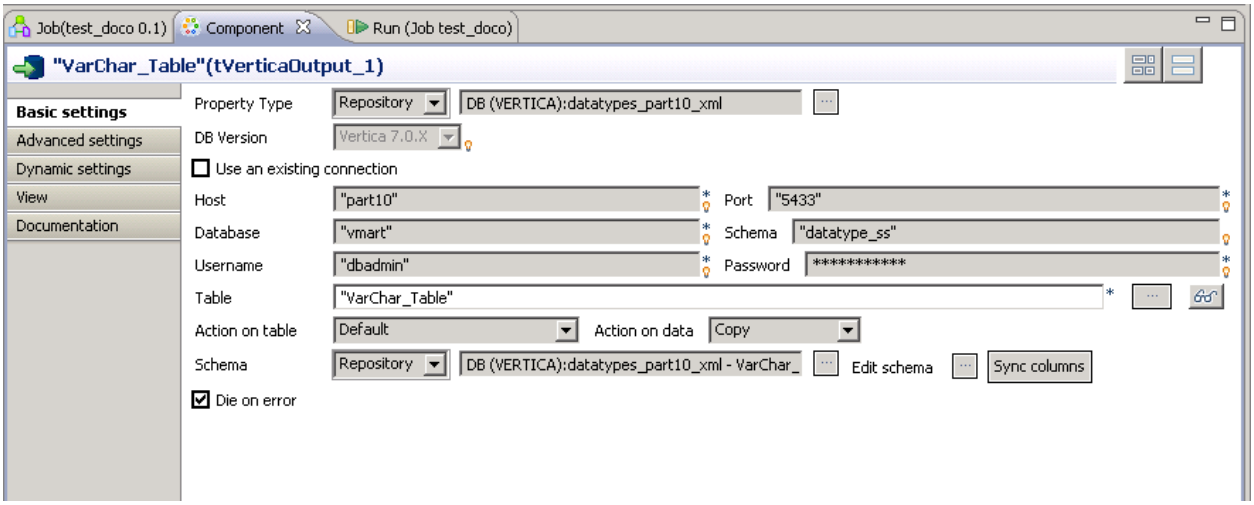
You can select the **Use field options** checkbox to enable fields for INSERT, UPDATE, and DELETE. The default settings in the example above generate the following SQL commands:

```
2014-09-16 17:10:59.679 Init Session:0x7f0a4800fe30-13000000009c0a8 [Txn]
<INFO> Starting Commit: Txn: 13000000009c0a8 'INSERT INTO
datatype_ss.VarChar_Table (KeyColumn,Varchar_Column,Varchar_Max_Column)
VALUES (?,?,?)'
```

Note that these default settings do not generate a COPY statement.

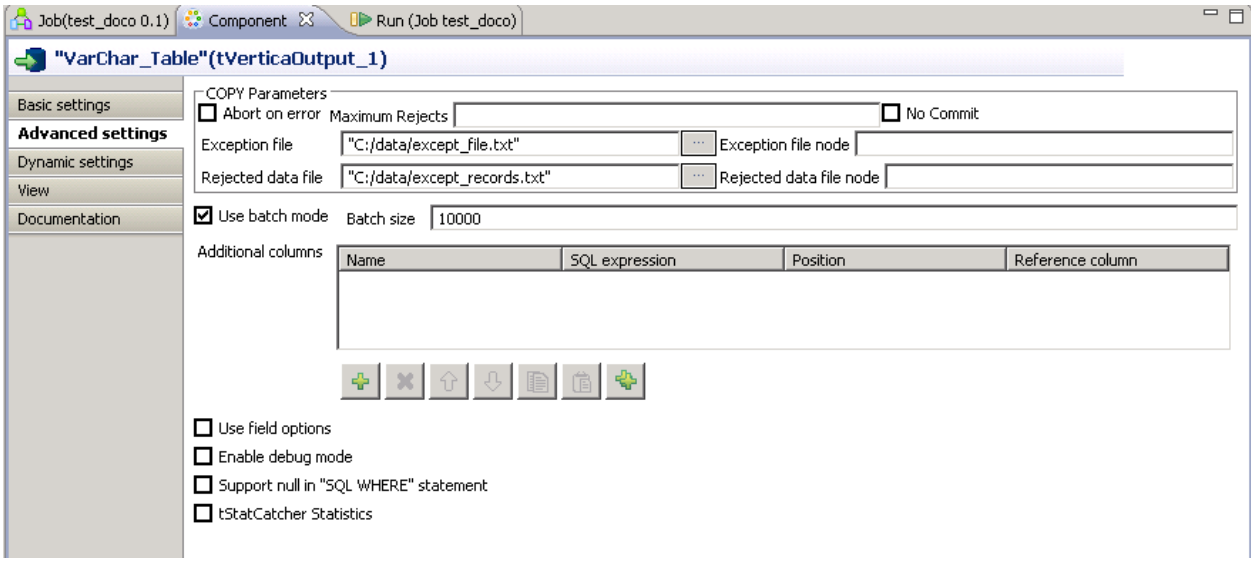
Basic Settings for Large, Bulk Loads (Writing to ROS)

For large, bulk loads, you should use the DIRECT syntax to write to ROS. To do so, you must set the **Action on data** option to COPY.



Advanced Settings for Large, Bulk Loads (writing to ROS)

Use Advanced Setting to specify file names and locations for exception files when using COPY.



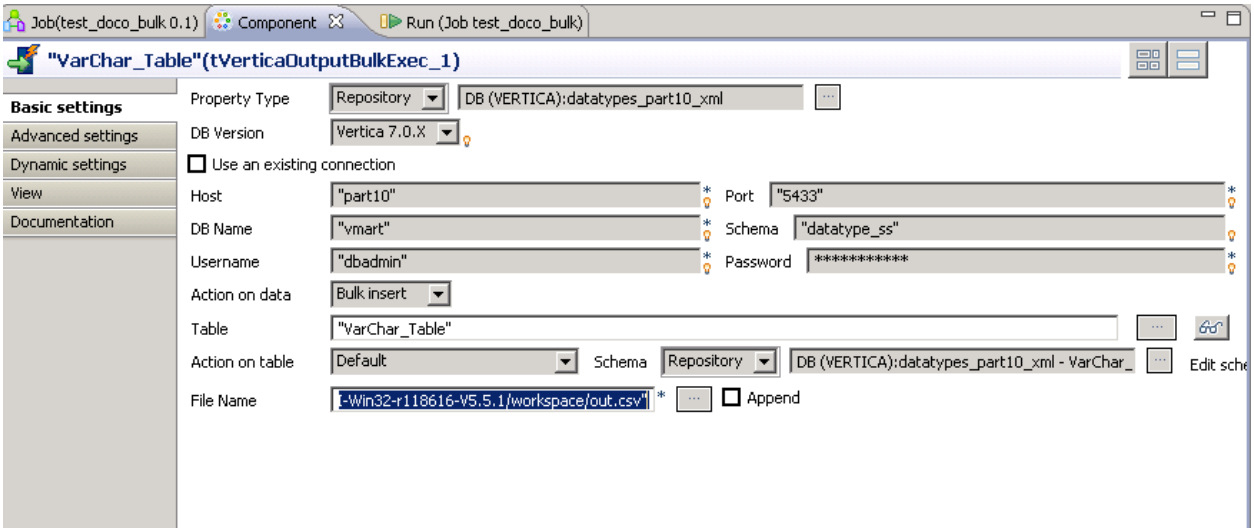
The settings above generate the following SQL:

```
2014-09-16 17:26:50.474 Init Session:0x7f0a4800fe30-13000000009c13e [Txn]
<INFO> Rollback Txn: 13000000009c13e 'COPY datatype_ss.VarChar_Table
(KeyColumn,Varchar_Column,Varchar_Max_Column) FROM local STDIN UNCOMPRESSED
WITH DELIMITER ';' RECORD TERMINATOR '' ENCLOSED BY '|' NULL AS '' EXCEPTIONS
'C:/data/except_file.txt' REJECTED DATA 'C:/data/except_records.txt' DIRECT
NO COMMIT'
```

tVerticaOutputBulkExec

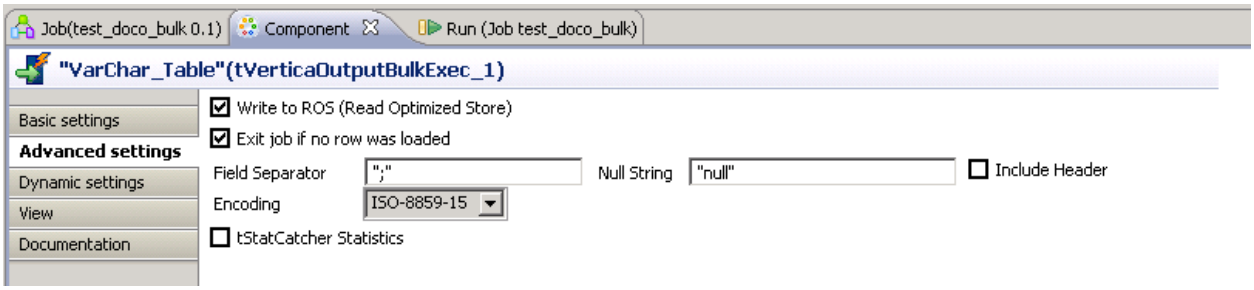
This component writes to a file and then copies the data using standard input (STDIN).

Basic Settings



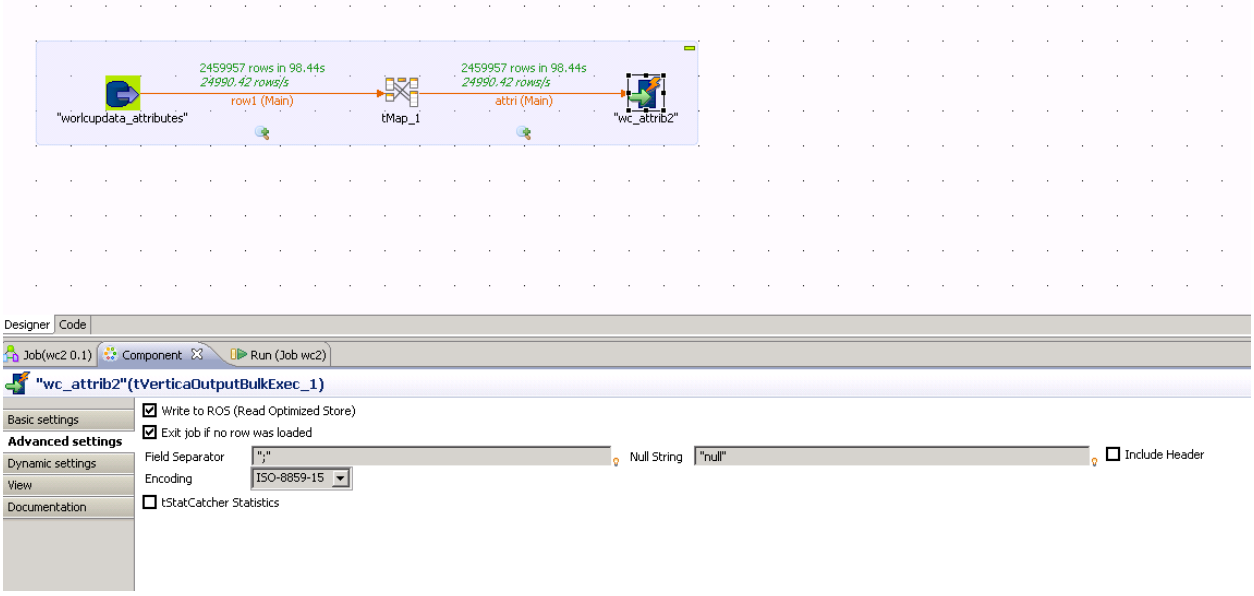
Advanced Settings

By default, tVerticaOutputBulkExec writes to ROS.



These settings result in the following SQL:

```
2014-09-16 17:46:14.526 Init Session:0x7f0a48010510-13000000009c1fe [Txn]
<INFO> Starting Commit: Txn: 13000000009c1fe 'COPY datatype_ss.VarChar_Table
FROM local STDIN DELIMITER ';' NULL 'null' DIRECT returnrejected'
```



tVerticaRow

The tVerticaRow component allows you to specify any valid Vertica SQL, including COPY statements. You can use tVerticaRow to load data into HP Vertica flex tables, and for any other scenarios that require Vertica structures that are not supported with custom Talend components.

Example

In the example below, the source file is on the HP Vertica server and uses the high performance HP Vertica COPY (not COPY FROM LOCAL.) You can load data in this way whenever the source files are on the HP Vertica cluster.

```
"COPY talend.store_target FROM '/home/dbadmin/store.csv' DELIMITER ';' NULL
'' DIRECT;"
```

Example: Loading into Flex Tables

The example below shows how to run the HP Vertica flex example included in the HP Vertica package directory.

To create the flex table:

```
create flex table mountains();
```

To load data to the flex table:

```
copy mountains from
‘/opt/vertica/packages/flextable/examples/mountains.json’ parser
fjsonparser();
```

To create the view of the flex table:

```
SELECT compute_flextable_keys_and_build_view('mountains');
```

Example: HP Vertica-to-HP Vertica COPY

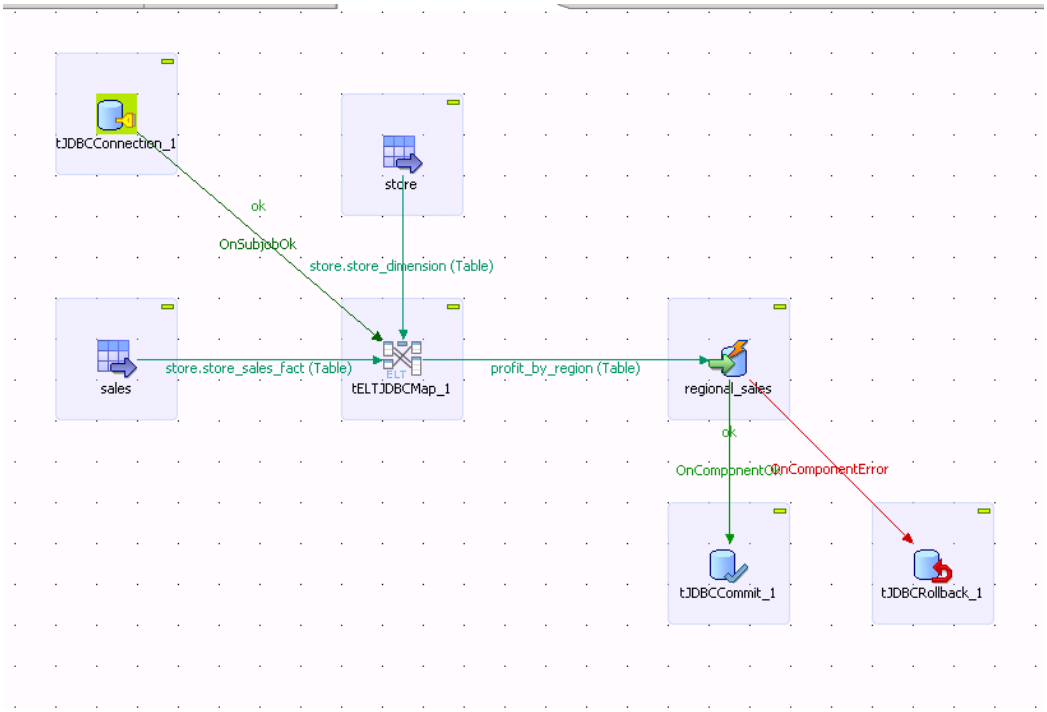
The example below shows an HP Vertica-to-HP Vertica COPY:

```
“CONNECT TO VERTICA vmart USER dbadmin PASSWORD '' ON 'VertTest01',5433;COPY
customer_dimension FROM  VERTICA vmart.customer_dimension DIRECT;”
```

You can also use this component to copy from HP Vertica to HP Vertica where no transformations are required. Note that you'll need to perform additional steps to define export addresses. Connecting to a public network requires some configuration. For information about using this statement to copy data across a public network, see Using Public and Private IP Networks in the HP Vertica documentation.

Using Generic Talend ELT Components with HP Vertica

The ELT family of components group together database connectors and processing components for ELT mode, where the target DBMS becomes the transformation engine. When possible, SQL is combined and processed in a single query on the database. The image below illustrates how Talend supports ELT with Vertica. This example uses the generic components for the entire job.



When this job runs, Talend generates the following SQL:

```
'INSERT INTO
store.profit_by_region(store_region,store_state,cost_dollar_amount,gross_prof
it_dollar_amount) (SELECT STORE.store_region , STORE.store_state ,
sum(SALE.cost_dollar_amount ), sum(SALE.gross_profit_dollar_amount ) FROM
store.store_dimension STORE INNER JOIN store.store_sales_fact SALE ON(
SALE.store_key = STORE.store_key ) group by STORE.store_region,
STORE.store_state order by STORE.store_region, STORE.store_state )'
```

As of 2014, Talend does not provide Vertica specific ELT components. You can use the generic components, but be aware the INSERT syntax will NOT be optimized. As you can see from the SQL in the example above, the /+*DIRECT*/ hint has not been added.

Other Components for Bulk Loading

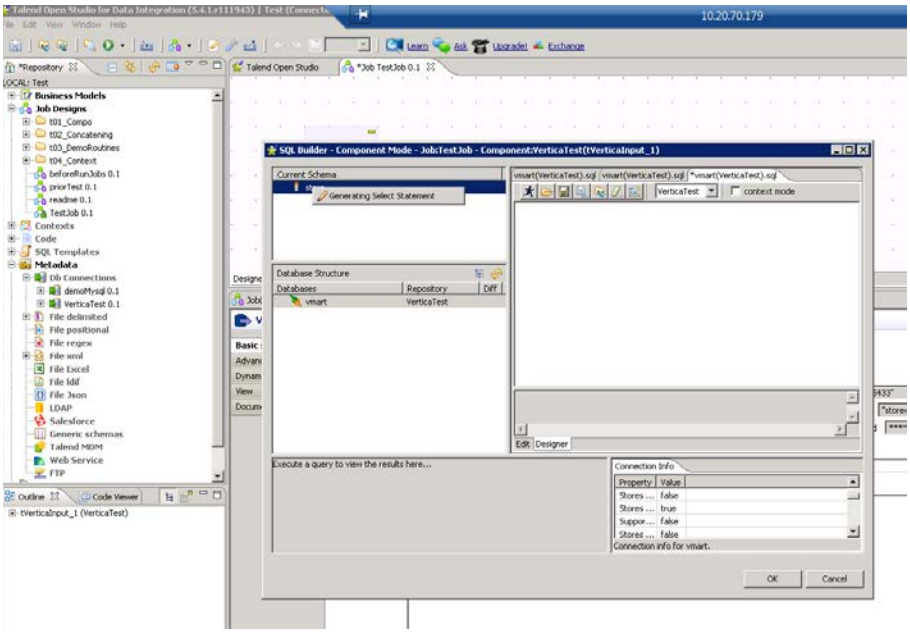
The tVerticaOutputBulk and tVerticaBulkExec components are generally used together as parts of a two-step process. The first step generates an output file. In the second step, the output file is used in the INSERT operation used to load data into a database.

You can also choose to do both of these steps at once using the tVerticaOutputBulkExec component. However, using tVerticaOutputBulk and tVerticaBulkExec allows the data to be transformed before it is loaded in the database.

Using the Talend SQL Builder

When using the SQL builder, be sure to include the schema name in the query, as shown below:

```
select * from store.store_dimension;
```



Enabling Parallelization in Talend

You can enable parallelization data flows by partitioning an input data flow of a Talend subjob into parallel processes and simultaneously executing these processes.

Enabling Parallelization in the Talend Enterprise Edition

If you are using the Talend Enterprise Edition, you can enable or disable the parallelization using a single click. Talend Studio then automates the implementation across a given job. See the links below to find out more about this feature.

- http://www.talend.com/products/data-integration/matrix
- https://help.talend.com/display/KB/How+to+automatically+enable+parallelization+of+data+flows+for+better+performance

Enabling Parallelization in the Talend Community Edition

If you are using the Talend Community Edition, you should add a WHERE clause to the original query to chunk the data. This example will result in 4 chunks.

```
original_sql + " and hash(" + primaryKey + ") % " + noOfThreads + " = " + i
```

Example:

```
select if.* from inventory_fact if, warehouse_dimension wd where
if.warehouse_key = wd.warehouse_key
```

The query above results in the 4 queries below:

```
select if.* from inventory_fact if, warehouse_dimension wd where
if.warehouse_key = wd.warehouse_key and hash(product_key, date_key) %  4 = 1;

select if.* from inventory_fact if, warehouse_dimension wd where
if.warehouse_key = wd.warehouse_key and hash(product_key, date_key) %  4 = 2;

select if.* from inventory_fact if, warehouse_dimension wd where
if.warehouse_key = wd.warehouse_key and hash(product_key, date_key) %  4 = 3;

select if.* from inventory_fact if, warehouse_dimension wd where
if.warehouse_key = wd.warehouse_key and hash(product_key, date_key) %  4 = 4;
```

You should choose keys that have equal distribution. For example, the two keys chosen in the example above provide the following counts:

```
Key : 235164 Value : product_key , Key : 50148 Value : date_key
count, chunk
7501441, 1
7500008, 2
7497028, 0
7501523, 3
```

HP Vertica Tips and Techniques

This section discusses considerations, modifications, and best practices for using HP Vertica with Talend DI. It contains these sections:

Managing HP Vertica Resources

To prevent query users from being impacted by ETL jobs, you should create a separate resource pool for Talend. Your resource pool settings should depend on the amount of memory on the machine, as well as other factors, such as how many other resource pools are created for other users.

Example:

```
CREATE RESOURCE POOL batch_pool MEMORYSIZE ‘4G’ MAXMEMORYSIZE ‘84G’
   MAXCONCURRENCY 36;
drop user talend;
drop schema talend;
CREATE SCHEMA IF NOT EXISTS talend;
CREATE user talend identified by 'talend@pwd' SEARCH_PATH talend;
GRANT USAGE on SCHEMA talend to talend;
GRANT USAGE on SCHEMA PUBLIC to talend;
GRANT USAGE on SCHEMA online_sales to talend;
GRANT USAGE on SCHEMA store to talend;
GRANT SELECT on ALL TABLES IN SCHEMA PUBLIC to talend;
GRANT SELECT on ALL TABLES IN SCHEMA store to talend;
GRANT SELECT on ALL TABLES IN SCHEMA online_sales to talend;
GRANT CREATE ON SCHEMA talend to talend;
GRANT USAGE ON RESOURCE POOL batch_pool to talend;
ALTER USER talend RESOURCE POOL batch_pool;
```

Managing HP Vertica Storage Containers

HP Vertica supports INSERT, UPDATE, DELETE, and bulk load operations (COPY), intermixed with queries in a typical data warehouse workload. The storage model consists of three elements that operate identically on each HP Vertica node:

- *Write Optimized Store (WOS)* is a memory-resident data structure for storing INSERT, UPDATE, DELETE, and COPY (without /*+DIRECT*/ hints) actions. To support very fast data load speeds, the WOS stores records without data compression or indexing. The WOS organizes data by epoch and holds both committed and uncommitted transaction data.
- *Read Optimized Store (ROS)* is a highly optimized, read-oriented, disk storage structure. The ROS makes heavy use of compression and indexing. You can use the COPY…DIRECT and INSERT (with /*+DIRECT*/ hints) statements to load data directly into the ROS.
- *The Tuple Mover (TM)* is the Vertica Analytic Database database optimizer component that moves data from memory (WOS) to disk (ROS). The Tuple Mover runs in the background, performing some tasks automatically at time intervals determined by its configuration parameters.

For more information on HP Vertica ROS and WOS, see Loading Data into the Database, in HP Vertica Best Practices for OEM Customers.

Managing HP Vertica ROS Containers

HP Vertica creates ROS containers:

- With every Moveout
- Each time COPY DIRECT is executed
- Whenever a table is partitioned

ROS containers consume system resources and so their number has a direct effect on system performance. A large number of these can degrade performance, while too few ROS containers can prevent the system from taking advantage of inherent parallelism. Too many ROS containers can also result in Vertica errors (See Issues for more details.)

For best performance, HP Vertica recommends:

- 10 containers per projection when not loading data
- Up to 20 containers per projection during data load
- No more than 50 containers per projection

ROS Container Errors

ROS container exceptions can occur if you accidently target frequent, small loads directly to ROS instead of WOS. This can occur if the batch load API does not correctly convert INSERTS into a COPY statements. (You can check the HP Vertica logs to see whether this is the case.)

Example

```
ERROR:  Too many ROS containers exist for the following projections:
customer_100061.ark_DBD_1_seg_ptoole2_ptoole2 (limit = 1,000,000, ROS files =
999673, DV files = 0, new files = 479)
HINT:  Please wait for the tuple mover to catch up.  Use 'select * from
v_monitor.tuple_mover_operations;' to monitor.
```

Best Practices for Loading and Updating Data

You can avoid creating too many or too few ROS containers by following these recommendations:

- Load and delete data less frequently in larger batches**.**
- If you are working with many small files of raw data, use a single COPY statement to load these files at one time. Use the wildcard support to COPY from all files, or by concatenating all files to a named pipe and having COPY load from the pipe.
- Use a single DELETE/UPDATE statement to delete or update batches of rows**.** When possible, delete/update batches of rows at a time to avoid creating too many delete vector ROS containers.

Managing HP Vertica WOS Containers

The WOS can use a maximum of 25% of the physical memory on each node. HP Vertica issues a WOS overflow error when you reach or exceed this limit. For example, if you have up to 1GB available for your WOS, you'll receive a WOS Overflow error if you fill the 1GB before the data can be processed by the tuple mover's Moveout operation.

Monitoring Data Load

You can monitor the progress of running jobs using either standard SQL or by using the HP Vertica Management Console.

Using SQL to Monitor Load Streams

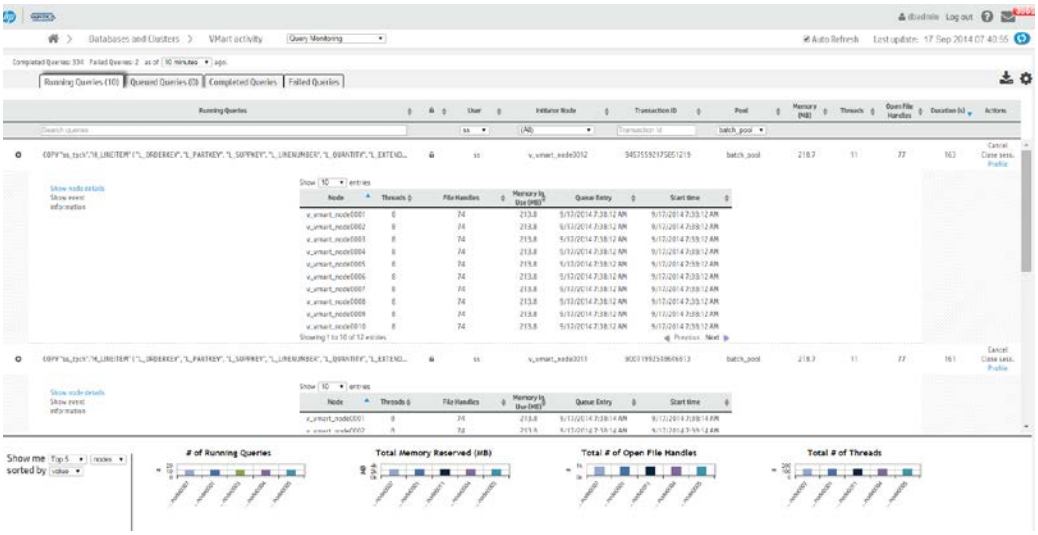
You can use the SQL statements below to view how your data load is progressing.

```
\echo ..Load streams
select
table_name as table,
stream_name as stream,
to_char(date_trunc('second', current_timestamp - load_start::timestamp),
'hh24:mi:ss') as run_time,
to_char(load_start::timestamp, 'yyyy-mm-dd hh24:mi') as load_start,
to_char(accepted_row_count, '999,999,999,999') as accepted,
to_char(rejected_row_count, '999,999,999,999') as rejected,
to_char(unsorted_row_count, '999,999,999,999') as unsorted,
to_char(sorted_row_count, '999,999,999,999') as sorted,
sort_complete_percent as sort_pct,
to_char(accepted_row_count/extract(epoch from current_timestamp -
load_start::timestamp), '999,999') as rps,
to_char(sorted_row_count/extract(epoch from current_timestamp -
load_start::timestamp), '999,999') as sort_rps
from load_streams ls;
--
--
\echo ..Load Totals
select
lpad(count(distinct table_name)::char, max(length(table_name))) as tables,
```

```
lpad(count(distinct stream_name)::char, max(length(stream_name))) as streams,
to_char(date_trunc('second', current_timestamp - min(load_start::timestamp)),
'hh24:mi:ss') as run_time,
to_char(min(load_start), 'yyyy-mm-dd hh24:mi') as load_start,
to_char(sum(accepted_row_count), '999,999,999,999') as accepted,
to_char(sum(rejected_row_count), '999,999,999,999') as rejected,
to_char(sum(unsorted_row_count), '999,999,999,999') as unsorted,
to_char(sum(sorted_row_count), '999,999,999,999') as sorted,
sum(sort_complete_percent) as sort_pct,
to_char(sum(accepted_row_count)/extract(epoch from current_timestamp -
min(load_start::timestamp)), '999,999') as rps
from load_streams ls;
```

Using Management Console to Monitor Running Jobs

You can use HP Vertica Management Console to better understand resource usage when jobs are executing. The following screenshot shows various running COPY statements as well as the resources consumed by the ETL job. Listed below is an example of a batch load job running and the resources that are consumed.



Downloading and Installing HP Vertica Management Console

You can download the version of Management Console you need from my.vertica.com. Follow the installation instructions in the *HP Vertica Installation Guide*. Click here for the HP Vertica 7.1.x installation instructions.

After the installation, you can access Management Console at this URL:

http://www.vertica.com/?s=console

Additional Information on Management Console

HP Vertica Core documentation:

- http://www.vertica.com/documentation
- Installation
- Management Console Overview
- Management Console videos:
    - http://www.vertica.com/customer-experience/vertica-101/managment-console/
    - http://www.vertica.com/customer-experience/vertica-101/managment-console/
    - http://www.vertica.com/customer-experience/vertica-101/managment-console/

Using Linux to Monitor Disk and Resource Usage

Use these Linux commands to monitor disk and resource usage.

Command	Description
df	Use this command to monitor the disk to ensure enough space.
vmstat	Use this command to report to monitor CPU usage at specified intervals. For example, run vmstat 60 to obtain a report every 60 seconds. During the load, you will see high CPU usage in bursts for the sequence of memory sorts in the first part of chunk processing and high block I/O (vmstat columns *bi* and *bo*) during the rest of the time Here *bi* stands for *blocks in* and *bo* for blocks out, and these are reported in KB/sec. The system should not become idle until the load is done, where idle means low CPU use *and* low bi + bo. The maximum observed bi + bo gives an estimate of the disk bandwidth (KB/sec) used in the load, including any temporary disk. The `vmstat` display also shows swapping I/O using columns *si* (*swap in*) and *so (swap out)*, also in KB/sec. If the swapping I/O is significant (si + so > 20% of maximum-seen bi + bo, thus stealing up to 20% of the disk bandwidth) over many minutes, especially during the high block I/O periods, it means the system is under stress. In this case, you should reduce the parallelism of the load at the earliest opportunity, by reducing the number of load streams.

Validating Load Results

HP Vertica does NOT enforce constraints at data load. Constraints are enforced when data is loaded into a table with a pre-joined dimension, or when the table is joined to a dimension table during a query. Therefore, you could experience constraint errors in any of these scenarios:

- If there is not exactly one dimension row that matches each foreign key value
- An inner join query is processed. An outer join is treated as an inner join due to the presence of foreign key.
- A new pre-join projection anchored on the table with the foreign key constraint is refreshed

Note: If you are testing your queries for performance, you may not see the full query speed right away, because HP Vertica will be busy reorganizing the data for several hours depending on how much data was loaded and the type of load.

Exporting Data from the Source Database to Flat Files

If, for any reason, you cannot connect directly to the source database, you may need to export data from the source system to flat files. When doing, so consider the following:

- Smaller tables generally fit into a single load file. Split any large tables into 250-500GB load files. For example, a 10 TB fact table requires 20-40 load files to maintain performance.
- The default delimiter for the COPY statement is a vertical bar (|). Before loading your data, make sure that no CHAR(N) or VARCHAR(N) data values include this delimiter character. To test for the existence of a specific character in a column, use a query such as this:

  ```
  SELECT COUNT(*) FROM T WHERE X LIKE '%|%'
  ```

  If only a few rows contain |, you can eliminate them from the load file using a WHERE clause and load the rows separately using a different delimiter.

Expected Results for Various Scenarios

This table describes the results for various situations depending on the type of SQL generated by Talend.

Mapping Exceptions	Vertica COPY LOCAL	SQL INSERT
CHAR field with n characters mapped to CHAR column with less than n characters	Record inserted with silent truncation	Record not inserted; Warning message issued
EN (Edited Numeric) field mapped to INTEGER column with data that forces numeric overflow	Record is rejected and sent to rejected records log file	0 is silently loaded in place of number that would cause overflow
EN field mapped to NUMERIC column with data that exceeds scale	Record is rejected and sent to rejected records log file;	Record with value that exceeds scale is silently rejected

Data Type Mappings

Although Talend has the ability to generate Vertica DDL automatically when moving data from a non-HP Vertica database to HP Vertica, you should have a solid understanding of some of the differences between the two databases. For instance, Oracle supports NVARCHAR data types and HP Vertica does not.

For SQL data types supported by HP Vertica, see the SQL Reference Manual. Contac t HP Vertica for example scripts for converting DDL from SQL Server and Oracle to Vertica.

UTF-8 Considerations

If the source system contains multi-byte character data, then when converting the DDL from the source to the HP Vertica target, you should increase the size of the column by a factor of 3 (at least) to allow for the possibility of non-UTF 8 character sets. For example, you should change a NVARCHAR(4) source column DDL varchar(12) in HP Vertica.

HP Vertica and Oracle Data Type Mappings

Oracle uses proprietary data types for all main data types (for example, VARCHAR, INTEGER, FLOAT, and DATE.) When migrating to HP Vertica, follow these guidelines:

- If your data is multi-lingual, you should convert your schema before migrating. Converting your schema before migrating can minimize errors and minimize time lost spent fixing erroneous data issues.
- You must convert the non-standard NUMBER data type to the SQL-standard INT or INTEGER.
- You can include the necessary foreign key clauses in the table definitions themselves or in separate `ALTER TABLE T ADD CONSTRAINT …` commands. The foreign key constraints play an important role in guiding the HP Vertica Database Designer in its work.

You can find data type mappings for HP Vertica and Oracle databases in the HP Vertica SQL Reference Manual.

Using Vertica COPY

The COPY statement bulk loads data into an HP Vertica database. You can load one or more files, or pipes on a cluster host. You can also load directly from a client system, using the COPY statement with its FROM LOCAL option.

Raw input data must be in UTF-8, delimited text format. Data is compressed and encoded for efficient storage. See the *HP Vertica Administrators Guide* for more details on COPY command.

Loading Over the Network

A 1Gbps (gigabits per second) network can deliver about 50 MB per second, or 180 GB per hour. HP Vertica can load about 30-50 GB per hour/node for a 1-Ksafe projection design. Therefore, you should use a dedicated 1Gbps LAN. Using a LAN with a performance that is less than 1Gbps will be proportionally slower. HP Vertica recommends not loading data across an external network, because the delays over distance slow down the TCP protocol to a small fraction of its available bandwidth, even without competing traffic.

Note: The actual load rates you obtain can be higher or lower depending on the properties of the data, number of columns, number of projections, and hardware and network speeds. You can improve load speeds further by using multiple parallel streams.

Using Non-ASCII data

HP Vertica stores data in the UTF-8 compressed encoding of Unicode. The resulting UTF-8 codes are identical to ASCII codes for the ASCII characters (codes 0 to 127 in one byte). If your table data (CHAR columns) is all ASCII, it should be easy to transport, since all current operating systems treat ASCII in the same way. If you have UTF-8 data, be sure that the extraction method you use does not convert CHAR column values to the current (non-UTF-8) locale of the source system.

On most UNIX systems, you can use the locale command to see the current locale. You can change the locale for a session by setting the LANG environment variable to en_US.UTF-8. If you have data in another character encoding such as Latin-1 (ISO 8859), you must convert it to UTF-8 if your data is actively using the non-ASCII characters of Latin-1 such as the euro sign and the diacritical marks of many European languages. You can use the Linux tool iconv to perform the necessary conversions. Large fact tables are unlikely to contain non-ASCII characters, so these conversions are usually needed only for the smaller dimension table's data.

Internationalization

HP Vertica supports the following internationalization features. See the HP Vertica Administrators Guide for more information on internationalization.

Unicode Character Encoding: UTF-8 (8-bit UCS/Unicode Transformation Format)

All input data received by the database server is expected to be in UTF-8 format, and all data output by HP Vertica is in UTF-8. The ODBC API operates on data in UCS-2 on Windows systems, and in UTF-8 on Linux systems. (The A UTF-16 ODBC driver is available for use with the DataDirect ODBC manager.) JDBC and ADO.NET APIs operate on data in UTF-16. The HP Vertica client drivers automatically convert data to and from UTF-8 when sending to and receiving data from Vertica using API calls. The drivers do not transform data loaded by executing a COPY or COPY LOCAL statement.

Locales

The locale parameter defines the user's language, country, and any special variant preferences, such as collation. HP Vertica uses the locale to determine the behavior of various string functions as well for collation for various SQL commands that require ordering and comparison; for example, GROUP BY, ORDER BY, joins, the analytic ORDER BY clause, and so forth.

By default, the locale for the database is `en_US@collation=binary` (English US). You can establish a new default locale that is used for all sessions on the database, as well as override individual sessions with different locales. You can also set the locale through ODBC, JDBC, and ADO.net.

Keep the following in mind when working with locales:

- Projections are always collated using the en_US@collation=binary collation regardless of the session collation. Any locale-specific collation is applied at query run time.
- The maximum length parameter for VARCHAR and CHAR data type refers to the number of octets (bytes) that can be stored in that field and not number of characters. When using multi-byte UTF-8 characters, the fields must be sized to accommodate from 1 to 4 bytes per character, depending on the data.
- When the locale is non-binary, you can use the collation function to transform the input to a binary string that sorts in the proper order. This transformation increases the number of bytes required for the input according to this formula :

```
result_column_width = input_octet_width * CollationExpansion + 4
```

(CollationExpansion defaults to 5)

Known Issues

This section describes known issues in HP Vertica/Talend connections. This list was last updated on 12/9/14.

Case for schema names

Schema names in HP Vertica are case-insensitive. However, in Talend, you must provide the exact case of schema name.

Schema name not provided with Generic JDBC connection

The Generic JDBC Connection does not append the schema name with the table name; therefore, if you use this connection, jobs will fail in Talend.

Vertica Data Type Mapping and Min-Max Values

We have identified several data type transfer issues in our testing. In the cases listed below, all the rows are transferred, but the data precision is lost for extremely large values. This happens for INTEGER, DOUBLE, AND DECIMAL data types. For more information, see the Talend known issue below:

https://jira.talendforge.org/browse/TDI-29446

To ensure the values described in this issue transfer correctly, you should map FLOAT to DOUBLE.

The issue described above also occurs with INTEGER. As you can see, the default mapping for our INTEGER data types is INTEGER. (Note that, if the type is changed to LONG, the data type is changed to BIGINT and the transfers will be accurate.)

Editing Data Type Mappings

To edit the mappings in Talend Open Studio, navigate to **Window > Preferences > Talend > Specific Settings > Metadata of TalendType**.

Note: Toucan find the default mappings file (mapping_vertica.xml) for HP Vertica here:

```
<Talend_Install_Dir>/
configuration\org.eclipse.osgi\bundles\1707\1\.cp\mappings
```

```
C:\unzipped\talend_di\TOS_DI-Win32-r118616-
V5.5.1\configuration\org.eclipse.osgi\bundles\1707\1\.cp\mappings
```

You can make changes to this file, but you should contact Talend support for guidance.

https://help.talend.com/display/KB/Changing%20the%20default%20data%20type%20mapping