# Transpara tStore

Data Extractor – Open Database Connectivity (ODBC)

# 1 ODBC Extractor (tODBC)

Transpara tStore Suite

## 1.1 Overview

We use two terms when we talk about moving data from place to place

- Interface
  - An interface is a connection for on-demand gathering of data. This is the long-standing terminology used by for VisualKPI.
  - Note that some data is "cached" in VisualKPI for performance or other reasons. This is not the same as long-term storage; the cache is only available to specific applications.
- Extractor
  - This is the term that we use for moving data from a foreign system to tStore for longer-term storage and general availability of the data.

## 1.2 ODBC

Microsoft Open Database Connectivity is a standard, widely-adopted software layer that provides SQL access to a huge variety of databases, systems and equipment.

The Transpara ODBC extractor leverages this capability to pull data, on a schedule, from any configured data source, and deliver the data to tStore.

The Transpara ODBC Extractor consists of:

- a graphical user interface for construction, testing and configuration of ODBC queries
- a Windows service that supports the above GUI
- a Windows service that executes the queries

Note:

Tested using:

- Python 3.9.1
- Python 3.10.2

# 2 Transpara ODBC Installation

Transpara tODBC consists of both client and server components. The client must be installed on Microsoft Windows. The server can be installed on either Windows or as a Linux Docker Container.

The Linux container supports a limited array of ODBC drivers including Microsoft SQL Server and PostgreSQL. The Windows installation supports all appropriate ODBC drivers.

## 2.1 Microsoft Windows

This installation procedure is for the interim bridge release of the Transpara ODBC Extractor. The production installation procedure will be integrated with the server manager.

The INTERIM tODBC installation kit is a single zip file with the following contents:

| File | Description |
|---|---|
| Extractor Manager (and associated DLLs, etc.) | The INTERIM GUI for configuring and testing ODBC queries. |
| config.py | Internal module for Python script configuration. |
| install_services.bat | Batch file to install Windows services. |
| nssm.exe | Executable to support running Python scripts as Windows services. |
| odbc.md | This document |
| odbc.py | The Transpara ODBC Extractor |
| odbc_api.py | The API that supports the INTERIM GUI and targeted final GUI. |
| output_tstore.py | The Python module shared by all Transpara Extractors for delivering data to tStore through the tStore API. |
| python*.exe | For convenience, one or more compatible versions of Python for Windows. |
| requirements.txt | a list of required Python modules for use with pip install -r |

### 2.1.1 Preparation

- Identify target tStore host ("tStore")
  - reachable by network from the ODBC Extractor node
  - allows connections on the tStore API Port (default 10001)
- Identify Windows server for the ODBC Extractor ("Windows Server")

  - can reach the tStore instance by network
  - service account
    - administrator access
    - interactive login rights (during the installation, at least)
  - can reach any required ODBC data sources
  - has all required ODBC drivers for desired connections
- Python, if required, install Python (3.9.1 is the minimum)

```
python -V
python-3.9.1-amd64.exe /quiet InstallAllUsers=1 PrependPath=1 Include_t
est=0
```

- Install required Python modules:

```
python -m pip install -r requirements.txt
```

### 2.1.2 Folders and Files

- log in to the Windows Server using the service account with administrative privileges
- create a dedicated folder ("tODBC Folder"), for example:

```
mkdir "c:\Program Files\Transpara\tODBC\"
```

- put the installation kit from Transpara (odbcExtractorXXX.zip) into the tODBC folder
- extract all files in the archive into the tODBC folder

### 2.1.3 Command Line

The following steps need to be performed from an administrator command window:

- open an administrator command window (cmd.exe) (right-click, run as Administrator)
- change directory to the tODBC Folder

### 2.1.4 Create and Start Windows Services

Two Microsoft Windows Services will be created and configured to start automatically when the server boots. Note that a logon account with appropriate privileges must be used.

```
install_services.bat
```

*Note: You will be prompted for service account credentials, tStore API URL and the desired TCP/IP port.*

*Note: Examine the api.err, api.out, odbc.err and odbc.out files for troubleshooting issues with the API or the ODBC Extractor.*

## 2.2 Linux

Installation on Linux leverages standard Docker and docker-compose using the following files:

- docker-compose.yaml
- .env

Note that you must load the tODBC container image from a tar file or directly by pulling it from dockerhub.com (with appropriate permissions).

- Edit the .env as appropriate

```
TODBC_ID=95bb8ff-ca57-4043-8363-5e345876f3bc
TODBC_TAG=1.0.3
TSTORE_API_URL=http://tstore_api:80/api/v1/write
TODBC_API_PORT=5000
NAME=todbc1
```

| Parameter | Description |
|---|---|
| TODBC_ID | internal use only by Transpar |
| TODBC_TAG | identifies the version of the image that will be used. Do not modify unless instructed to by Transpara technical support. |
| TSTORE_API_URL | Edit this to point to the target Transpara tStore instance. If the tODBC container will run on the same network as tStore, the node name can be used. Otherwise, specify a FQDN or IP Address. |
| TODBC_API_PORT | The default value for the GUI to connect. Change only if required. |

With the .env file in place, issue the following command:

```
docker-compose up -d
```

# 3 Data Sources

### 3.1.1 Data Source Name (DSN)

Configure one or more DSNs and ensure that they are correct by testing the connection. The DSN will be used to identify which data source should be used for any particular query in the ODBC Extractor.

### 3.1.2 Connect Strings

It is also possible to use a connect string without a configured DSN. For example:

- Linux SQL Server

```
Driver={ODBC Driver 17 for SQL Server}; Server=sqlserver.cadman.transpara.com
; DataBase=demodata; UID=xxxxxx; PWD=xxxxxxxx
```

- Windows SQL Server

```
SQL ServerDriver={SQL Server}; Server=sqlserver; DataBase=demodata; Integrate
d Security=SSPI;
```

- Linux PostgreSQL

```
Driver={PostgreSQL Unicode};Server=10.10.0.224;Port=5435;Database=postgres;Ui
d=postgres;Pwd=BorgGoesLive!22;
```

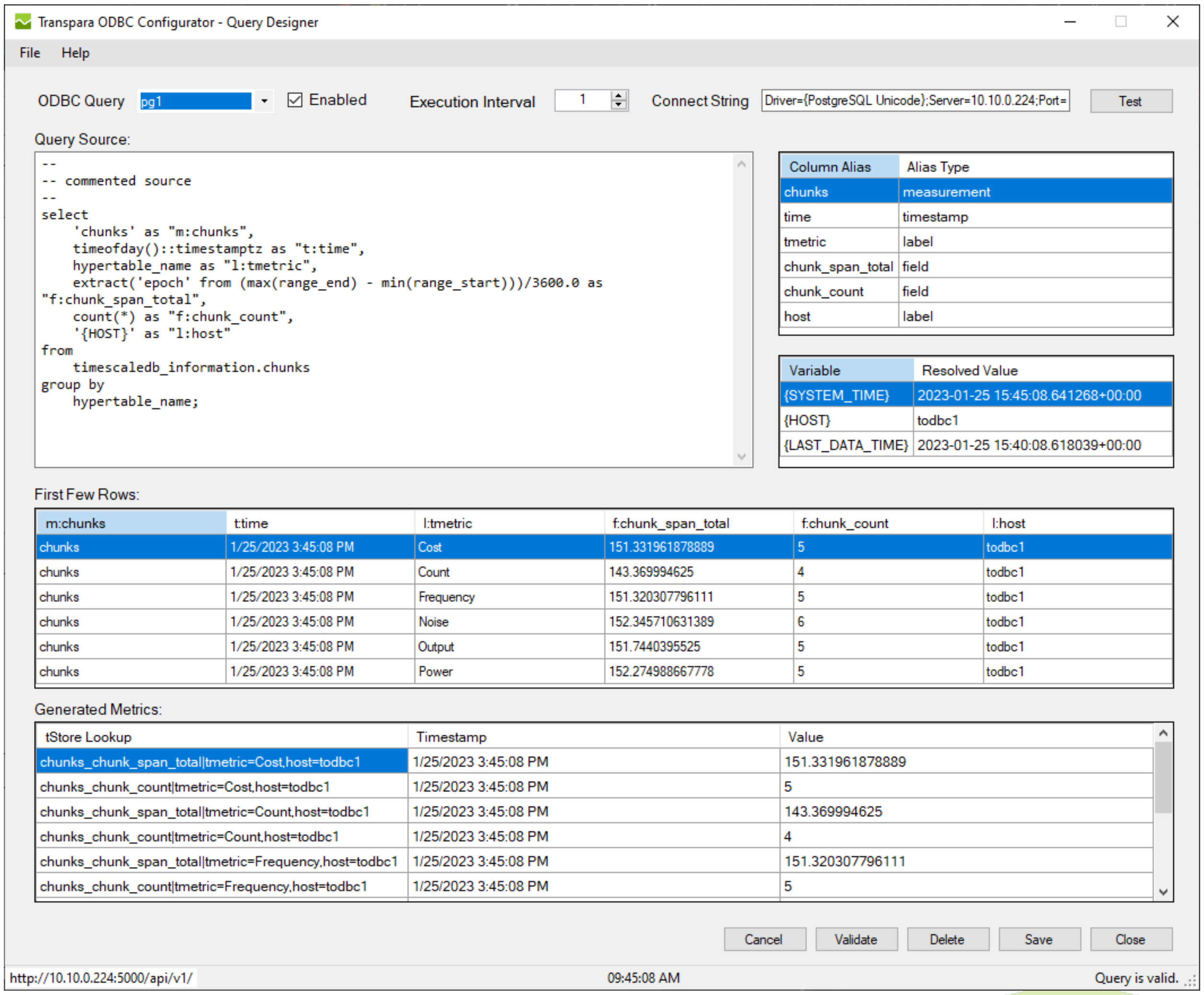
## 3.2 Additional Linux Drivers

These have not been tested by should be compatible.

- Oracle
  - oracle-instantclient12.1-basiclite-12.1.0.2.0-1.x86_64.rpm
  - oracle-instantclient12.1-odbc-12.1.0.2.0-1.x86_64.rpm
- MySQL
  - mysql-connector-odbc_8.0.32-1ubuntu22.10_amd64.deb

# 4 Extractor Manager GUI

This Windows application is used to configure, test and schedule ODBC queries to deliver data to tStore.



ODBC Query pg1 [✓] Enabled Execution Interval 1 Connect String Driver={PostgreSQL Unicode};Server=10.10.0.224;Port=

Query Source:

```
--
-- commented source
--
select
    'chunks' as "m:chunks",
    timeofday()::timestamptz as "t:time",
    hypertable_name as "l:tmetric",
    extract('epoch' from (max(range_end) - min(range_start)))/3600.0 as
"f:chunk_span_total",
    count(*) as "f:chunk_count",
    '{HOST}' as "l:host"
from
    timescaledb_information.chunks
group by
    hypertable_name;
```

| Column Alias | Alias Type |
|---|---|
| chunks | measurement |
| time | timestamp |
| tmetric | label |
| chunk_span_total | field |
| chunk_count | field |
| host | label |

| Variable | Resolved Value |
|---|---|
| {SYSTEM_TIME} | 2023-01-25 15:45:08.641268+00:00 |
| {HOST} | todbc1 |
| {LAST_DATA_TIME} | 2023-01-25 15:40:08.618039+00:00 |

First Few Rows:

| m:chunks | t:time | l:tmetric | f:chunk_span_total | f:chunk_count | l:host |
|---|---|---|---|---|---|
| chunks | 1/25/2023 3:45:08 PM | Cost | 151.331961878889 | 5 | todbc1 |
| chunks | 1/25/2023 3:45:08 PM | Count | 143.369994625 | 4 | todbc1 |
| chunks | 1/25/2023 3:45:08 PM | Frequency | 151.320307796111 | 5 | todbc1 |
| chunks | 1/25/2023 3:45:08 PM | Noise | 152.345710631389 | 6 | todbc1 |
| chunks | 1/25/2023 3:45:08 PM | Output | 151.7440395525 | 5 | todbc1 |
| chunks | 1/25/2023 3:45:08 PM | Power | 152.274988667778 | 5 | todbc1 |

Generated Metrics:

| tStore Lookup | Timestamp | Value |
|---|---|---|
| chunks_chunk_span_total\|tmetric=Cost,host=todbc1 | 1/25/2023 3:45:08 PM | 151.331961878889 |
| chunks_chunk_count\|tmetric=Cost,host=todbc1 | 1/25/2023 3:45:08 PM | 5 |
| chunks_chunk_span_total\|tmetric=Count,host=todbc1 | 1/25/2023 3:45:08 PM | 143.369994625 |
| chunks_chunk_count\|tmetric=Count,host=todbc1 | 1/25/2023 3:45:08 PM | 4 |
| chunks_chunk_span_total\|tmetric=Frequency,host=todbc1 | 1/25/2023 3:45:08 PM | 151.320307796111 |
| chunks_chunk_count\|tmetric=Frequency,host=todbc1 | 1/25/2023 3:45:08 PM | 5 |

http://10.10.0.224:5000/api/v1/ 09:45:08 AM Query is valid.

## 4.1 Main Form

Screen elements are described in the table below from left to right and from top to bottom.

| Screen Element | Description and Usage |
|---|---|
| ODBC Query | Dropdown list containing the set of ODBC queries that have been configured. |
| Enabled | Checkbox to indicate whether the ODBC query should be processed by the ODBC Extractor engine. If this is not checked, no data will be delivered to tStore. Note, it is not possible to enable a query until the query has been validated. |
| Connect String | A text box containing the connect string for the selected query. Note that the connect string can be of several different including simply "dsn=dsnname" |

| Screen Element | Description and Usage |
|---|---|
| Test | A button to the right of the Connect String. Pressing this button will attempt to connect to the ODBC data source as specified in the connect string. The status bar will reflect the outcome of the connectivity test with "Connection successful" or otherwise. |
| Query Source | SQL query to be presented to the ODBC data source. A query cannot be used until it is "validated", "enabled" and "saved". |
| Column Aliases | This grid displays the selection list with the tODBC interpretation of the column aliases (e.g., timestamp, field, label, measurement). Double click on any row to change the alias type for the selected column. |
| Substitution Variables | A list of the available substitution variables along with their resolved values at the time of query validation. |
| First Few Rows | A handful of rows of actual data which are the the result of executing the ODBC query against the ODBC data source. |
| Generated Metrics | This grid indicates how the data will be represented within tStore. Note that it is possible to copy a "lookup" (for use with other tools) by highlighting a lookup and pressing Control-C. |
| Buttons | Cancel - Cancels all modifications made to the on-screen query and reloads the queries as currently stored in the database. |
| | Validate - |
| | Delete - |
| | Save - |
| | Close - |
| Menu Bar | File, New - |
| | File, Save - |
| | File, Save As - |
| | File, Connect ODBC API - |
| | File, Export Queries - |
| | File, Quit - |
| | Help, ABout - |

## 4.2 Typical Work Flows

### 4.2.1 Creating a New Query

| Action | Details |
|---|---|
| Enter Connect String | |
| Press Test | |
| Enter Query Source | |
| Press Validate | |
| Adjust Column Aliases | |
| Examine First Few Rows | |
| Examine Generated Metrics | |
| Set Execution Interval | |
| Enable Query | |
| Save Query | |

# 5 ODBC Queries

The Transpara tODBC engine executes on a frequency and processes a set of preconfigured ODBC SQL Queries. Each query has an associated Connect String and an execution interval (in minutes).

In general, any SQL query can be used by the Transpara ODBC Extractor ("tODBC Queries"). All columns returned by tODBC queries *must* be aliased using a specific two character prefix, described below.

### 5.1.1 Column Aliases

Transpara ODBC Extractor Queries require specific column aliases. These aliases indicate how that particular column should be interpreted by the Extractor.

| Column Alias Tag | Meaning | Required |
|---|---|---|
| **t**:*timestamp_name* | Timestamp column; This will be interpreted as the timestamp value. | One and only one |
| **f**:*field_name* | Field column. Each field column represents a value to be stored in tStore. Multiple field columns can be configured, but there must be at least one. | At least one. |
| **m**:*metric_name* | The measurement or metric name. This will be prepended to each field in creating the tStore metric. | Optional. |
| **l**:*label_name* | Labels are used to differentiate recordings and contain metadata about the recording. This may include a tagname, plant area, analysis type. It is also common to indicate the source of the data. | Optional. |

### 5.1.2 Substitution Parameters

Substitution parameters can be used anywhere in your SQL query, and as the name implies, will be replaced with the correct data when the query is executed. Substitution parameters are enlocsed in curly braces, e.g., {HOST}.

| Parameter | Description |
|---|---|
| **HOST** | The hostname of the machine running the ODBC engine. Note that this is likely not the hostname of the origina of the data (which is identified in the connect string or DSN.) |
| **LAST_DATA_TIME** | tODBC keeps track of the most recent piece of data retrieved for each query executed. In this way it is possible to only retrieve newer values with each execution. |

### 5.1.3 SQL Best Practices

It is a good idea to limit the number of rows that may be potentially returned by your SQL query. There is no limit to the number of values that can be stored in tStore (subject to disk space). Of course, the more rows returned, the longer it will take to process and digest.

Limiting the number of rows can be done with a subsitution parameter (LAST_DATA_TIME). It is also wise to include a *limit* clause in your SQL to set the absolute maximum number of rows that may be returned.

Note that the most recent data is returned by including an *order by* clause in the SQL.

## 6 Sample Queries

- Transpara Extracted OSIsoft PI Tags (PostgreSQL)

```
select
    'pitags' as "m:pitags",
    time as "t:time",
    value as "f:value",
    val(point_id) as "l:tagname",
    val(server_id) as "l:server"
from
    tcache
where
    time > '{LAST_DATA_TIME}'
order by
    time desc limit 30
```

- Timescale Chunk Metadata (PostgreSQL)

```
--
-- commented source
--
select
    'chunks' as "m:chunks",
    timeofday()::timestamptz as "t:time",
    hypertable_name as "l:tmetric",
    extract('epoch' from (max(range_end) - min(range_start)))/3600.0 as "f:
chunk_span_total",
    count(*) as "f:chunk_count",
    '{HOST}' as "l:host"
from
    timescaledb_information.chunks
group by
    hypertable_name;
```

- three_consec

```
--
-- This demonstration query will look for three consecutive values
-- for cpu_usage_idle (for cpu-total)
--
SELECT
          'cpu' as "m:cpu",
          cpu as "l:cpu",
          time as "t:time",
          value as "f:three_consec"
FROM
    (
          SELECT
                val(cpu_id) cpu,
                      time,
                      value,
                      lag(value) OVER w value_1,
                      lag(value, 2) OVER w value_2
          FROM
                      cpu_usage_idle
          WHERE
                      time > current_timestamp - INTERVAL '10m'
    AND
                      val(cpu_id) = 'cpu-total'
          WINDOW w AS (ORDER BY time)
    ) last_three
WHERE
          value < 99.5
AND
          value_1 < 99.5
AND
          value_2 < 99.5
ORDER BY
          time desc
```

- Transpara Demo Data (SQL Server)

```
select
          [Name] as "m:tagname",
          [Timestamp] as "t:Timestamp",
          [Value] as "f:Value",
          '{HOST}' as "l:source"
from
          [TimeSeriesData] d,
          [TimeSeriesTags] t
where
          d.[ID] = t.[ID]
and
          [Timestamp] between current_timestamp - 5.0/60.0/24.0 and current
_timestamp
order by
          [Timestamp] desc
```

- Random data (PostgreSQL)

```
SELECT
    'Random' as "m:random",
    '{HOST}' as "l:odbc_host",
    current_database() as "l:db",
    inet_server_addr() as "l:server",
    time as "t:timestamp",
    (random()*30)::int as "f:ivalue",
    random()*80 - 40 as "f:fvalue"
FROM
    generate_series(
          current_timestamp - interval '5m',
          current_timestamp,
          INTERVAL '2 min') AS time;
```