



Migrating from Netezza to Yellowbrick Data is Fast and Painless

A Technical Guide to a Compatible Upgrade from Netezza




Yellowbrick

In partnership with

smartassociates 

Migrating from Netezza to Yellowbrick Data is Fast and Painless

Table of Contents

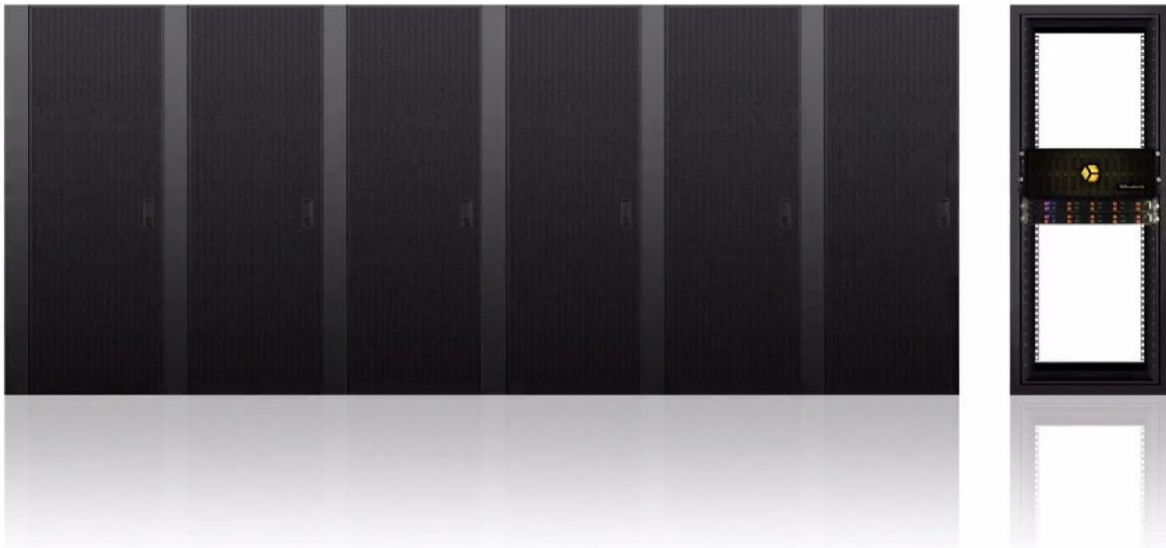
<i>Introduction</i>	3
<i>Same-day installation and deployment</i>	4
<i>A simple database migration</i>	4
Determining the current Netezza DDL	4
Migrating table definitions.....	5
Migrating view definitions	5
Migrating stored procedures.....	5
Exporting the data	5
Importing the data.....	6
Migrating user-defined functions (UDFs)	6
Redirecting users and applications	7
<i>Simplify operations with intelligent automation</i>	7
Automate smart workload management	7
Automate system monitoring.....	7
Increase visibility	8
<i>Summary</i>	8
<i>Appendix. Sizing, query history, and network configuration</i>	9
Sizing.....	9
Exporting database query history.....	9
Preparing the network.....	10

Introduction

Yellowbrick has been designed to make migrating from Netezza fast and simple.

From a system perspective, Yellowbrick:

- Is an integrated software and hardware appliance that installs in your data center easily, as a small rack-mount appliance.
- Offers similar capacity points to Netezza, with options from small 32TB appliances to large systems approaching 1PB of raw storage and over 500 CPU cores.
- Has similar enterprise integrations to Netezza for LDAP, SNMP, syslog, as well as a similar security model.



From the database software perspective, the picture is similarly compelling. Yellowbrick:

- Uses the same underlying PostgreSQL dialect as Netezza.
- Supports PL/pgSQL stored procedures.
- Has a library of functions added for Netezza compatibility.
- Has intuitive syntax like Netezza (for example, nzload is ybload and nzsqli is ybsqli).
- Offers a similar set of ecosystem integrations through standard ODBC, JDBC, and ADO drivers.
- Is certified and supported by many of the same enterprise tools you use today, such as Tableau, Microstrategy, and Informatica.

A Yellowbrick Solutions Architect will help you scope and plan your migration as part of a Proof of Concept (POC). Once your POC is complete, your Solutions Architect will then help you migrate your schema, data, application logic, ecosystem tools, and users to the Yellowbrick platform. You'll have a production-ready system installed and running in under a day and will complete your migration within days or weeks. Your post-migration activities are similarly easy:

your team will have very little tuning and configuration, will not have to pay for training and certification for a new system, and can get back to day-to-day business quickly.

What's more, Yellowbrick has improved on Netezza in many areas. It offers superior concurrent and mixed workload support so your apps and databases will run at top speed. Database administration is also much easier: you will not need to groom or vacuum tables, and statistics are automatically gathered and kept up-to-date.

Yellowbrick is also easy to grow. You can scale compute and storage capacity incrementally by adding more blades to the system, up to nearly 1PB of raw storage. This makes sizing your environment more about determining your initial storage needs than estimating what you need five years from now.

This document provides an overview of migration tasks, so you can see how easy it is to migrate to Yellowbrick and serves as a guide to further streamline the process.

Same-day installation and deployment

Yellowbrick can be deployed in under two hours and be production-ready on day one. Its flash-based architecture means no moving parts to fail during transport. A Yellowbrick Solutions Architect will be onsite to install the system for you. Because Yellowbrick delivers up to 1.2PB of user storage in a 6-10U appliance that fits in a standard rack, you can expect to reclaim racks of space in your data center to save money.

A simple database migration

Yellowbrick supports organizing your data into databases and schemas, just like Netezza and every other PostgreSQL-inspired database. This means you do not have to worry about the migration breaking your applications. You can continue to maintain your database organization as you always have, with one database per tenant. If you want to expand to multiple schemas in a single database, you can do so easily.

Yellowbrick includes a query history database that is installed and fully documented. Your queries will be logged and optimized on day one, and you no longer have to endure dumps of in-memory query history or manually configure the NZ_QUERY_HISTORY table as you might have been doing with your Netezza system. Yellowbrick provides a richer, more granular level of detail about all activity within the database and is easy to configure. You can set the retention period for your history with just one click.

Determining the current Netezza DDL

The `nz_ddl_database` command provides all the information about the DDL in your Netezza system, including table, view, synonym, and sequence definitions; user and group role creations; stored procedures; user defined functions; and much more.

To extract the DDL for each database, execute the following command:

```
$ nz_ddl_database <database> > <database>.ddl
```

Your Yellowbrick Solutions Architect will perform an inventory of your DDLs and map them appropriately to Yellowbrick, according to best practices, using automated conversion tools.

Migrating table definitions

Migrating your table definitions is a seamless process. Many Yellowbrick customers move the Netezza DDL (CREATE TABLE) with absolutely no change to their data types. In addition, you will have the opportunity to improve your table distribution with the following options:

- *Replication* for all tables that compress to a Gigabyte or less.
- *Hash-based co-location* of all tables greater than a Gigabyte on a distribution column that links all the tables.
- *Randomly distributed* tables for other use cases.

Yellowbrick also includes the following table features that enable you to further optimize your database:

- Table clustering and sorting that is like Netezza zone maps, but more granular.
- Table partitioning, which is unavailable in Netezza. This feature reduces resource consumption for very large aggregates and supports fact-to-fact joins that are impossible to execute on a Netezza system.
-

Migrating view definitions

Because Yellowbrick is based on PostgreSQL, all your valid SQL operations and database views will run as-is.

Migrating stored procedures

Yellowbrick supports PL/pgSQL stored procedures. Your Yellowbrick Solutions Architect will perform an impact analysis of your query history to determine which, if any, stored procedures you should migrate (that is, which stored procedures your queries are using).

Exporting the data

You export Latin as well as Unicode data from your Netezza system with one of these options:

- Direct streaming from Netezza into the Yellowbrick system (requires giving Yellowbrick permissions).
- Exporting Netezza data to CSV files. You can do this on a Windows or Linux client using the following methods:
 - `nz_backup`
 - `nzsql`

- SQL for remote external tables

Data will export in ASCII-delimited format that can be compressed.

Importing the data

If you exported the data to a CSV file (as opposed to direct streaming), you will import it using the ybload tool. ybload is intuitive like Netezza nzload , but with dramatic improvements. It is designed for parallelization, so you will not need to chunk files manually, and it easily saturates the 10Gb network connection shipped with Yellowbrick. ybload also compresses data before sending it to Yellowbrick and can load multiple files at once. For example, users can run one ybload with a thousand files as input instead of invoking nzload a thousand separate times. These optimizations make loading simple and dramatically accelerate load speeds compared to Netezza.

Yellowbrick can also load data directly from cloud-based object storage that's S3-compatible. Compressed or uncompressed files are supported. If you are an HDFS user storing data in Hadoop, Yellowbrick uses a Spark-based importer that can consume most common Hadoop file formats such as Parquet or Avro.

Migrating user-defined functions (UDFs)

Netezza systems often have an Oracle compatibility toolkit installed that provides a set of UDFs. Typically customers use only a small number of UDFs, and Yellowbrick includes many of these functions out-of-the-box. Your Yellowbrick Solutions Architect will perform an impact analysis to see if there are any functions Yellowbrick does not provide that are being used by queries.

Users, groups, and security configuration

Like Netezza, Yellowbrick supports local or LDAP authentication methods.

Creating database users (or roles) is the primary mechanism for managing local database security. After creating users, you can set up access control by granting or revoking privileges on databases, schemas, tables, columns, and stored procedures. Use the following SQL commands: CREATE ROLE, GRANT, and REVOKE.

Yellowbrick LDAP configurations support directory service providers such as Active Directory (AD) for OpenLDAP. LDAP, LDAPS (LDAP over SSL), and LDAP+TLS configurations. This is an improvement over Netezza LDAP integration, which is limited to just authentication and requires Netezza customers to manually create USERS or GROUPS. This can result in mismatches to the LDAP service.

Yellowbrick provides an alternative to creating non-superusers manually in the database by synchronizing the users and groups stored in an LDAP directory and creates users and roles with the same credentials. You can synchronize specific users and groups by defining search criteria and filters.

Redirecting users and applications

You can connect ODBC, JDBC, and [ADO.NET](#) applications to a Yellowbrick database using standard PostgreSQL drivers you download and install. This means you do not have to purchase new licenses to redirect your existing applications. More importantly, third-party tools (such as BI, ETL, and analytical tools) that are certified for PostgreSQL drivers will automatically be certified for Yellowbrick.

Yellowbrick ships the open-standard PostgreSQL versions of these Windows or Linux drivers with every system. ETL tools leverage yload and yunload for high speed data movement in and out of the Yellowbrick system as described in “Importing Data,” above.

Simplify operations with intelligent automation

As easy as it is to migrate to Yellowbrick, the biggest impact Yellowbrick will have on your organization is *after the migration completes*. Yellowbrick automates most mundane maintenance tasks so you can free your nights and weekends and finally tackle the projects you’ve always wanted to.

Automate smart workload management

Yellowbrick is designed to deliver high performance to mixed workloads that modern enterprises require. It can support thousands of concurrent users running mixtures of real-time transactional atomic inserts, bulk loads of data, ELT workloads, interactive queries, ad hoc queries, and long-running, batch-oriented reports.

While today’s mixed workloads require a high-performance solution, as the number and diversity of workloads increases, it is even more critical that you have the ability to specify, control, and automate the work that matters most. Yellowbrick gives you this with:

- Real-time visibility into workload performance, with alerts, that ensures you know immediately when workloads are struggling. Historical workload reporting also helps you manage workloads effectively across the enterprise.
- Workload management reporting that enables you to see if the performance of different workload groups is meeting business requirements.
- The ability to re-prioritize any workload, on the fly, so you can adapt to unexpected changes in the business environment.

Automating workload management in Yellowbrick is simple and can be done through the Yellowbrick web interface or SQL ALTER commands.

Automate system monitoring

Yellowbrick includes 24x7 predictive remote monitoring technology that enables Yellowbrick Support Engineers to handle day-to-day monitoring activities. Yellowbrick Support Engineers review real-time diagnostic data, provide recommendations, proactively address potential issues, and deliver prompt incident resolution.

You can monitor Yellowbrick yourself, instead of or in addition to the Yellowbrick service. You can do this through automated alerts that easily integrate to the endpoint tool of your choice. This is a welcome relief from manually configuring each nzevent that you want to monitor.

Increase visibility

System views give you comprehensive, real-time visibility into the Yellowbrick system. Monitoring is performed continuously without affecting system performance and is thoroughly documented so you can find what you need, when you need to.

Summary

Migrating to Yellowbrick from Netezza is fast and painless. In days or weeks you can have a running system that is intuitive like your Netezza system and largely compatible. Better still, your Yellowbrick system will be faster, easier to manage, and incrementally scalable.

To learn more about how Yellowbrick can get you back to business quickly, with a system that is ideal for meeting all your future demands, visit <http://www.smart-associates.biz/> or contact us at enquiries@smart-associates.biz.



Smart Associates (Aotearoa) Ltd	Smart Associates Ltd	Smart Associates ApS
203/11 Vinegar Lane Grey Lynne Auckland 1021 New Zealand	Valley View, The Old Quarry Haslemere Surrey GU27 3SS United Kingdom	c/o NJORD Advokatpartnerselskab Pilestræde 58 1212 København K Denmark
T: +64 (9) 415-8120	T: +44 (208) 133-6008	T: +45 36 98 71 11

Yellowbrick is a registered trademark of Yellowbrick in the United States, other countries, or both.

Appendix. Sizing, query history, and network configuration

Sizing

You can accurately size your system by running the `nz_compressedTableRatio` command on each database, as follows:

```
$ nz_compressedTableRatio <database> > <database>.sizing
```

Exporting database query history

Netezza collects query history in a table called `NZ_QUERY_HISTORY`. Follow these steps to export your query history into a file that a Yellowbrick Solution Architect can use to help optimize your environment.

1. Determine if you are collecting query history with the following commands:

```
$ cd /nz/support/bin
```

```
$ ./nz_find_object nz_query_history
```

2. If you have been collecting history, connect to the database and run the following SQL:

```
select max(QH_TSUBMIT) from NZ_QUERY_HISTORY;
```

```
-----
```

```
MAX
```

```
2013-10-28 11:04:58
```

```
(1 row)
```

3. Check the date to make sure you are still collecting. The max date should be today.
4. Export the query data into a file with the following command:

```
create external table foo SAMEAS NZ_QUERY_HISTORY
```

```
USING (DATAOBJECT('/tmp/nz_qhist.out') escapeChar '\ ' delim '|'); cd
```

```
insert into foo select * from NZ_QUERY_HISTORY; drop table
```

```
foo;
```

5. Compress the contents using gzip:

```
gzip /tmp/nz_qhist.out
```

6. Check the size of the file to be sure the data exported to the

```
file: ls -l /tmp/nz_qhist.out
```

- 7.

Preparing the network

Configure network firewalls, routers, and so forth, for the following port and connections:

- Yellowbrick database connection. Used by the Yellowbrick remote monitoring feature. This feature sends system information to Yellowbrick Support Engineers that monitor the system. Note that only system information is sent (not data), and this feature can be disabled.
- Yellowbrick database port. The database uses TCP port 5432 (YBPORT).
- Bulk Load Manager port (ybload).
 - Bulk load and monitoring: 11111
 - Data transfer ports: 31000-32000. ybload uses 30 ports in this range to load data across the cluster.
- 10-gigabit connections are preferred for ETL/backup clients. Yellowbrick provides multiple ports for redundancy.
- Yellowbrick uses SSL encryption for all data-in-motion encryption, including the connection to the Yellowbrick Data database (ybsql) and odbc/jdbc connections.