

# New Features Guide

## HP Vertica Analytic Database

Software Version: 7.0.x



Document Release Date: 2/24/2014



## 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.

# Contents

Contents .....	3
New Features and Changes in HP Vertica 7.0.x .....	7
New Features and Changes in HP Vertica 7.0 SP1 (7.0.1) .....	7
Flex Table Additions and Updates .....	7
Adding Columns to Pre-Join Projections .....	8
Improved Diagnostics Collection .....	8
Time-based Data Retention .....	9
Plug-in for Informatica USE SINGLE TARGET Attribute .....	10
New HCatalog Connection Parameters .....	10
New Java UDX Resource Management Function .....	10
Management Console .....	10
Using Management Console to Apply Your Flex Zone License .....	10
Monitoring Profiling Progress .....	11
New Features and Changes in HP Vertica 7.0.0 .....	11
Flex Zone .....	11
Flex Tables Load and Query .....	11
Spread Integration into HP Vertica Server .....	13
Large Cluster Arrangement .....	13
Fault Groups .....	14
Comprehensive Pre-install System Verification .....	14
JDBC Key/Value API .....	14
Client Library Changes .....	14
JDBC 4.0 Support .....	15
Client Driver Connection Failover Support .....	15
Client Driver Connection Load Balancing .....	15
Client Driver Connection Kerberos Authentication .....	16
Streaming Batch Inserts .....	16
Database Designer Improvements .....	16
Database Designer Terminology Changes .....	16

Using Management Console to Create an Optimal Design .....	17
Running Database Designer Programmatically .....	17
Running Database Designer Concurrently .....	18
Identifying and Grouping Similar Queries in Database Designer .....	18
Avoiding Data Skew in Database Designer .....	19
Analyzing Correlated Columns .....	19
Security .....	19
LDAP Support for StartTLS and LDAPS .....	20
Extended Support for Kerberos Authentication .....	20
Multiplexed Networking .....	21
Management Console Enhancements in 7.0 .....	21
Using Management Console to Create an Optimal Design .....	21
Viewing EXPLAIN Plans and Profiling Data in Management Console .....	22
Management Console Themes .....	22
Performance .....	24
Improved Performance When a Node Fails .....	24
Approximate Count Distinct Aggregate Functions .....	24
Optimizing Aggregate Queries with Partially Sorted GROUPBY .....	24
Improved MERGE Performance .....	25
ADO.NET Performance Improvements .....	25
SQL Enhancements and Changes .....	26
ALTER TABLE UDSFs as Default Expression .....	26
New Statements and Functions .....	26
Approximate Count Distinct Aggregate Functions .....	28
New and Modified System Tables .....	28
New Configuration Parameters .....	29
COPY Load Changes .....	30
New EXPLAIN Plan Output .....	32
SDK Enhancements .....	33
New Java SDK for UDx Development .....	33
UDChunker for Parallel Parsing Capabilities .....	33

Metadata for UDX Libraries .....	34
VBR Backup Changes .....	34
Add Nodes to the Cluster Before Object Restore .....	34
Simplified Configuration File Mappings .....	34
Third-Party Tools Integration Updates .....	36
Connectivity Pack Enhancements .....	36
New Informatica Plug-in .....	36
HCatalog Connector for Querying Data in Hive Data Warehouse .....	37
HDFS Connector .....	37
Documentation Updates .....	37
New Look-and-Feel and Better Search .....	37
New Guides .....	37
SDK and Supported Platforms Online Documentation .....	38
Deprecated Functionality in This Release .....	39
Retired Functionality .....	41
We appreciate your feedback! .....	45



# New Features and Changes in HP Vertica 7.0.x

This guide describes new features and changes that have been added across the HP Vertica 7.0.x release line.

For a list of known and resolved issues, see the HP Vertica 7.0 Release Notes, available at <http://www.vertica.com/documentation>

## New Features and Changes in HP Vertica 7.0 SP1 (7.0.1)

Read the topics in this section for information about new and changed functionality in HP Vertica 7.0 SP1 (7.0.1).

### Flex Table Additions and Updates

This release includes the following changes to HP Vertica flex tables:

- New ArcSight log data parser, `fcefparser`
- New `audit_flex()` function
- The `maptostring()` function returns canonical JSON by default

### *New ArcSight CEF Parser*

The `COPY` statement supports a third flex table ArcSight CEF (Common Event Format) parser :

- `fcefparser`

Use this parser to load flex tables with logging data from the HP ArcSight Enterprise Security Management (ESM) and ArcSight Logger log management software.

### *New Function to Audit Flex Table Data Usage*

Use the new `audit_flex()` function to audit the data storage used in ROS containers for the `__raw__` column of flexible tables. Use this function to audit the flex data usage in the entire database, a schema, projection, or one flex table.

See the `AUDIT_FLEX` function in the SQL Reference Manual.

## ***New Parameter for maptostring() Function***

The `maptostring()` function has a new, optional boolean parameter, `canonical_json`. The function returns canonical JSON by default (`canonical_json=true`). The new default changes the function's behavior from previous releases. To maintain the previous behavior, use `maptostring()` with `canonical_json=false`.

For examples of using `maptostring()` with its new default behavior, see [mapToString](#) in the Flex Tables Guide.

## **Adding Columns to Pre-Join Projections**

In versions of HP Vertica prior to 7.0.0, when you added a column to a table, HP Vertica also added the column to the superprojection for that table.

HP Vertica 7.0.1 provides a new syntax that inserts any new table column to the table, the superprojection, and to all pre-join projections where the table is specified.

The new syntax is:

```
ALTER TABLE t ADD COLUMN c_new <column definition> CASCADE;
```

This feature has the following limitations:

- If you specify a default value for the new column that is not a constant, HP Vertica does not add the new column to the pre-join projections. Instead, it does add the column to the table and its superprojection.
- If you do not specify a default value for the new column, HP Vertica assigns a NULL default. Since NULL is a constant, HP Vertica adds the column to all pre-join projections.
  - When HP Vertica adds a column to a table, it acquires an O lock on the table until the operation completes. If you use the CASCADE keyword, HP Vertica also acquires O locks on all the anchor tables of any pre-join projections associated with that table.

For more information, see [ALTER TABLE](#).

## **Improved Diagnostics Collection**

HP Vertica 7.0.1 (7.0 SP1) introduces several enhancements to diagnostics collection using the `scrutinize` command.

## ***Authenticated upload***

The new `--auth-upload` argument collects your company name from your HP Vertica Enterprise Edition license. When you specify the authenticated upload argument:



- You do not need a user name and password to send your diagnostics file to HP Vertica, which means that more people in your organization get the help they need
- HP Vertica Technical Support uses your company name to authenticate your identity

## ***Collect tasks that don't require a database connection***

The new `--vsq1-off` argument lets you exclude tasks from diagnostics collection that require a database connection. For example, you might want to use `--vsq1-off` if HP Vertica is up but running slowly. This argument can also be helpful for dealing with problems during an upgrade. Use `--vsq1-off` if you haven't created a database but need help troubleshooting other cluster issues.

## ***Exclude a category of tasks from collection***

The updated `--exclude-tasks` argument lets you omit the collection of tasks by type. For example, you can exclude Data Collector tables, catalog metadata, and other categories that scrutinize collects by default. Also use `--exclude-tasks` when running the full diagnostic collection might be too time consuming or could prevent you from running queries.

For full syntax and parameters, see [Collecting Diagnostics \(scrutinize Command\)](#) in the *Administrator's Guide*.

## **Time-based Data Retention**

In previous releases, you used the `SET_DATA_COLLECTOR_POLICY()` function to specify how much data to retain on disk. When data exceeded the defined space allocation, HP Vertica purged older data to make room for new.

In HP Vertica 7.0.1 (7.0 SP1), you can configure both size and time restraints at the same time using `SET_DATA_COLLECTOR_POLICY()`.

HP Vertica 7.0.1 also introduces a new function, `SET_DATA_COLLECTOR_TIME_POLICY()`, to configure just the time interval for an individual Data Collector (DC) table or for all DC tables.

Retaining DC logs for a specified time is useful if you want to:

- Analyze the last  $n$  days of cluster activity
- Use joined DC tables to understand what queries were running during a particular interval

See the following topics in the *SQL Reference Manual*:

- [SET\\_DATA\\_COLLECTOR\\_POLICY](#)
- [SET\\_DATA\\_COLLECTOR\\_TIME\\_POLICY](#)

## Plug-in for Informatica USE SINGLE TARGET Attribute

HP Vertica has added an additional VERTICA\_WRITER plug-in attribute, **USE SINGLE TARGET**. If you have checked the option **USE SINGLE TARGET**, data is sent to the one IP address you entered when you configured your workflow connections.

You might set this option, for example, if you want to specify one ingestion node rather than use the plug-in's default ROUNDROBIN scheme for ingesting data.

For more information, refer to the section, [Accessing VERTICA\\_WRITER Plug-in Attributes](#), in the Informatica Plug-in Guide.

## New HCatalog Connection Parameters

This release introduces three new parameters for managing how the HCatalog Connector handles connectivity issues. These parameters let you tell the HCatalog connector how long it should wait for a connection to the WebHCat server, and enforce a minimum data transfer speed from the WebHCat server.

These parameters can be set globally via three new configuration parameters. See [HCatalog Connector Parameters](#) in the Administrator's Guide for details. They can also be set on a per-schema basis using new arguments to the [CREATE HCATALOG SCHEMA](#) statement.

For tips on using these parameters, see [Preventing Excessive Query Delays](#) in the Hadoop Integration Guide.

## New Java UDx Resource Management Function

This release includes a new HP Vertica meta-function named [RELEASE\\_ALL\\_JVM\\_MEMORY](#). It closes all Java Virtual Machines (JVMs) used to run Java UDxs. Only a superuser can use this function.

See [Java UDx Resource Management](#) in the Programmer's Guide for more information.

## Management Console

HP Vertica Analytics Platform introduced new functionality to the Management Console user experience.

- [Applying Your Flex Zone License in Management Console](#)
- [Monitoring Profiling Progress](#)

## Using Management Console to Apply Your Flex Zone License

In this release, Management Console 7.0.1 allows you to apply a Flex Zone license in addition to HP Vertica Enterprise Edition licenses. See [Understanding HP Vertica Licenses](#) and [FlexTable](#)

[License Installations.](#)

## ***Monitoring Profiling Progress***

In HP Vertica Management Console 7.0.1 includes the ability to monitor profiling progress. When loading profile data for a query, Management Console can provide updates about the query's progress and resources used. See [Monitoring Profiling Progress](#).

## **New Features and Changes in HP Vertica 7.0.0**

Read the topics in this section for information about new and changed functionality in HP Vertica version 7.0.0.

### **Flex Zone**

With version 7.0, HP introduces a new offering called HP Vertica Flex Zone, based on the flex tables technology available in 7.0. A Flex Zone license entitles you to more than the one terabyte of unstructured data included with your HP Vertica license, giving you even more flexibility with your data.

If your primary goal is to work with [flex tables](#), you can purchase a Flex Zone license alone. When you purchase Flex Zone, you will receive a complimentary Enterprise Edition license, entitling you to one terabyte of columnar data, with no node limit.

To obtain an Enterprise Edition or Flex Zone license key, contact HP Vertica at:  
<http://www.vertica.com/about/contact-us/>

### **Flex Tables Load and Query**

This release includes a new kind of table, flex (or flexible) tables, to support loading and querying unstructured and semi-structured data. Use flex tables and their associated functionality to store and manipulate unstructured data and complete these tasks:

- Create flex tables (with or without defined columns)
- Load data into flex tables
- Use standard SQL queries to extract data from flex tables
- Alter flex tables to add regular (materialized) columns for data of interest and to create a hybrid table, containing both unstructured and structured data

Using hybrid flex tables results in similar performance to regular, columnar HP Vertica tables.

### ***Table Statements***

These are the new statements that support flex tables:

- [CREATE FLEX TABLE](#)
- [CREATE FLEX EXTERNAL TABLE AS COPY](#)

## ***New Flexible Table Parsers***

Use the [COPY](#) statement to load data directly into the new table with one of two new parsers:

- `fjsonparser`
- `fdelimitedparser`

## ***New System Table for Flex Table Function***

The new flex data function to materialize columns ([MATERIALIZED\\_FLEXTABLE\\_COLUMNS\(\)](#)) has an associated system table for the results of running the function. See [MATERIALIZED\\_FLEXTABLE\\_COLUMNS\\_RESULTS](#) in the SQL Reference Manual.

## ***Flex Table Map Functions***

After loading data into a flex table, use the new map functions to display and extract information about unstructured data. For instance, you can view a list of what keys (virtual column names) exist in the raw data within a flex table, or find out if a particular key, or value exists. These are the map functions to support flex tables:

- [mapAggregate](#)
- [mapLookup](#)
- [mapContainsKey](#)
- [mapContainsValue](#)
- [mapItems](#)
- [mapKeys](#)
- [mapKeysInfo](#)
- [mapLookup](#)
- [mapSize](#)
- [mapToString](#)
- [mapValues](#)

- [mapVersion](#)
- [emptyMap](#)

## ***Flex Table Data Functions***

Use the new flex table helper functions to compute data keys and values, and perform other tasks.

While these functions are included as part of the HP Vertica database, they are for use exclusively with flex tables:

- [BUILD\\_FLEXTABLE\\_VIEW](#)
- [COMPUTE\\_FLEXTABLE\\_KEYS](#)
- [COMPUTE\\_FLEXTABLE\\_KEYS\\_AND\\_BUILD\\_VIEW](#)
- [MATERIALIZE\\_FLEXTABLE\\_COLUMNS](#)
- [RESTORE\\_FLEXTABLE\\_DEFAULT\\_KEYS\\_TABLE\\_AND\\_VIEW](#)

## **Spread Integration into HP Vertica Server**

Vertica Analytic Database uses spread as its control messaging network. In Vertica Analytic Database 7.0, spread is managed at database start/stop time, rather than during the Vertica Analytic Database installation process. This means that the spread process does not run when the database is not running.

The spread system service and system user are no longer used.

## **Large Cluster Arrangement**

In Vertica Analytic Database 7.0, a large cluster arrangement delegates control message responsibilities to a subset of Vertica Analytic Database nodes (called control nodes) to improve control message performance. The feature is necessary (and enabled by default) for clusters of 120 nodes or more; however, some environments, such as cloud deployments that might have higher network latency, could benefit from a smaller number of control nodes.

You can set the number of control nodes on the cluster using one of the following two new methods:

- For new installations of Vertica Analytic Database, use the `--large-cluster <integer>` argument. See [Installing HP Vertica with the `install\_vertica` Script](#) in the Installation Guide.
- To expand an existing database cluster, use the new cluster-management functions. See [Defining and Realigning Control Nodes on an Existing Cluster](#) in the Administrator's Guide.

This feature is integrated with elastic cluster and the new [fault groups](#) to provide cluster flexibility and reliability that you expect from Vertica Analytic Database. For more information, see [Large Cluster](#) in the Administrator's Guide.

## Fault Groups

Fault groups provide a way for you to configure Vertica Analytic Database for your physical cluster layout to reduce the impact of correlated failures inherent in your environment. For example, by describing your rack layout, Vertica Analytic Database could tolerate a rack failure.

Vertica Analytic Database 7.0 supports complex, hierarchical fault groups of different shapes and sizes by providing a fault group helper utility, SQL statements, system tables, and other monitoring tools.

Fault groups are integrated with elastic and large cluster layouts to provide the cluster flexibility and reliability that you expect from Vertica Analytic Database.

## See Also

- [Large Cluster Arrangement](#) in this guide
- [High Availability With Fault Groups](#) in the Concepts Guide
- [Fault Groups](#) in the Administrator's Guide

## Comprehensive Pre-install System Verification

With version 7.0, HP introduces an updated HP Vertica installer that verifies your hardware and OS settings. If the installer encounters any configuration mismatches that it cannot automatically correct, it provides a brief message about the issue and links directly to detailed documentation on how to correct the issue.

For more details, see [Operating System Configuration Task Overview](#) in the Installation Guide.

## JDBC Key/Value API

With version 7.0, HP introduces a new JDBC Key/Value API. The JDBC Key/Value API allows you to quickly query data when a single or only a few rows are being returned and the data exists on a single node. This feature is ideal for high-volume "short" requests that return a small number of results.

For more details, see [About the JDBC Key/Value API](#).

## Client Library Changes

Several enhancements have been made to the client libraries in this version of the HP Vertica Analytics Platform.

## ***JDBC 4.0 Support***

The Vertica Analytic Database Version 7.0.x JDBC driver now supports the JDBC 4.0 standard. This support includes:

- JNDI service registration , so that the JVM locates and loads the JDBC driver automatically when your client application attempts to create a connection to Vertica Analytic Database.
- Improved exception classes that let your client application better respond to errors.
- Wrapper and better DatabaseMetaData support that lets client applications learn about the Vertica Analytic Database JDBC driver's features.
- Better connection pooling support.

See [New Features in the Version JDBC Driver](#) for more information about the driver's JDBC 4.0 support.

## ***Client Driver Connection Failover Support***

The Vertica Analytic Database Version 7.0.x client driver libraries now support connection failover. When creating a connection to the Vertica Analytic Database database, you can supply a list of additional hosts that the client driver should try to contact if the primary host specified in the connection cannot be reached.

For more information see the following section sin the Programmer's Guide:

- [ODBC Connection Failover](#)
- [JDBC Connection Failover](#)
- [ADO.NET Connection Failover](#)
- [VSQL Command Line Options](#) (see the -B option)

## ***Client Driver Connection Load Balancing***

Every client connected to a host consumes a small amount of the host's resources. If one node has many more client connections than other hosts in the Vertica Analytic Database cluster, the resources consumed by the clients can being to affect its performance as a database node.

To help solve this issue, all Vertica Analytic Database client drivers now support connection load balancing which spreads the overhead of maintaining client connections across all hosts in the cluster.

For more information, see the following sections in the Programmer's Guide:

- [Enabling Native Connection Load Balancing in JDBC](#)
- [Enabling Native Connection Load Balancing in ODBC](#)
- [Enabling Native Connection Load Balancing in ADO.NET](#)
- [VSQL Command Line Options](#) (see the -C option)

## ***Client Driver Connection Kerberos Authentication***

All Vertica Analytic Database 7.0 client drivers support new connection strings for Kerberos version 5 authentication through the GSSAPI (Generic Security Services Application Program Interface). See [Extended Support for Kerberos Authentication](#) in this guide.

## ***Streaming Batch Inserts***

In version 7.0, the HP Vertica JDBC driver introduces streaming batch inserts, a new loading model built into the standard batch insert API. Streaming batch inserts allow you to add a row to the database using `addBatch()`. Streaming batch inserts improve load performance by reducing memory usage and providing more opportunities for parallel processing.

## **Database Designer Improvements**

HP Vertica 7.0 offers the following improvements to the design, performance, and usability of Database Designer, speeding up design and deployment operations with minimal impact to design quality.

- [Database Designer Terminology Changes](#)
- [Using Management Console to Create an Optimal Design](#)
- [Running Database Designer Programmatically](#)
- [Running Database Designer Concurrently](#)
- [Identifying and Grouping Similar Queries in Database Designer](#)
- [Avoiding Data Skew in Database Designer](#)
- [Analyzing Correlated Columns](#)

## ***Database Designer Terminology Changes***

In HP Vertica 7.0, there are two changes in terminology in the way the documentation describes the features of Database Designer.

There are two design types:



- Comprehensive
- Incremental (formerly query-specific)

For more information, see [Design Types](#).

There are three optimization objectives, formerly known as design priorities:

- Load
- Query
- Balanced

For more information, see [Optimization Objectives](#).

## ***Using Management Console to Create an Optimal Design***

HP Vertica 7.0. allows you run the Database Designer from Management Console to create an optimal design for your database.

Management Console provides a wizard to walk you through the steps to set the Database Designer parameters. The interface provides all the capability of the Administration Tools, and some additional capabilities such as:

- Exporting a database
- Viewing the design script
- Setting the design K-safety
- Allowing Database Designer to create unsegmented projections

For more information, see the following topics in the Administrator's Guide:

- [Viewing EXPLAIN output using Management Console](#)
- [Viewing profiling data using Management Console](#)

## ***Running Database Designer Programmatically***

In HP Vertica 7.0, if you have been granted the DBDUSER role and have enabled the role, you can access Database Designer functionality programmatically. In previous releases, Database Designer was available only via the Administration Tools. Using the DESIGNER\_\* command-line functions, you can perform the following Database Designer tasks:

- Create a comprehensive or incremental design
- Add tables and queries to the design
- Set the optimization objective to prioritize for query performance or storage footprint
- Assign a weight to each query
- Assign the **K-safety** value to a design
- Analyze statistics on the design tables
- Create the script that contains the **DDL** statements that create the design projections
- Deploy the database design
- Specify that all projections in the design be segmented
- Populate the design
- Cancel a running design
- Wait for a running design to complete
- Deploy a design automatically
- Drop database objects from one or more completed or terminated designs
- For information about this new capability and a tutorial on how to use these functions, see [About Running Database Designer Programmatically](#) in the Programmer's Guide.
- For detailed information about each function, see [Database Designer Functions](#) in the SQL Reference Manual.

## ***Running Database Designer Concurrently***

In HP Vertica 7.0, users assigned the DBADMIN or DBDUSER role can run Database Designer concurrently. If two or more users attempt to access the same tables at the same time, Database Designer may not be able to complete. If you plan to create multiple designs that access the same table, your best bet is to execute them sequentially, not concurrently.

## ***Identifying and Grouping Similar Queries in Database Designer***

In HP Vertica 7.0, before creating a design, Database Designer identifies design queries that affect the design that Database Designer creates in the same way and assigns them a signature. For queries with the same signature, Database Designer weights the queries depending on how many queries are in that cluster and considers one of the weighted queries when creating the design.

This feature allows Database Designer to accept a large number of design queries. but by grouping similar queries together, Database Designer does not have to consider all the queries while creating the design, saving processing time, and producing a better design.

## ***Avoiding Data Skew in Database Designer***

In HP Vertica 7.0, Database Designer uses an improved algorithm to segment projections, creating a uniform distribution of data across the cluster and minimizing data skew.

## ***Analyzing Correlated Columns***

In HP Vertica 7.0, when creating a design programmatically, Database Designer can analyze columns that are strongly correlated. (By default, Database Designer does not analyze correlations.)

For example, state name and country name columns are strongly correlated because the city name usually, but perhaps not always, identifies the state name. The city of Conshohocken is uniquely associated with Pennsylvania, whereas the city of Boston exists in Georgia, Indiana, Kentucky, New York, Virginia, and Massachusetts. In this case, city name is strongly correlated with state name.

When creating a design, Database Designer can:

- Analyze correlated columns.
- Order the correlated columns in a way that maximizes the compression ratio.
- Place them early in the projection sort order.
- Choose Run-Length Encoding (RLE) for the columns that satisfy the recommended RLE criteria.

The new function [ANALYZE\\_CORRELATIONS](#) allows you to find and record correlated columns in a schema. ANALYZE\_CORRELATIONS also collects statistics.

For Database Designer to take advantage of these correlations, run Database Designer programmatically. Run [DESIGNER\\_SET\\_ANALYZE\\_CORRELATIONS\\_MODE](#) to specify that Database Designer should consider existing column correlations. Make sure to tell Database Designer not to analyze statistics to prevent Database Designer from override the existing statistics.

Column correlation analysis only needs to happen once.

HP Vertica recommends that you analyze correlated columns only if the table contains at least 4000 rows.

## **Security**

HP Vertica 7.0 includes the following security enhancements:

- [LDAP Support for StartTLS and LDAPS](#)
- [Extended Support for Kerberos Authentication](#)

## ***LDAP Support for StartTLS and LDAPS***

HP Vertica 7.0 supports

- StartTLS on the LDAP port (389) or on a specified port.
- LDAPS (LDAP over SSL) on the LDAPS port (636) or on a specified port.

**Important:** If you have HP Vertica enabled for LDAP over SSL/TLS, before upgrading your database, read the information in [Configuring LDAP Over SSL/TLS When Upgrading HP Vertica](#) in the Installation Guide.

## ***Extended Support for Kerberos Authentication***

In previous releases, HP Vertica supported Kerberos authentication for vsql clients on Linux platforms only. In version 7.0, HP Vertica supports Kerberos version 5 through the GSSAPI (Generic Security Services Application Program Interface) on all client platforms. The GSSAPI standard provides better compatibility with non-MIT Kerberos implementations, such as Microsoft Active Directory.

**Deprecated:** In HP Vertica 7.0, the `krb5` authentication method has been deprecated in favor of the `gss` method.

For more information, see the following topics in the Administrator's Guide:

- [Implementing Client Authentication](#)
- [Security Parameters](#) (ClientAuthentication entry)

## ***Kerberos Driver Connection Parameters***

HP Vertica provides the following new driver connection parameters for client authentication with the Vertica Analytic Database server:

- `KerberosServiceName`—Provides the service name portion of the HP Vertica Kerberos principal; for example: `vertica/host@EXAMPLE.COM`
- `KerberosHostname`—Provides the instance or host name portion of the HP Vertica Kerberos principal; for example: `vertica/host@EXAMPLE.COM`

HP Vertica also provides the following new client-specific connection parameters:

- For JDBC drivers, `JAASConfigName` lets you provide the name of the JAAS configuration that contains the JAAS `Krb5LoginModule` and its settings.
- For ADO.NET drivers, `IntegratedSecurity` lets you specify a Boolean value that, if set to true, uses the calling user's Windows credentials for authentication, instead of user/password in the connection string.

See the following topics in the Programmer's Guide:

- [ODBC DSN Parameters](#)
- [JDBC Connection Properties](#)
- [ADO.NET Connection Properties](#)

## ***Kerberos vsql Command Line Options***

HP Vertica also provides new [command line options](#) for **vsq**l. These options are equivalent to the drivers' `KerberosServiceName` and `KerberosHostName` connection strings:

- `-k KRB SERVICE` provides the service name portion of the Kerberos principal
- `-K KRB HOST` provides the host name portion of the Kerberos principal

## **Multiplexed Networking**

The HP Vertica 7.0 data networking layer has been restructured to be more suitable to large clusters and complex queries by reducing the number of necessary node-to-node connections established.

## **Management Console Enhancements in 7.0**

HP Vertica Analytics Platform 7.0. introduced significant improvements to the Management Console user experience.

- [Using Management Console to Create an Optimal Design](#)
- [Viewing EXPLAIN Plans and Profiling Data in Management Console](#)
- [Management Console Themes](#)

## ***Using Management Console to Create an Optimal Design***

HP Vertica 7.0. allows you run the Database Designer from Management Console to create an optimal design for your database.

Management Console provides a wizard to walk you through the steps to set the Database Designer parameters. The interface provides all the capability of the Administration Tools, and some additional capabilities such as:

- Exporting a database
- Viewing the design script
- Setting the design K-safety
- Allowing Database Designer to create unsegmented projections

For more information, see the following topics in the Administrator's Guide:

- [Viewing EXPLAIN output using Management Console](#)
- [Viewing profiling data using Management Console](#)

## ***Viewing EXPLAIN Plans and Profiling Data in Management Console***

In HP Vertica 7.0., you can now view EXPLAIN plans and profiling data using Management Console. The interface gives visual information that quickly allows you to see how much time was spent in each phase of query processing. In addition, you can see detailed information about

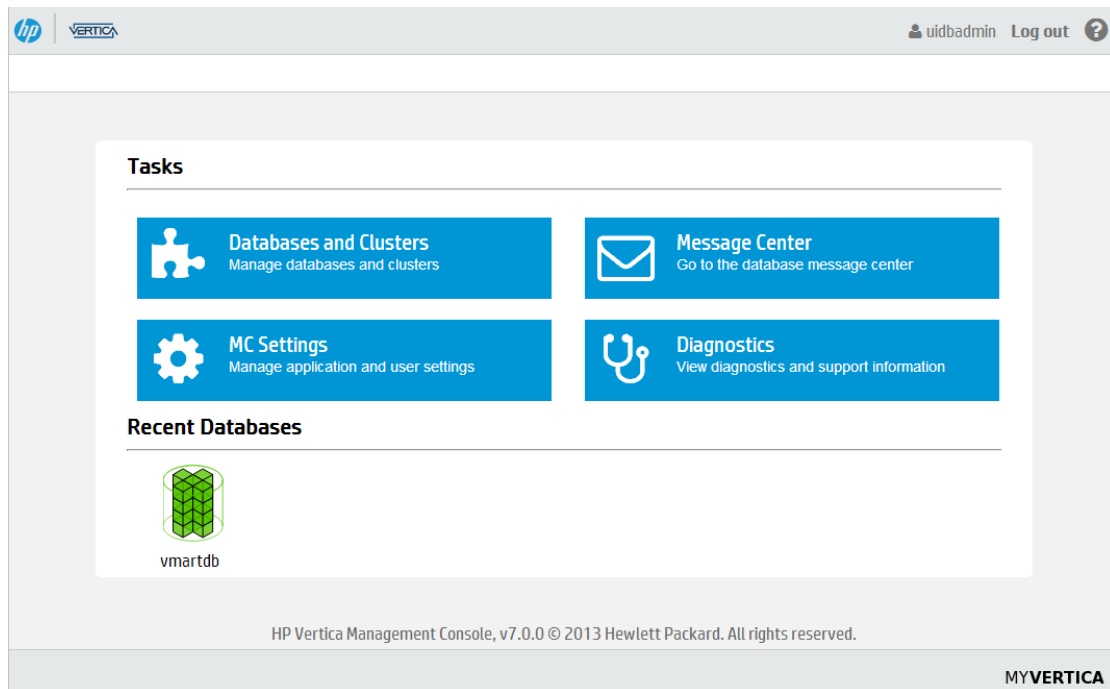
- Projection metadata
- Execution events
- Optimizer events

For more information, see the following topics in the Administrator's Guide:

- [Viewing EXPLAIN output using Management Console](#)
- [Viewing Profile Data Using Management Console](#)

## ***Management Console Themes***

With HP Vertica Analytics Platform 7.0.0, you can now choose different "themes" for the HP Vertica Management Console. Management Console themes change the look and feel of the interface. In addition to Management Console's classic look, now available is the new HP Theme, shown below. Administrators can apply the HP Theme or the HP Vertica Classic themes through MC Settings.



## Performance

In this release, HP Vertica introduces numerous infrastructure improvements to increase performance across the entire platform.

### ***Improved Performance When a Node Fails***

HP Vertica provides performance optimization when cluster nodes fail by distributing the work of the down nodes uniformly among available nodes throughout the cluster.

### ***Approximate Count Distinct Aggregate Functions***

HP Vertica 7.0 provides three new aggregate functions. These functions calculate an approximate count of distinct values in a column or expression.

- [APPROXIMATE\\_COUNT\\_DISTINCT](#)
- [APPROXIMATE\\_COUNT\\_DISTINCT\\_SYNOPSIS](#)
- [APPROXIMATE\\_COUNT\\_DISTINCT\\_OF\\_SYNOPSIS](#)

The expected value that `APPROXIMATE_COUNT_DISTINCT()` returns is equal to `COUNT(DISTINCT)`, with an error that is lognormally distributed with standard deviation *s*. You can control the standard deviation directly by setting the *error\_tolerance*.

Use `APPROXIMATE_COUNT_DISTINCT()` as a replacement for `COUNT (DISTINCT)` when an approximate value is sufficient or the values need to be combined.

`APPROXIMATE_COUNT_DISTINCT_SYNOPSIS()` and `APPROXIMATE_COUNT_DISTINCT_OF_SYNOPSIS()` can roll up the data. This capability enables you to pre-compute distinct counts and later combine them in different ways, just as you can do with a non-distinct `COUNT()` or `SUM()` aggregate.

**Note:** The `APPROXIMATE_COUNT_DISTINCT*` functions cannot appear in the same query block as `DISTINCT` aggregates.

For detailed information about when to use these functions, see [Optimizing COUNT \(DISTINCT\) by Calculating Approximate Counts](#) in the Programmer's Guide.

For more information about `COUNT(DISTINCT)`, see [COUNT \[Aggregate\]](#) in the SQL Reference Manual.

### ***Optimizing Aggregate Queries with Partially Sorted GROUPBY***

Using a combination of the `GROUPBY PIPELINED` and `GROUPBY HASH` algorithms, HP Vertica provides an optimization known as *partially sorted GROUPBY*, which is enabled by default. This optimization allows the HP Vertica optimizer to improve the performance of most aggregate queries



on large data sets by leveraging the sort order of one or more columns in the `GROUP BY` clause. The queries may or may not include `DISTINCT` aggregate columns, such as `COUNT(DISTINCT . . .)` or `SUM(DISTINCT . . .)`. The partially sorted `GROUPBY` optimization uses smaller hash tables, which reduces the possibility of hash tables spilling to temporary files on disk while processing the query.

You do not need to edit your queries to take advantage of this optimization. By default, the HP Vertica optimizer chooses the partially sorted `GROUP BY` optimization automatically for queries that can benefit from it. To see if a query will execute using this optimization, view the query `EXPLAIN` plan.

The partially sorted `GROUPBY` optimization does not speed up queries in which the input data sets are small enough to fit entirely in main memory.

For more information, see

- [Partially Sorted GROUPBY](#) in the Programmer's Guide
- [Partially Sorted GROUPBY EXPLAIN Plan Example](#) in the Administrator's Guide

## ***Improved MERGE Performance***

HP Vertica 7.0 now prepares an optimized query plan, by default, to improve `MERGE` performance when the `MERGE` statement and its tables meet the following criteria:

- The target table join column has a `PRIMARY KEY` or `UNIQUE` constraint
- `UPDATE` and `INSERT` clauses include every column in the target table
- `UPDATE` and `INSERT` clause column attributes are the same

When the above conditions are not met HP Vertica prepares a non-optimized plan for `MERGE`, and the statement runs with the same performance as in HP Vertica 6.1 and prior.

**Note:** When HP Vertica prepares an optimized query plan for a merge operation, it enforces strict requirements for unique and primary key constraints in the `MERGE` statement's join columns. If you haven't enforced constraints, the `MERGE` statement could fail. For details, see [Optimized Versus Non-Optimized MERGE Plan](#) in the Administrator's Guide.

`MERGE` statement syntax has not changed in HP Vertica 7.0.

## ***ADO.NET Performance Improvements***

The HP Vertica 7.0 ADO.NET driver, provided as part of the Microsoft Connectivity Pack, has been updated with many performance improvements to increase performance and lower latency when reading data from the ADO.NET `VerticaDataReader` and data loading in general.

## SQL Enhancements and Changes

This section contains the SQL enhancements made in Vertica Analytic Database 7.0.

### ***ALTER TABLE UDSFs as Default Expression***

In previous versions, using the ALTER TABLE statement to add a column with a default value expression did not permit the use of user-defined scalar function (UDSFs). HP Vertica 7.0 now supports a default column value expression derived from a UDSF.

For more information about UDFs, see [Developing and Using User Defined Functions](#) in the Programmer's Guide.

For an example of adding a table column with a derived default expression, see [Adding a Column with a Default Value](#) in the Administrator's Guide.

### ***New Statements and Functions***

HP Vertica 7.0 adds several new statements and functions for improved usability and administration.

### ***Security Hash Functions***

Vertica Analytic Database 7.0 adds the following SQL hash functions to support SHA-1 and SHA-2 security encryption:

- [SHA1\(\)](#)
- [SHA224\(\)](#)
- [SHA256\(\)](#)
- [SHA384\(\)](#)
- [SHA512\(\)](#)

### ***Large Cluster Functions***

HP Vertica 7.0 adds the following functions to help you create and manage large cluster arrangements. See [Large Cluster](#) in the Administrator's Guide.

- [SET\\_CONTROL\\_SET\\_SIZE\(\)](#)
- [REALIGN\\_CONTROL\\_NODES\(\)](#)
- [RELOAD\\_SPREAD\(true\)](#)

## ***Fault Group Statements***

HP Vertica 7.0 adds several new **DDL** statements you use to create, alter, and drop fault groups. See [Fault Groups](#) in the Administrator's Guide.

- [CREATE FAULT GROUP](#)
- [ALTER FAULT GROUP](#)
- [DROP FAULT GROUP](#)
- [ALTER DATABASE](#)

## ***Flex Tables Statements and Functions***

HP Vertica 7.0 adds a new library to support loading and querying flex tables.

These are the statements for creating flex tables:

- [CREATE FLEX TABLE](#)
- [CREATE FLEX EXTERNAL TABLE AS COPY](#)

Once you have created a flex table and loaded data into it, these helper functions let you manipulate (or restore) the contents:

- [BUILD\\_FLEXTABLE\\_VIEW](#)
- [COMPUTE\\_FLEXTABLE\\_KEYS](#)
- [COMPUTE\\_FLEXTABLE\\_KEYS\\_AND\\_BUILD\\_VIEW](#)
- [RESTORE\\_FLEXTABLE\\_DEFAULT\\_KEYS\\_TABLE\\_AND\\_VIEW](#)

These are the map functions for the unstructured content in your flex table:

- [mapAggregate](#)
- [mapLookup](#)
- [mapContainsKey](#)
- [mapContainsValue](#)
- [mapItems](#)
- [mapKeys](#)
- [mapKeysInfo](#)

- [mapLookup](#)
- [mapSize](#)
- [mapToString](#)
- [mapValues](#)
- [mapVersion](#)
- [emptyMap](#)

## ***Local Temporary Views***

HP Vertica 7.0 adds the following new statement to create local temporary views: [CREATE LOCAL TEMPORARY VIEW](#).

## ***Approximate Count Distinct Aggregate Functions***

HP Vertica 7.0 provides three new aggregate functions. These functions calculate an approximate count of distinct values in a column or expression.

- [APPROXIMATE\\_COUNT\\_DISTINCT](#)
- [APPROXIMATE\\_COUNT\\_DISTINCT\\_SYNOPSIS](#)
- [APPROXIMATE\\_COUNT\\_DISTINCT\\_OF\\_SYNOPSIS](#)

## ***New and Modified System Tables***

This section lists the new and modified system tables in this release. See the [SQL Reference Manual](#) for details.

## ***New in V\_CATALOG Schema***

Added in HP Vertica 7.0:

- [HCATALOG\\_COLUMNS](#), [HCATALOG\\_SCHEMATA](#), [HCATALOG\\_TABLE\\_LIST](#), [HCATALOG\\_TABLES](#) provide metadata for tables stored in a Hive database that have been made available within Vertica Analytic Database using the HCatalog Connector. See [HCatalog Connector for Querying Data in Hive Data Warehouse](#).
- [FAULT\\_GROUPS](#) lets you view fault groups and their hierarchy in the cluster
- [CLUSTER\\_LAYOUT](#) shows the relative position of the arrangement of nodes participating in the database and the fault groups that affect them

- [LARGE\\_CLUSTER\\_CONFIGURATION\\_STATUS](#) shows the current control nodes (spread hosts) and the control message designations in the Catalog so you can see if they match
- [MATERIALIZED\\_FLEXTABLE\\_COLUMNS\\_RESULTS](#) lists the results of running the flex table function [MATERIALIZED\\_FLEXTABLE\\_COLUMNS](#).

## ***New in V\_MONITOR Schema***

Added in HP Vertica 7.0:

## ***Modified in V\_CATALOG Schema***

Enhanced in HP Vertica 7.0:

- [DATABASES](#) has a new VARCHAR column named LOAD\_BALANCE\_POLICY that sets the database's policy for native connection load balancing. See [Client Driver Connection Load Balancing](#) for more information.
- [TABLE\\_CONSTRAINTS](#) has a new VARCHAR column named TABLE\_NAME that lists the name of the table that contains the UNIQUE, FOREIGN KEY, NOT NULL, or PRIMARY KEY constraint.
- [TABLES](#) has the new boolean column is\_flextable, specifying whether a table was created as a flex table.

## ***Modified in V\_MONITOR Schema***

Enhanced in HP Vertica 7.0:

The V\_MONITOR schema system tables [SESSIONS](#), [SYSTEM\\_SESSIONS](#), [USER\\_SESSIONS](#), [CURRENT\\_SESSION](#), and [SESSION\\_PROFILES](#) have the following new columns:

- CLIENT\_TYPE
- CLIENT\_VERSION
- CLIENT\_OS

The [USER\\_SESSIONS](#) table has a new column, RUNTIME\_PRIORITY.

The [USER\\_LIBRARIES](#) table has nine new columns: AUTHOR, LIB\_BUILD\_TAG, LIB\_VERSION, LIB\_SDK\_VERSION, SOURCE\_URL, DESCRIPTION, LICENSES\_REQUIRED, SIGNATURE, and EPOCH.

## ***New Configuration Parameters***

This section lists new configuration parameters in HP Vertica 7.0. See [Configuration Parameters](#) in the Administrator's Guide for details.

## ***DBDCorrelationSampleRowCount***

HP Vertica 7.0 adds the `DBDCorrelationSampleRowCount` configuration parameter to specify the minimum number of rows that a table should have before analyzing column correlations.

For more information, see [ANALYZE\\_CORRELATIONS](#) in the SQL Reference Manual.

## ***EnableCooperativeParse***

HP Vertica 7.0 adds the `EnableCooperativeParse` configuration parameter. The parameter is on by default to permit the use of a `UDChunker` to parse load data cooperatively across cores on the initiator node. For more information, see [Developing User Defined Load \(UDL\) Functions](#) in the Programmer's Guide. For monitoring data loads, see the `LOAD_STREAMS` system table in the SQL Reference Manual.

## ***FlexTableRowSize***

HP Vertica 7.0 adds the `FlexTableRowSize` configuration parameter to specify the default size of the `__raw__` column when creating a flex table. The `__raw__` column in a flex table contains the loaded map data. Its data type is a `LONG VARBINARY`. The default value is 130000 (bytes). You can change the default up to a maximum of 32000000.

For more information, see [Setting FlexTable Parameters](#) in the Administrator's Guide.

## ***FlexTableDataTypeGuessMultiplier***

HP Vertica 7.0 adds the `FlexTableDataTypeGuessMultiplier` configuration parameter. This value is a multiplier (2.0 by default) to set column widths when casting columns from a `LONG VARBINARY` data type for flex table views. Multiplying the longest column member by this factor pads the column width to support subsequent data loads. Because there is no way to determine the column width of future loads, the padding adds a buffer to support values at least twice that of the previously longest value. For more information, see [Setting FlexTable Parameters](#) in the Flex Tables Guide.

## ***COPY Load Changes***

This section describes the changes affecting data loading and the `COPY` statement in Vertica Analytic Database 7.0.

### ***COPY Changes with Parallel Parsing Support***

This release includes significant changes to the `COPY` statement. The main goal of reimplementing `COPY` was to improve managing its three main data load phases:

- Obtaining data (source management)
- Filtering data (unzipping and converting)
- Parsing data (get data ready to load into a table)

In particular, improvements to data parsing (often the most time-consuming load phase) add parallel parser activities for delimited and fixed-width data loads. COPY now automatically allocates multi-threaded parsing tasks across cores on the server node for delimited and fixed-width parsers. Parallel parsing techniques can improve performance for large data loads, such as for file sizes exceeding many hundreds of megabytes.

**Note:** Parallel parsing activities are supported when the load file resides on the HP Vertica server. Parsing activities are not in effect with COPY LOCAL statements, nor are they applicable when loading data from HP Vertica-supported drivers.

## Disabling Parallel Parsing

Parallel parsing is on by default in this release, and controlled by the new ENABLECOOPERATIVEPARSE configuration parameter, see [New Configuration Parameters](#).

If your main requirements are to load many small files (totals in MBs of data), using parallel parsing might not increase load performance. You can disable parallel parsing capabilities as follows:

```
=> SELECT SET_CONFIG_PARAMETER ('EnableCooperativeParse',0);
```

## Saving Rejected Data to a Table

The parser phase of loading data is responsible for rejected data rows, if they occur. You can now save rejected rows to a table, rather than to a file.

Storing rejected data rows in a table also saves a message describing the reason (called an *exception*) a row was rejected. The option to save rejected data rows to a table is on by default. If you prefer to save rejected data rows and the reason for the rejection in files, do not use the REJECTED DATA AS TABLE clause.

**Note:** If you include the NO COMMIT and REJECTED DATA AS TABLE clauses in your COPY statement and the *reject\_table* does not already exist, Vertica Analytic Database saves the rejected-data table as a LOCAL TEMP table and returns a message that a LOCAL TEMP table is being created.

For more information about storing rejected load data in a table, see [Saving Load Rejections \(REJECTED DATA\)](#) in the Administrator's Guide.

## ***Parallel Load Library Now Obsolete***

The COPY improvements in this release preclude the need for the Parallel Load Library (Pload) functionality, which is now obsolete. The library is included in the HP Vertica RPM, but should not be used for new applications and will follow the policy described in [Retired Functionality](#). The Pload documentation has been removed.

## ***New EXPLAIN Plan Output***

In Vertica Analytic Database 7.0, EXPLAIN output has been simplified to help you identify running versus non-running nodes, as well as nodes that are not ephemeral. For details, see [Viewing Node Down Information](#) in the Administrator's Guide.



## SDK Enhancements

### *New Java SDK for UDX Development*

You can now use Java to develop user defined extensions (UDxs) . The Java SDK supports the following extension types:

- User Defined Scalar Function (UDSF)
- User Defined Transform Function (UDTF)
- User Defined Load (UDL)

All Java-based UDxs run in fenced mode.

See [Developing User Defined Functions in Java](#) and [Developing UDLs in Java](#) in the Programmer's Guide for more information.

### *UDChunker for Parallel Parsing Capabilities*

HP Vertica provides the UDChunker parser extension to the User Defined Load (UDL) facilities. While parallel parsing capabilities are now built in to the COPY statement parsers for delimited and fixed-width data (see [COPY Load Changes](#) ), UDL developers can add similar functionality to their own parsers, by implementing the new UDChunker method.

### *Implementing Parallel Parsing in UDL Parsers*

Application developers who write their own User Defined Load C++ applications can now implement parallel parsing support in their own parsers. In this release, the UDL has been extended with a new UDChunker class. The new class chunks input stream contents and pushes the data to multiple parser threads.

Implementing a UDChunker lets you specify unique capabilities for parsing activities as a multi-threaded procedure, as different parser instances share tasks across cores.

The new UDChunker functionality is backwards compatible with any existing UDLs. If you do not implement a chunker, the UDL uses the default implementation (returns NULL) and continues with the existing, single-threaded parser implementation.

For more information, see [Developing UDLs in C++](#) in the Programmer's Guide.

### *Rejection Files for Cooperative Parsing*

With parallel parsing in effect while loading very large files, multiple rejection files can exist, one file for each parser instance for the same input source. To differentiate rejection and exceptions files, COPY appends a unique number identifier to the file name of the rejected data file.

## Metadata for UDX Libraries

The C++, Java, and R SDKs now support adding metadata to UDX libraries. This metadata includes library's author, version information, and a URL that points users to more information about the library. This metadata appears in the [USER\\_LIBRARIES](#) system table after the library has been loaded into the database catalog.

For more information, see the following topics in the Programmer's Guide:

- [Adding Metadata to C++ Libraries](#)
- [Adding Metadata to Java UDX Libraries](#)

## VBR Backup Changes

HP Vertica 7.0 introduces this new functionality in the `vbr.py` utility:

- Add one or more nodes to the original database cluster before completing an object-level restore.
- Use a single `[Mapping]` section for all relevant parameters in the configuration file.

## Add Nodes to the Cluster Before Object Restore

Previous versions required the cluster configuration in effect when you created an object-level backup not to have any additional nodes when you restored a backup. You can now add additional nodes to the cluster after creating an object-level snapshot, and successfully restore the objects. For more information, see [Restoring Object-Level Backups](#) in the Administrator's Guide.

## Simplified Configuration File Mappings

When you use the `vbr.py --setupconfig` command to create a new configuration file, the file includes a single `[Mapping]` section. The section contains entries for each cluster node, with parameters representing each database node (`dbNode`), its associated backup host (`backupHost`), and backup directory (`backupDir`). The following example shows the `[Mapping]` section with parameters for two nodes:

```
[Mapping]
node0 = clust-1:/tmp/backup
node1 = clust-1:/tmp/backup
```

In previous versions, `vbr.py` configuration files included multiple `[Mapping]` sections, each containing three parameter and value pairs, `dbNode`, `backupHost`, and `backupDir`. The following example lists the mapping sections with parameters for two nodes, `node0` and `node1` :

```
[Mapping0]
dbNode = node0
backupHost = clust-1
backupDir = /tmp/backup/

[Mapping1]
dbNode = node1
backupHost = clust-1
backupDir = /tmp/backup/
```

```
[Mapping1]
dbNode = node1
backupHost = clust-1
backupDir = /tmp/backup/
```

The new mapping representation supersedes the previous presentation, but you can continue to use existing `vbr.py` configuration files.

**Note:** While the `vbr.py` utility continues to support the previous configuration file Mapping entries, you cannot use both formats in one configuration file. Create a new configuration file to use the new mapping style, or manually change all Mapping sections.

## Third-Party Tools Integration Updates

### *Connectivity Pack Enhancements*

This release includes an upgrade to the HP Vertica Microsoft Connectivity Pack. HP Vertica's Microsoft Connectivity Pack allows you to connect with your HP Vertica server to use Microsoft components that are previously installed on your system. The Connectivity Pack includes the ADO.NET client driver and additional tools for integration with Microsoft Visual Studio and Microsoft SQL Server.

This version of the HP Vertica Microsoft Connectivity Pack continues to support Visual Studio 2008 and SQL Server 2008. This release adds support for Visual Studio 2010 and 2012, and SQL Server 2012, and integration with the following Microsoft components.

- Business Intelligence Development Studio (BIDS) for Visual Studio 2008.
- SQL Server Data Tool - Business Intelligence (SSDT-BI) for Visual Studio 2010/2012.
- SQL Server Analysis Services (SSAS) for SQL Server 2008 and 2012.
- SQL Server Integration Services (SSIS) for SQL Server 2008 and 2012.
- SQL Server Reporting Services (SSRS).

### *Upgrading*

As of HP Vertica Release 7.0, you no longer need to uninstall previous versions of the ADO.NET driver before installing the new HP Vertica Microsoft Connectivity Pack. The current installer automatically uninstalls previous versions of the ADO.NET driver during the installation process and installs the newest version. Previous versions required that you use the Windows uninstall utility to first uninstall the ADO.NET driver; this is no longer necessary.

### *New Informatica Plug-in*

As of HP Vertica Release 7.0, by installing and configuring the HP Vertica Informatica Plug-in, you can use HP Vertica with Informatica PowerCenter both as a source and as a target database. You no longer need to use an ODBC connection in order to use HP Vertica as a data source, as was the case in previous releases.

The new plug-in includes multiple enhancements when using HP Vertica as reader (source) and/or writer (target); for information on these enhancements, review the manual, HP Vertica Informatica Plug-in Guide.

A major new performance feature is **EnableStreamingBatchInsert**. If you are connecting to a database that is running HP Vertica Release 7.x, for best performance, you should always

implement **EnableStreamingBatchInsert**. For setting this option, refer to the manual, HP Vertica Informatica Plug-in Guide.

## HCatalog Connector for Querying Data in Hive Data Warehouse

This release includes a new way to access data stored in Hadoop. The HCatalog Connector lets you create a schema in Vertica Analytic Database that maps to any Hive database available through a WebHCat (formerly known as Templeton) server. The Connector lets you query data stored within Hive the same way you query a native Vertica Analytic Database table. See [Using the HCatalog Connector](#) in the Hadoop Integration Guide.

## HDFS Connector

HP Vertica announces an enhanced Hadoop Distributed File System Connector. The version 7.0 connector supports both unauthenticated and Kerberos-authenticated connections with your Hadoop cluster. This connector replaces the Standard and Secure HDFS Connectors from previous versions.

## Documentation Updates

HP Vertica 7.0.x introduces a number of changes to the standard documentation set.

### ***New Look-and-Feel and Better Search***

We've updated and improved the look-and-feel of our PDF and Webhelp output. In addition, we've improved the Webhelp search experience over that provided in previous releases.

### ***New Guides***

To address new features, as well as make existing content easier to find, we've introduced a number of new guides to the documentation set:

New Document	Description
<i>Flex Tables Guide</i>	Describes how to use the <a href="#">Flex Tables</a> feature, introduced in HP Vertica 7.0.x
<i>Hadoop Integration Guide</i>	Describes how to integrate Apache's Hadoop and its components (such as HCatalog and HDFS) with HP Vertica.
<i>Best Practices for OEM Customers</i>	Describes best practices and tips for HP Vertica OEM customers who embed their HP Vertica database with their own application and deliver it as an on-the-premises solution

<i>Connectivity Pack for Microsoft</i>	Describes how to integrate Microsoft Visual Studio and Microsoft SQL Server with HP Vertica.
<i>HP Vertica Plug-In for Informatica Guide</i>	Describes how to connect to Informatica PowerCenter from HP Vertica.

## ***SDK and Supported Platforms Online Documentation***

In previous releases, the Vertica Analytic Database Webhelp did not include our SDK or Supported Platforms documentation. With release 7.0, you can access these guides as part of our online documentation:

- Supported Platforms
- Java SDK API documentation
- C++ SDK API documentation
- JDBC API documentation

Links to these guides appear in the left navigation pane of the online documentation. Content from these guides is now returned with any search of the HP Vertica online documentation.

## Deprecated Functionality in This Release

In version 7.0, the following HP Vertica functionality has been deprecated.

- The Kerberos `krb5` client authentication method. Use the `gss` method, described in [Implementing Client Authentication](#) in the Administrator's Guide
- The Administration Tools option `check_spread`
- The following `EXECUTION_ENGINE_PROFILES` system table counter values:
  - file handles
  - memory allocated
  - memory reserved

## See Also

For information about the meaning of obsolete and deprecated functionality, see [Retired Functionality](#).





## Retired Functionality

This section describes the three levels of HP Vertica retired functionality, Obsolete, Deprecated, and Removed.

**Obsolete.** HP Vertica processes obsolete functionality the same way as fully supported functionality (without warnings, interruptions, error, and so on). Release documentation notifies users that the functionality will be deprecated in a future release. Documentation about the feature may be removed when the functionality becomes obsolete.

**Deprecated.** HP Vertica accepts the request to execute deprecated functionality, but results in a user notification, typically a warning message or error message. HP Vertica may process the functionality, or not. Release documentation notifies users of each deprecated function or component, and documents any user-visible behavior for the deprecated feature.

**Removed.** The retired functionality is removed, and HP Vertica no longer supports the functionality. Release documentation notifies users that functionality is removed in this release.

The following functionality has been made obsolete, deprecated, or removed in past and present versions:

Functionality	Component	Obsolete Version	Deprecated Version	Removed Version
VT_ tables	Server	4.0	4.1	5.0
EpochAdvancementMode	Server	4.0	4.1	5.0
MANAGED load (server keyword and related client parameter)	Server, clients	4.1	5.0	
getNumAcceptedRows getNumRejectedRows	ODBC, JDBC clients	4.1	5.0	
use35CopyParameters	ODBC, JDBC clients	4.1	4.1	5.0
BatchAutoComplete	All clients	4.1	4.1	5.0
ReportParamSuccess	All clients	4.1	4.1	5.0
EnableStrataBasedMrgOutPolicy	Server	4.1	4.1	5.0
MergeOutPolicySizeList	Server	4.1	4.1	5.0
LCOPY (see Note section below table)	Server, clients	—	4.1 (Client) 5.1 (Server)	5.1 (Client)
backup.sh	Server	6.0		

Functionality	Component	Obsolete Version	Deprecated Version	Removed Version
restore.sh	Server	6.0		
Pload Library	Server	7.0		
copy_vertica_ database.sh	Server	6.0		
Ganglia	Server	—	6.0	
Volatility and NULL behavior parameters of CREATE FUNCTION	Server	6.0	6.1	
RESOURCE_ACQUISITIONS_ HISTORY system table	Server	—	—	6.0
Query Repository, which includes: SYS_DBA.QUERY_REPO table Functions: <ul style="list-style-type: none"> <li>CLEAR_QUERY_REPOSITORY()</li> <li>SAVE_QUERY_REPOSITORY()</li> </ul> Configuration parameters: <ul style="list-style-type: none"> <li>CleanQueryRepoInterval</li> <li>QueryRepoMemoryLimit</li> <li>QueryRepoRetentionTime</li> <li>QueryRepositoryEnabled</li> <li>SaveQueryRepoInterval</li> <li>QueryRepoSchemaName</li> <li>QueryRepoTableName</li> </ul> See <b>Notes</b> section below table.	Server	—	—	6.0
IMPLEMENT_TEMP_DESIGN()	Server, clients	—	6.1	
scope parameter of CLEAR_PROFILING	Server	—	6.1	
krb5 client authentication method	All clients	—	7.0	

Functionality	Component	Obsolete Version	Deprecated Version	Removed Version
Administration Tools option check_spread	Server, clients	—	7.0	
MERGE_PARTITIONS()	Server	7.0		
EXECUTION_ENGINE_PROFILES counters file handles, memory allocated, and memory reserved	Server	—	7.0	

## Notes

- LCOPY: Supported by the 5.1 server to maintain backwards compatibility with the 4.1 client drivers.
- Query Repository: You can still monitor query workloads with the following system tables:
  - [QUERY\\_PROFILES](#)
  - [SESSION\\_PROFILES](#)
  - [EXECUTION\\_ENGINE\\_PROFILES](#)

In addition, HP Vertica Version 6.0 introduced new robust, stable workload-related system table:

- [QUERY\\_REQUESTS](#)
- [QUERY\\_EVENTS](#)
- [RESOURCE\\_ACQUISITIONS](#)
- The RESOURCE\_ACQUISITIONS system table captures historical information.
- Use the Kerberos gss method for client authentication, instead of krb5. See [Authentication Record Format](#).



## We appreciate your feedback!

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on New Features Guide (Vertica Analytic Database 7.0.x)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to [vertica-docfeedback@hp.com](mailto:vertica-docfeedback@hp.com).