|  |
| --- |
| Threeglav.com |
| StreamHorizon Developer Guide |
| Version 3.0.2 |



|  |
| --- |
|  |



Table of Contents

[Introduction 2](#_Toc381340572)

[Getting started 2](#_Toc381340573)

[Installation requirements 2](#_Toc381340574)

[Installation and directory structure 2](#_Toc381340575)

[Mapping dimensions 3](#_Toc381340576)

[Processing input data (feeds) 3](#_Toc381340577)

[Creating output files 3](#_Toc381340578)

[Inserting directly into database 3](#_Toc381340579)

[Configuration parameters 3](#_Toc381340580)

[Context attributes 3](#_Toc381340581)

[Creating plugins 3](#_Toc381340582)

[Clustering StreamHorizon 3](#_Toc381340583)

[Performance tuning 3](#_Toc381340584)

[Output type 3](#_Toc381340585)

[Database 3](#_Toc381340586)

[Thread pools 3](#_Toc381340587)

[storage, Read and write buffers 3](#_Toc381340588)

[Caches 3](#_Toc381340589)

[Clustering 3](#_Toc381340590)

[Remote commands 4](#_Toc381340591)

[Flushing dimension cache 4](#_Toc381340592)

[Monitoring 4](#_Toc381340593)

[Miscellaneous 4](#_Toc381340594)

[Common mistakes 4](#_Toc381340595)

[Planned platform extensions 4](#_Toc381340596)

# Introduction

## Getting started

## Installation requirements

Installation of StreamHorizon platform is simple and easy. Download installation archive and uncompress it to your hard drive in directory of your choice (we will refer to this directory as $ENGINE\_HOME from now on).

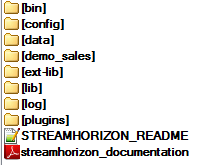
There are very few requirements that need to be met in order to run StreamHorizon.

1. Mainstream operating system (Linux, Windows, Solaris)
2. JDK 1.7+ (we recommend Oracle HotSpot)
3. Database that supports JDBC (most likely data produced by StreamHorizon will go into some kind of database)

Even if you intend to run multiple (clustered) StreamHorizon instances there is no need to install it more than once per physical machine.

## Installation and directory structure

After installing StreamHorizon platform to $ENGINE\_HOME directory you will see following directory structure inside:



|  |  |
| --- | --- |
| Directory name | Purpose |
| **$ENGINE\_HOME/bin/** | All startup scripts should be placed here. There are few default scripts that can be used for starting single instance. If you are adding new startup scripts always place them in this directory. |
| $ENGINE\_HOME/config/ | All configuration files are here:   * What engine should log and at what level * Here is where you copy your main engine configuration file(s) * Configuration for clustered caches used by engine |
| $ENGINE\_HOME/data/ | This is engine private directory used for housekeeping. You should not add, delete or modify anything inside. |
| $ENGINE\_HOME/demo\_sales/ | Demo feature showing capabilities of engine (performance and functionality). |
| $ENGINE\_HOME/ext-lib/ | Additional dependencies needed by plugins. |
| $ENGINE\_HOME/lib/ | External dependencies needed by engine are placed here. You should not add, delete or modify anything inside. |
| $ENGINE\_HOME/log/ | Engine log files will be found here. |
| $ENGINE\_HOME/plugins/ | Java plugins. Plain \*.java files can be placed here and engine will compile and use them (when configured to do so). |

# Mapping dimensions

# Processing input data (feeds)

# Creating output files

# Inserting directly into database

# Configuration parameters

# Context attributes

# Creating plugins

StreamHorizon platform provides extension points where users can add their own processing logic written in SQL, Java or shell scripts.

# Clustering StreamHorizon

# Performance tuning

## Output type

## Database

## Thread pools

## storage, Read and write buffers

## Caches

# Clustering

# Remote commands

StreamHorizon engine accepts remote commands via HTTP. This feature can be turned off. Currently we support following commands:

## Flushing dimension cache

In case when dimension cache is stale (for example data was loaded into table bypassing StreamHorizon engine) it is important to inform engine to stop using old data. This can be done by executing

http://localhost:<remoting.server.port>/flushDimensionCache/?dimension=DIMENSION\_NAME\_AS\_DEFINED\_IN\_CONFIG

When this is executed engine will go through a set of steps:

1. Wait for all threads to finish their current work
2. Pause all threads so that they do not accept new work
3. Flush data cached for specified dimension
4. Notify all threads that they can continue processing data

Please note that after flushing dimension cache there is expected slow-down in execution because cache will be populated as needed and this might require execution of database queries.

# Monitoring

StreamHorizon exposes internal processing metrics via JMX. It is also possible to use extension points to collect and calculate metrics about performance of engine. See demo provided with installation for more details (sh\_metrics table).

# Miscellaneous

## Common mistakes

## Planned platform extensions