

# Collector

Machbase Collector is a tool that extracts log data and inputs the data to the Machbase database in real time after converting it.

Machbase Collector can collect log data in real time and input it through the network by being installed in a separate device from the Machbase server. It operates as a separate process from the Machbase server and can run multiple Collectors at the same time. Each Collector process processes one data source.

## Concept

---

### Index

---

- Concept
- Characteristic
  - Consistent Interface
  - Collection Method
  - Log Data Types
  - Easily Supported Custom Logs
  - Prevents Data Loss in the Event of a Failure
  - Ensures High Availability
  - Integrated Monitoring Through MWA
  - Log Pre-

Processing  
Using  
Python  
Script

The above figure shows Node-2 and Node-3 Collectors collect data and input to Node-1 where the database server is installed.

- In Node-2 and Node-3, the Collectors can be seen **as running a separate process on a specific log file** to send data.
- You can see that each collector process gets **detailed information about the log data using a given tpl file**.
- **The Collector manager** is installed in each Node, manages and monitors the Collector process running on that node.

## Characteristic

The main features of the Machbase Collector are described below.

### Consistent Interface

Machbase does not require any additional programs in addition to SQL-based commands to execute the Collector. Simply use the following command to manage and monitor the Collector.

```
CREATE Collector MANAGER LOCALHOST AT '127.0.0.1:9999';
CREATE Collector LOCALHOST.MYADP FROM 'syslog.tpl';
ALTER Collector LOCALHOST.MYADP START;
```

### Improved Data Collection Performance

Machbase Collector is designed to collect data with a separate Collector for each log data type so each process can process each log file at high speed.

Since separate processes each process log data, they are not affected by other log file processing. Collector is executed with optimized code for each log type, and data is input with dedicated protocol that minimizes resource usage, so that the best performance can be obtained.

### Collection Method

Collectors can be used to collect log data in a variety of ways. The data collection method can be set by modifying the tpl file. The following collection methods are supported.

Method Name	Description
FILE	Collects files from local host.
SFTP	Collects files from remote host.
SOCKET	Collects data coming into port.
ODBC	Collects files from other databases.

### Log Data Types

Machbase Collector supports regular expressions for various types of log data.

The user can simply modify the existing regular expression to analyze various log files. Currently, the following log types are supported.

Regular Expression File Name	Supported Type	Data Default Location (can be modified)
machbase.rgx	Machbase trace log	\$MACHBASE_HOME/trc/machbase.trc
apache_access.rgx	Apache web server access file	/var/log/apache2/access.log
apache_error.rgx	Apache web server error file	/var/log/apache2/access.log
syslog.rgx	syslog file	/var/log/syslog
custom.rgx	Custom type	Custom file

### Easily Supports Custom Logs

Machbase Collector can process various kinds of log files that can be represented as regular expressions.

Even if you do not have a log file, you can test sample log messages and regular expressions using machregex.

## **Prevents Data Loss in the Event of a Failure**

Machbase Collector provides the ability to correctly retransmit data that failed to be transmitted in the event of a failure, after the failure has been resolved.

When a failure occurs, the Collector records the last location it sent to the server, resets the fault, and then resends the data from that location. So, even if you do not write any additional operations or code to overcome the obstacle, you can transfer it to the server without losing any data.

## **Ensures High Availability**

To ensure high availability of services, multiple Collectors can operate simultaneously on the same data source, and these Collectors transfer data to different Machbase servers.

In this way, even if an error occurs in the Machbase server, the same data is continuously stored in another server, so that the service can be continued.

After resolving the error and restarting the server, the Collector can retransmit the untransmitted log data correctly, thus automatically replicating the data to provide high availability.

## **Integrated Monitoring Through MWA**

Machbase Collector manager synchronizes the Collector's execution information to the Machbase server.

Using this, it is possible to perform integrated monitoring through MWA (Machbase Web Admin).

By using MWA, it is possible to monitor various status information of the running Collector and the status information of the server running the Collector in real time.

## **Log Pre-Processing Using Python Script**

You can write a Python script to manipulate the Collector before it processes the data.

The input data can be processed so as not to input unnecessary data, or the parsed data can be changed.