

# Backup and Restore Guide

Fabasoft app.telemetry Backup and Restore Guide

Copyright ©

Fabasoft R&D GmbH, A-4020 Linz, 2019.

All rights reserved. All hardware and software names used are registered trade names and/or registered trademarks of the respective manufacturers.

These documents are highly confidential. No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

Distribution, publication or duplication is not permitted.

# Contents

<b>1 Introduction</b>	<b>4</b>
<b>2 Restoring a Fabasoft app.telemetry Server for Data Analysis</b>	<b>4</b>
2.1 Problem resolution	4
<b>3 Migrating a Fabasoft app.telemetry Server to another Machine</b>	<b>5</b>
<b>4 Backup and Restore Data</b>	<b>5</b>
4.1 Configuration	5
4.2 Certificates	6
4.3 Software-Telemetry Data	6
4.3.1 Source Directory	6
4.3.2 Transferring Data	7
4.3.3 Target Directory	7
4.4 Request Data (Database)	7
4.4.1 Select tables to transfer	7
4.4.2 Select parts of the records of your tables	8
4.4.3 Transfer Database Data using a Database Backup	8
4.4.4 Transfer selected Database Data using Database Backup	9
4.4.5 Transfer database tables using Microsoft SQL Server using bcp	9
4.5 Fabasoft Folio Name Resolution	11

# 1 Introduction

Fabasoft app.telemetry is an Application Performance Monitoring product designed for the use with business-critical applications on distributed production systems.

This guide describes how to migrate an existing Fabasoft app.telemetry server to another system. Furthermore this document describes how to export only selective data segments to an offline system for data analysis.

The procedures described are based on standard operating system and database system calls. There is no support implemented in Fabasoft app.telemetry for conversion of data formats between different versions of databases or for the migration between PostgreSQL and Microsoft SQL Server.

## 2 Restoring a Fabasoft app.telemetry Server for Data Analysis

To restore a dataset for data analysis to another server you need the following files:

- configuration file (`infra.xml`)
  - Software-Telemetry data (file system)
  - Request data (database)
  - Fabasoft Folio name resolution file (`fsedata.xml`) – optional
1. To start setting up your target system you have to setup a clean Fabasoft app.telemetry server as described in the “*Installation Guide*”.
  2. Stop all Fabasoft app.telemetry services/demons.
  3. Remove all telemetry data.
  4. The process of backup and restore of all of these data is described in the following sections.
  5. When you restore a configuration, this configuration would startup on the new server and would start to connect to all of the configured agents. To avoid this, you have to manually deactivate all agents by setting the status attribute to "2" for all `<Agent>` elements.
  6. Then start the Fabasoft app.telemetry services and the webserver.
  7. Switch to the “*Edit-View*” and edit the “*Database Connection*” to point to your database server.

### 2.1 Problem resolution

- No requests in the “*Telemetry/Log Pool*” view
  - Check database configuration
  - Check if the app.telemetry server may read the database tables
  - Check if table name matches `prefix + table name` of the log pool
- “*Request not found*” when selecting a request
  - Check location of telemetry data
  - Check that the app.telemetry service may read the telemetry data
  - Make sure the app.telemetry service has been started after restoring the data
- No Resolution of Fabasoft “*COO*”-addresses to names/references.
  - Check if `fsedata.xml` file is in the correct location

- Check if the Fabasoft app.telemetry worker may read the file
- Restart the Fabasoft app.telemetry worker

## 3 Migrating a Fabasoft app.telemetry Server to another Machine

When you setup a new server as the replacement of a running Fabasoft app.telemetry server instance, you take the following steps:

1. Setup Fabasoft app.telemetry server on the target server
2. Stop Fabasoft app.telemetry services on the target server
3. Remove all Software-Telemetry data
4. Optional: set alternative “*SoftwareTelemetryDataPath*”
5. Stop Fabasoft app.telemetry services on the source server
6. Backup and restore data as describe in the next chapter
7. Optional: Manually patch the database configuration in the `infra.xml`
8. Start Fabasoft app.telemetry services on the target server

You may reduce downtime of the system by incremental copy of Software-Telemetry data before you stop the source server. Software-Telemetry directories of past days are changed only in case of writing additional requests to still running sessions or in case of deleting session, so ensure that no relevant changes will be processed on those directories and it is save to copy those data beforehand.

**Warning:** Remember that it is essential, that you have the Software-Telemetry data directory of the current day 100% in sync after you stopped your source system before you start your target server.

You may transfer request data any time with the consequences of missing data in your responsibility. Bear in mind, that request statistics may be cached on the database (`stattime/statvalue` tables) as well as in memory. To recalculate those statistics, truncate the `stattime/statvalue` tables and restart the Fabasoft app.telemetry server service when the data migration is done.

## 4 Backup and Restore Data

### 4.1 Configuration

The Configuration of a Fabasoft app.telemetry instance is stored in an XML file and related encryption keys. This files are normally named “`infra.xml`”, “`encryption.key`” and “`encryption.pem`” and are located under the following directories

- **Linux:** `/etc/app.telemetry/configuration` (or in `/etc/app.telemetry`)
- **Microsoft Windows:** “`%ProgramData%\Fabasoft app.telemetry\configuration`”, which is located in “`C:\ProgramData\Fabasoft app.telemetry\configuration`” on most systems ( or in “`C:\ProgramData\Fabasoft app.telemetry`”)

As it is possible to specify the filename of the configuration file using command line parameters of the services, you may have to look at the respective settings to locate the active configuration file.

## 4.2 Certificates

Since Fabasoft app.telemetry 2015 Update Rollup 1 the communication between the Fabasoft app.telemetry services is secured using certificates. As all Fabasoft app.telemetry services only respond to known peers identified by their certificates, it is essential to backup at least the Fabasoft app.telemetry server certificate, which is used to contact all agents.

The certificates and the hashes of the trusted certificates are stored in the following folder

- **Linux:**  
`/etc/app.telemetry/{agent|configuration|server|sync|worker|ecomm|webserver}/`
- **Microsoft Windows:** “%ProgramData%\Fabasoft app.telemetry\  
{agent|configuration|server|sync|worker|ecomm|webserver}”

Up to four files – depending on the service – are stored in each folder:

- `srv_certificate.pem`: The certificate of the RPC used for encryption.
- `cli_certificate.pem`: The client certificate that is used to authenticate the service with another service.
- `cli_certificate.fingerprint`: The SHA-256 fingerprint of the client certificate automatically used by dependent services.
- `trusted_certificates.cfg`: List of SHA-256-hashes of client certificates of known services.

Certificates are created on service startup if not existing and the respective trusts are established for all services located on the Fabasoft app.telemetry server. All other agents will accept the first app.telemetry server contacting the agent and will add the certificate hash to their trusted certificates. So restoring an app.telemetry server requires the `server/cli_certificate.pem` to be restored so that the agents will accept the server’s incoming connections. It is recommended to backup and restore all certificate-, trust- and fingerprint-files to make sure, that all trusts are established correctly.

## 4.3 Software-Telemetry Data

Software-Telemetry data is stored on the file system and contains detailed information about the activities that occurred during request processing on the measured applications involved. So this data is the basis for detailed request analysis.

You can backup and restore telemetry data by copying data on file basis.

The files are organized on a daily basis in folders that are named by the date when the data have been received by the Fabasoft app.telemetry server. The time base for this structure is UTC, so be careful to copy the correct folder for data analysis of specific time ranges.

Inside the daily folders Software-Telemetry data is structured in several subfolders that contain data of different aspects of the telemetry data. Most of these folders are required to be able to correctly load request data for data analysis so there is no way to reduce data significantly by skipping any of the subfolders.

### 4.3.1 Source Directory

The root folder of the telemetry data is

- **Linux:** `/var/opt/app.telemetry/server/telemetry`
- **Microsoft Windows:** “%ProgramData%\Fabasoft app.telemetry\server\telemetry”

This default location may have been changed to place Software-Telemetry data onto a different data partition by

- Linux: setting the “*SoftwareTelemetryDataPath*” entry in `/etc/app.telemetry/server.conf`
- Microsoft Windows: setting the “*SoftwareTelemetryDataPath*” string value under “`HKLM\SOFTWARE\Fabasoft app.telemetry\Server`” in the registry

### 4.3.2 Transferring Data

As telemetry data will never be changed after being written (with the exception of adding requests to running telemetry sessions and deleting sessions) it is safe to copy the content of past days without stopping the `app.telemetry` server process. When copying the current day some files may be incomplete or locked by the `app.telemetry` server process.

If you move the active Fabasoft `app.telemetry` instance to another server, make sure to stop the Fabasoft `app.telemetry` server service before you copy the telemetry data of the current day. This is essential also because the telemetry data contain the latest request id in use. So if you fail to restore data correctly you may run into duplicate request id problems.

Compression tools may help transferring data by reducing the count of files and the size by compression, but do not expect high compression rates, because the largest portion of the data – contained in the “`rawdata`” folder – is partly compressed internally.

### 4.3.3 Target Directory

Like on the source system copy the backed up files to

- Linux: `/var/opt/app.telemetry/server/telemetry`
- Microsoft Windows: “`%ProgramData%\Fabasoft app.telemetry\server\telemetry`”

In case you want to put the data into a different folder, specify the location in the following configuration:

- Linux: set the “*SoftwareTelemetryDataPath*” entry in `/etc/app.telemetry/server.conf`
- Microsoft Windows: set the “*SoftwareTelemetryDataPath*” string value under “`HKLM\SOFTWARE\Fabasoft app.telemetry\Server`” in the registry

## 4.4 Request Data (Database)

Request data recorded by Fabasoft `app.telemetry` may be written into a database. This request information is also the key information to access detailed Software-Telemetry data. So make sure when you backup your data to provide also a backup of the particular time frame of the request data of the specific log pool tables.

There are several ways transferring database data to another system.

The most convenient way is to simply backup the database and restore it on another database server. This process is fully supported by database tools.

If you do not want to copy all of your data you have to extract parts of the data. You may either do this by copying data of single tables to backup files or you simply create a temporary database, fill this database with the data needed and use backup and restore procedures of the database system to transfer that database to the target system.

### 4.4.1 Select tables to transfer

If you choose to selectively transfer request data you have to copy parts or all of the records of the selected log pool tables.

The name of a log pool table consists of the “*Database Table Prefix*” specified for the log pool (referred to as “*prefix*”) and one or more tables declared by the entries of the “*Log Definition*”

*Columns*” section of the log pool. If there are no entries in “*Log Definition Columns*” a single “*request*” table will be generated holding the default properties of each request. Concatenate the prefix and the “*Database Table*” name to get the effective name of the table on the database.

There are 2 kinds of data tables, the “*Base Table*” which will always be there and one or more optional “*Additional Tables*”, which hold data which may occur multiple times in a single request (e.g. the “*query*” table in a Fabasoft Folio Webservice log pool). It is sufficient to copy data of the “*Base Table*” only as a basis for the telemetry data analysis of the request, whereas “*Additional Tables*” data may help selecting the right requests or to identify common properties of problematic requests. “*...stattime*” and “*...statvalue*” tables represent aggregated statistics based on “*Base Table*” data. These tables and their content will be regenerated on demand, so they need not be copied to the target system.

#### 4.4.2 Select parts of the records of your tables

You may backup all or only selected records of your base table. Common criteria for selecting are based on the *id* or the “*starttime*” column of the table. Whereas the *id* is the unique key of the row which is assigned by the Fabasoft app.telemetry server at the time when it started processing of the request, the timestamp is the GMT time of the device recording the telemetry data. As time synchronization between the devices in the infrastructure may not be accurate, this timestamp may not represent “natural” order of the requests, whereas the *id* represents an order in which the data have been received and processed by the app.telemetry server.

Only the “*Base Table*” includes the “*timestamp*” column, the “*Additional Tables*” connect to the “*Base Table*” data by the “*id*” column. So if you transfer data from “*Additional Tables*” make sure to transfer all needed records by selecting all *ids* referred to by the “*Base Table*” entries.

If you export data selected by time range and you have “*Additional Tables*”, the most convenient way of selecting the data is to find a minimum and maximum request *id* of that time range in the “*Base Table*” and select all records of “*Base Table*” and all “*Additional Tables*” based on that *id* range, instead of selecting the “*Base Table*” data by “*starttime*” and joining to the “*Additional Tables*” by *id*. This procedure may not be 100% accurate but may be much faster.

#### 4.4.3 Transfer Database Data using a Database Backup

The easiest way of transferring request data is to backup and restore the complete database. Use the tools of your database system to perform this task.

##### 4.4.3.1 Database Backup and Restore under Microsoft SQL Server

To back up a database use “Microsoft SQL Server Management Studio” or execute an SQL statement like:

###### Example: SQL Command

```
BACKUP DATABASE [apmdb] TO DISK = N'c:\temp\apmdb.bak'
```

This will generate a single file containing all information stored in the database.

On the target system create and restore the database using the Management Studio

##### 4.4.3.2 Database Backup and Restore under PostgreSQL

Use `pg_dump` and `pg_restore` to transfer databases.

E.g. to dump the whole database “apmdb” use



#### Example: PostgreSQL Command – Dump

```
pg_dump apmdb > apmdb.sql
```

To restore that backup execute

#### Example: PostgreSQL Command – Restore

```
psql -d apmdb -f apmdb.sql
```

You may export single tables only using

#### Example: PostgreSQL Command – Export

```
pg_dump -t "APMwebQuery" apmdb > apmwebquery.sql
```

As app.telemetry may use mixed case table names, it is necessary to escape table names correctly like in the statement above.

### 4.4.4 Transfer selected Database Data using Database Backup

A convenient way of transferring selected Database Data is to create an additional temporary database on the database system, copy the selected data to the temporary database and transfer this database to the target system using database system tools.

#### 4.4.4.1 Transfer selected Database Data using Database Backup using Microsoft SQL Server

With `pg_dump` you can backup selected database tables. To backup selected rows of a table, you have to copy the data into a new table using “`SELECT INTO`” and backup this table.

Make sure to recreate the indexes on the tables for “*Base Tables*” on the target system using the following commands:

#### Example: SQL Commands – Create Index

```
CREATE UNIQUE INDEX "apmweb_pkey" ON "apmweb" (id);  
CREATE UNIQUE INDEX "apmweb_idx" ON "apmweb" (starttime, id);
```

... and for „*Additional Tables*“ using:

#### Example: SQL Commands – Create Index

```
CREATE INDEX "apmwebquery_idx" ON "apmwebquery" (id);
```

### 4.4.5 Transfer database tables using Microsoft SQL Server using bcp

One way to transfer tables is to export table definition and table data to files, transfer these files to the target system and import them there.

#### 4.4.5.1 Backup table on Microsoft SQL Server

When transferring data from one database to another you have to create the database table on the target system. To extract the table structure use “*Microsoft SQL Server Management Studio*” and

select “Script Table as ...” > “CREATE To ...” > “File” from the context menu of the particular table and save the table definition to a file.

By default an app.telemetry log pool database table has two indexes, one of them is the primary key index on the id column. This index is already included in the table definition. The second is the index on starttime and id which you have to create manually on the target system. You don't need to back up the structure of this index, it is more efficient to create the index after data has been restored on the target system. You may have created additional indexes for to support special query restrictions, so take care about indexes yourself when migrating data to another server.

Microsoft SQL Server provides a tool called “bcp” for “Bulk Copy” operations. With this tool you can easily copy all or parts of the records of a table to a file.

#### Example: SQL Command – BCP

```
bcp "select * from [apmdb].dbo.[apmweb]" queryout "apmweb.bcp" -N -S localhost -T -E
```

Replace apmdb with your database name and apmweb with the name of your log pool table. The option “-N” directs bcp to output data in a native format, “-S” is followed by the database server name, “-T” means “Trusted Connection” and “-E” is there to keep identity columns.

As you specify a query in the command line you may also provide restrictions to the records being exported e.g. by specifying a time range.

The following query will export the entries with a starttime between Feb 18th, 2013 00:00 (UTC) and Feb 19th, 2013 00:00 (UTC).

#### Example: SQL Command

```
select * from [apmdb].dbo.[apmweb] where starttime between '20130218' and '20130219'
```

### 4.4.5.2 Restore table on Microsoft SQL Server

Create a database using “Microsoft SQL Server Management Studio”.

Recreate the tables in the database by executing the scripts with the table definitions generated on the source system. Make sure to execute the statements in the proper database.

Use bcp to import the datafile into the table:

#### Example: SQL Commands – BCP Import

```
bcp [newapmdb].dbo.[apmweb] in "apmweb.bcp" -N -S localhost -T -E -b 10000
```

Recreate the secondary index on the table:

#### Example: SQL Commands – Recreate Index

```
CREATE UNIQUE NONCLUSTERED INDEX [apmweb_idx] ON [dbo].[apmweb] ([starttime], [id])
```

## 4.5 Fabasoft Folio Name Resolution

1. Transfer the `fscdata.xml` file from the source system to the target system. The file is located under
  - o **Linux:** `/etc/app.telemetry/`
  - o **Windows:** “%ProgramData%\Fabasoft app.telemetry”, which is located in “C:\ProgramData\Fabasoft app.telemetry” on most systems
2. Make sure that the Fabasoft app.telemetry worker service may read this file.
3. After the next restart of the Fabasoft app.telemetry worker service the name resolution is in action.