dynatrace

Upgrade and migration guide

Please direct questions about your Dynatrace product or comments on this document to:

Dynatrace Customer Support
https://www.dynatrace.com/support/contact-support/

Upgrade and migration guide 5
Prepare to upgrade 8
Is this the right guide? 8
Get an overview 9
Check new requirements 9
Double-check on critical fixes 9
Update the source installation 10
Check Compatibility 10
Split server instances 10
Review custom-sized installations 10
Take notes 10
License 11
Download files 11
Running two versions 12
Component compatibility 13
Compatibility and Support 13
Agent-Collector Compatibility 13
Collector-Server Compatibility 14
Client-Server Compatibility 15
Running versions in parallel for migration 16
Upgrade server, collectors, and clients 18
1. Install the new Server, Collectors, and Clients 19
2. Deactivate and Upgrade License 20
3. Shut down the old version 20
4. Create backups 20
5. Services migration (Server, Collector and Memory Analysis Server) 20
6. Auto-start new services 22
7. Start the new Server components 22
8. Client migration 23
9. Start the new Client 23
10. Activate licenses for the new Server 23
11. Select server sizing 23
12. Apply update to new AppMon 24
13. Connect the Performance Warehouse 24
14. Configure session storage 24
15. Start new Collectors 24
Upgrade agents 25
1. Bootstrapped Agents 25
2. Non-bootstrapped Agents 26
3. Bootstrap Agents 26
4. z/OS Agents 26
Upgrading Bootstrap Agents 28
Preparation 28
Bootstrap agent upgrade process 29
Uninstall the outdated agents 29

Upgrading zOS Agents 31
Requirements 31
Upgrade procedure 32
Final migration checks 35
1. Manual configuration migration 35
2. Confirm operation of the new AppMon Server 36
Frequently Asked Questions 38
Troubleshooting migration 42
Case 1: Collectors: Large class cache 42
Case 2: Many agentres files 42
AppMon support 43
Collecting support information 43
Performance warehouse migration troubleshooting 45
Keystore Migration Troubleshooting 47
Symptoms 47
Solution 1 48
Solution 2 48
Migration Tool 52
Migration tool details 53
Overview 53
Migration Tool requirements 54
Use the Migration Tool 54
Examples 54
Migration tool syntax 58
Migrating server instances to separate installations 61
Overview 61
Separation 62
Manual steps 63
Running multiple server instances on one system 64
Server.config.xml changes for server Instance removal 66

Upgrade and migration guide

Looking for the 6.5 version of this page? Open in 6.5 documentation

Related pages

Dynatrace AppMon 2017 May Release Notes

Migration tool details

Deployment Guide

Downloads

dynatrace-migration tool

Product Downloads

Migration steps

Prepare to upgrade

Upgrade server, collectors, and clients

Upgrade agents

Final migration checks

Watch and learn

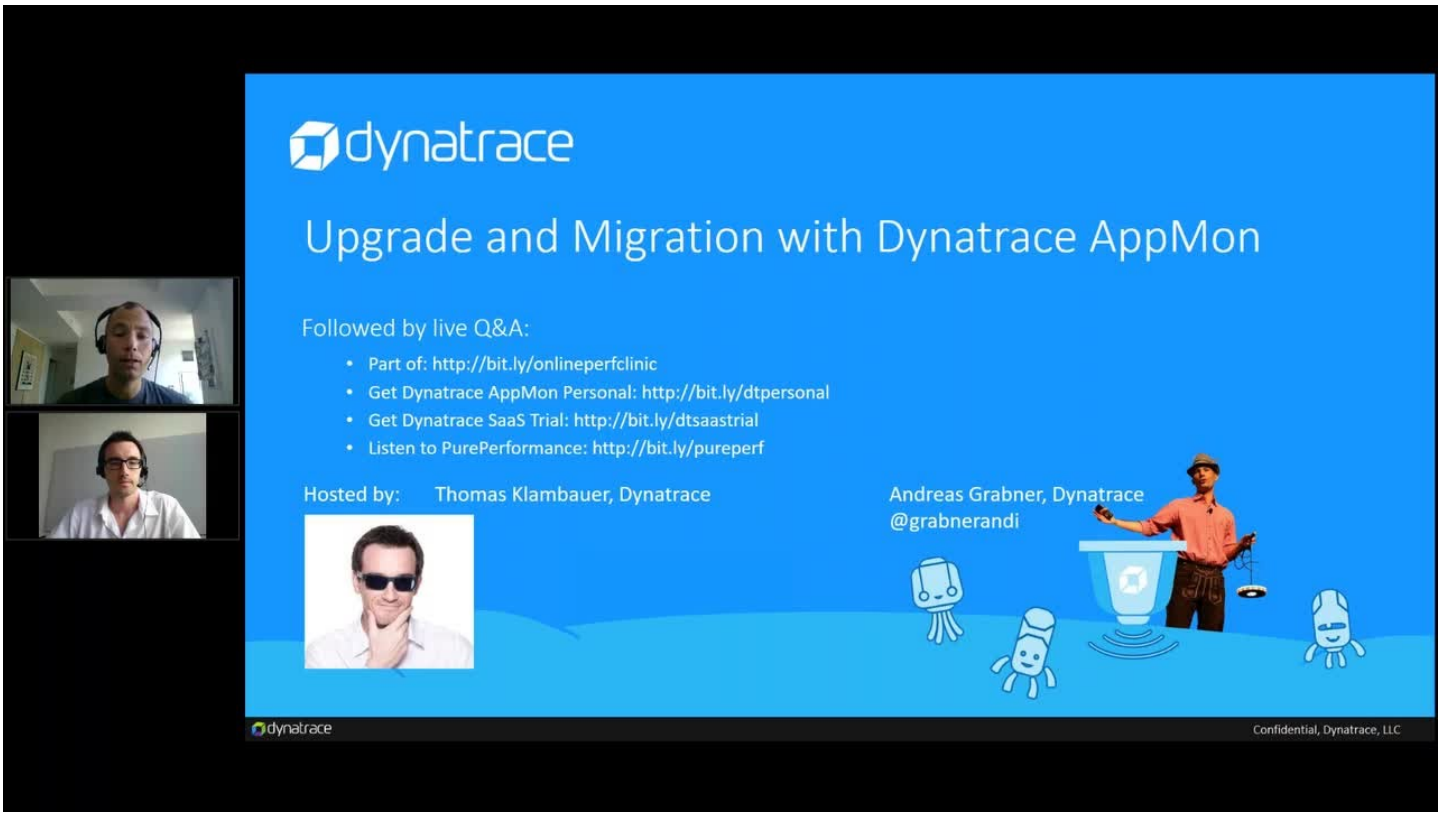
Deploying, Upgrading, and Migrating Dynatrace AppMon

Follow the instructions in this guide to:

- Upgrade your AppMon installation
- Migrate your AppMon installation to another host machine

1. **Prepare for your migration**
   Before you start, review your current installation releases for Servers, Collectors, and Agents to determine if those components can be directly upgraded to the new release. Also, review the System requirements to make sure you are using supported hardware and platforms.
2. **Install the new components and transfer data to it**
   Install the new components and **use the latest Migration Tool** to move your current data and configurations to the new installation. Check the version displayed when you run the tool from the command line. See Migration tool details for manual steps and edits needed to complete migration, and for more information on using the Migration tool.
3. **Upgrade your Agents**
   Most Agents automatically upgrade with the new release. In some cases, you may need to manually upgrade your Agents.
4. **Final checks and verification** Start AppMon and ensure Agents are sending data and your PurePaths and dashboards display properly.

See also the following Performance clinic videos for an upgrade demo:



Prepare to upgrade

Looking for the 6.5 version of this page? Open in 6.5 documentation

Related pages

Dynatrace AppMon 2017 May Release Notes

Deployment Guide

Component compatibility

License upgrade

Downloads

dynatrace-migration tool

Product Downloads

Migration steps

Upgrade server, collectors, and clients

Upgrade agents

Final migration checks

Review this checklist to ensure you are ready to migrate.

Is this the right guide?

- This guide only applies if you want to upgrade or migrate to version 7.0. Otherwise see the documentation of that respective version.
- For AppMon 6.2 and later, a direct upgrade is possible.

- For AppMon 6.1 and earlier, **you must migrate to 6.5 first**. See the 6.5 Documentation for instructions.
- For Proof of Concept (POC) and evaluation installations, a clean new installation is highly recommended.
- For Independent Software Vendor editions (PTC Windchill), use the documentation supplied by the vendor.

Get an overview

Create an inventory of your installed Collectors, Agents, and Clients to understand the scope of the upgrade.

Check new requirements

Verify the system requirements:

- Make sure the **Hardware requirements** is fulfilled for your sizing. AppMon 2017 May is more strict in hardware requirements than previous releases.
- AppMon 2017 May and later **does not** include a 32-bit Client anymore. See Set up the Client and Set up the Webstart Client for more details.
- This migration does **not** support a Server-embedded Collector. Install a stand-alone service for your current version before you upgrade.
- An embedded Performance Warehouse is not supported for migration. Switch to an external one first, as described here.
- If LDAP authentication is enabled, users need to be members of at least one AppMon LDAP group to log in. See LDAP for details.

Double-check on critical fixes

If you recently **received an update** with a critical fix or change in response to a support case, check the latest update Release notes to see if the issue id (for example, JLT-160157) is included.

This release together with its latest public update contains all changes of:

- Public update 6.5.18 and earlier (not affected by JLT-192329)
- Public update 6.3.25 and earlier
- All updates released for versions 6.2 and earlier

Later versions will be included in following 2017 May public updates. If you are missing required changes, contact support before scheduling the upgrade to clarify and resolve.

Update the source installation

It is recommended to roll out the most recent public update on the old installation before the upgrade.

Check Compatibility

We recommend to upgrade all Collectors and Agents along with the Server. This is also required to guarantee that certain new features will work.

However, if you don't plan on doing so, check the component compatibility beforehand.

As a best practice, plan your upgrade around scheduled maintenance windows. You must be able to restart Agent Groups / tiers should incompatibilities arise.

Split server instances

If in your current AppMon server installation path `<DT_HOME_OLD>` there exists a `server/instances` folder and you are using these **Server instances:** This feature and the `dtserver`'s and `dtfrontendserver`'s `-instance` parameter are deprecated as of AppMon 6.3. Collector instances are still supported.

Use this guide and check requirements there to migrate the Server instances to separate installations before upgrading.

Review custom-sized installations

Review your Server sizing by contacting support beforehand, if your `dtserver.ini` or `dfrontendserver.ini` file contains `-memory unsupported`.

Take notes

Document whether you are using **continuous transaction storage**. Check **Settings > Dynatrace Server > Storage** in the AppMon Client.

Document which additional plugins were installed by clicking **Settings > Dynatrace Server > Plugins** in the AppMon Client.

Document all Downtimes and incident rules of the Self-Monitoring System Profile.

Document the passwords to the following services: Email, LDAP, Proxy, DC-RUM, Gomez.

Performance Warehouse database

- Be prepared to create backups as part of this guide.
- Check in the System Requirements whether your SQL DB server is supported. If not, create a **backup** and switch to a supported DB before the upgrade.
- Verify that the configured **Performance Warehouse** user has the permissions to create and drop tables for the DBMS.

Session Store

- If the current storage location is within the installation directory (which is default) and you want to keep the stored sessions, move the session storage into a separate directory accessible from the both old and new installations before the migration. It will ease this and future migrations.
- Document the current configuration. In the AppMon Client, click **Settings > Dynatrace Server > Storage**.
- Be prepared to create backups as part of this guide.

License

If you use z/OS Agents, AppMon 6.3 and later no longer requires an additional license. Instead, contact license management to get the needed number of CICS and IMS agents and/or Java Agents added to your Server license.

Download files

Download the software required for the migration:

- **Installers** and ADKs for your application platforms and operating systems.

- The latest version of the **dynatrace-migration tool**
- The **latest Update**, if available, is located at the top of the 7.0 downloads list. See Install an Update.
- If you don't have internet connectivity from the Server, download the **upgraded licenses** from eServices.

Running two versions

You can run parallel versions of AppMon to test your upgrade, but be aware of the requirements for this configuration.

Next: Upgrade Server, Collectors, and Clients

Component compatibility

Looking for the 6.5 version of this page? Open in 6.5 documentation

Migration steps

Prepare to upgrade

Upgrade server, collectors, and clients

Upgrade agents

Final migration checks

Although certain version combinations may be compatible, we **strongly advise** all users to upgrade all components as soon as possible to the same version as the server.

Compatibility and Support

Components are said to be **compatible**, if they can connect and offer any kind of functionality together. For a **specific feature** to work, the support of **all involved components** (Agents, Collectors, Server, Client) is required.

Example 1: Infrastructure: Process Availability

When using AppMon 6.2 Host Agents with a 6.5 Server and Client, you will be able to configure Process Availability patterns, but these will have no effect and not fire incidents as the 6.2 Agent does not support them: This feature was only introduced starting with 6.5.

Agent-Collector Compatibility

Agents are upgraded depending on their type:

- **Bootstrapped Agents** update automatically when the instrumented application is restarted.

- Upgrade **Non-bootstrapped Agents** manually

To determine Agent versions and whether they are bootstrapped, use the AppMon Client's **Cockpits > Agent Overview** Dashboard.

Collector:	6.0	6.1	6.2	6.3	6.5	7.0
Agent						
5.6	OK	OK	OK	OK	OK	OK
6.0	OK	OK	OK	OK	OK	OK
6.1	n/a [1)]	OK	OK	OK	OK	OK
6.2	n/a [1)]	n/a [1)]	OK	OK	OK	OK
6.3	n/a [1)]	n/a [1)]	n/a [1)]	OK	OK	OK
6.5	n/a [1)]	n/a [1)]	n/a [1)]	n/a [1)]	OK	OK
7.0	n/a [1)]	n/a [1)]	n/a [1)]	n/a [1)]	n/a [1)]	OK

[1)] Upgrade the Server and Collectors before non-bootstrapped Agents.

Collector-Server Compatibility

Collectors are upgraded manually like Servers.

Collector:	6.0	6.1	6.2	6.3	6.5	7.0
Server						
7.0	OK	OK	OK	OK	OK	OK

Client-Server Compatibility

The major and minor version of the client has to match that of the Server. Lower-version Clients can auto-upgrade to the Server version if the major version matches.

Examples:

- 6.5.0 Client cannot connect to a 2017 May Server - requires a new installation
- 6.5.0 Client can connect to a 6.5.10 Server
- 6.3.0 Client when connecting to a 6.5.0 Server can auto-upgrade to 6.5.0.
- After such an auto-upgrade: a 6.5 Client cannot connect to a 6.3 Server

Running versions in parallel for migration

Multiple collectors on the same host

If you run more than one Collector version or instance on the same machine, verify that in their respective `collector.config.xml` files, on the `collectorconfig` XML element, they still have:

- unique ports (`agentport` attribute) assigned on which they listen for Agents
- unique `name` attributes
- `serveraddress` and `serverport` still indicate the correct Server information.

- Licensing allows for some flexibility allowing running versions in parallel (three-day grace period or three-week trial).
- Be sure to configure different ports for different version AppMon components to run in parallel and start / stop them at the right time.
- **Same major version *non-Webstart Clients*:** If you want to connect an old version Client to an old version Server again, answer **No** when it asks if it should update itself on connection to a new version Server.
- If you allow the update of the non-Webstart Client, it no longer connects to the older Server. To resolve this issue, use the Web Start Client to connect to the old Server or run / install a *same old version* Client.
- You cannot use the same Performance Warehouse database with two different Versions, thus you have to:
  - At some point disconnect the old version (only upgrades are possible) and connect the new version, or
  - Create a separate database with a different name, but then you will not have access to the data in the other database.
- **Do not** connect an older Server to the same Performance Warehouse DB once you connected the new version Server (and thus updated the DB).

Some components and connections in AppMon may result in potential pitfalls. To avoid some of these, consider:
In migration scenarios, having two versions of AppMon running at the same time is **not** supported. The standard migration procedure can be found here. The content of this page is meant as an aid for

experts.

Upgrade server, collectors, and clients

Looking for the 6.5 version of this page? Open in 6.5 documentation

On this page

1. Install the new Server, Collectors, and Clients
2. Deactivate and Upgrade License
3. Shut down the old version
4. Create backups
5. Services migration (Server, Collector and Memory Analysis Server)
6. Auto-start new services
7. Start the new Server components
8. Client migration
9. Start the new Client
10. Activate licenses for the new Server
11. Select server sizing
12. Apply update to new AppMon
13. Connect the Performance Warehouse
14. Configure session storage
15. Start new Collectors

Related pages

Dynatrace AppMon 2017 May Release Notes
Deployment Guide
Migration tool details

Downloads

dynatrace-migration tool

Product Downloads

Migration steps

Prepare to upgrade

Upgrade agents

Final migration checks

If you are migrating the Server to a new host without upgrading to a new version, you only need to perform the applicable steps.

1. Install the new Server, Collectors, and Clients

Install the new AppMon Server, AppMon Collectors and Clients into a **new directory**, referred to as `<DT_HOME_NEW>` in these instructions.

> **Important - Windows**
>
> At the end of the installation, you will be prompted to start the Server, Collector, and Client. **Do not start these components at this time**.

> **Important - Unix**
>
> Do **not** unzip the `.jar` installation files, but run `java -jar <FILENAME>.jar`.

See Silent Installation for automation.

2. Deactivate and Upgrade License

Deactivate and **upgrade** your AppMon license in eServices to not lose UEM volume.

3. Shut down the old version

1. **Stop** the old Client, Server, Frontend Server, Collector and Memory Analysis Server services.
2. **Disable** automatic startup of the old Server, Frontend Server, Collector and Memory Analysis Server:
   - Windows: Choose **Control Panel > Administrative Tools > Services**. In each service's properties, set the **Startup type** to **Manual**.
   - **Unix-like systems**: Use `chkconfig` or `update-rc.d` to disable all AppMon services.

4. Create backups

Using the configuration information collected during preparation:

1. Back up your **Performance Warehouse DB**.
2. Back up your **Session Store**.
3. **On *NIX**, back up all AppMon scripts in `/etc/init.d`.

Files in `DT_HOME` are backed up during the next step.

5. Services migration (Server, Collector and Memory Analysis Server)

Download and use the current Migration Tool on all Server and Collector machines as follows.

For migration without backup archives and further options see Migration tool details or run `java -jar dynatrace-migration.jar`.

5.1. File collection

Use `dynatrace-migration` to create a migration archive. For example: `java -jar dynatrace-migration.jar -migration -sourceDTHome "<DT_HOME_OLD>" -targetArchiveDir "<ARCHIVE_DIR>"`

Where `<DT_HOME_OLD>` is the old AppMon installation path. A `<MIGRATION_ARCHIVE>` file named `<Server_name>_<creation_dateTime>.dtma` is created in `<ARCHIVE_DIR>`.

Keep the resulting file as permanent backup.

5.2. File migration

On the target host, as Administrator or configured user, migrate the data to `<DT_HOME_NEW>` using the following: `java -jar dynatrace-migration.jar -migration -sourceArchive "<ARCHIVE_DIR>/<MIGRATION_ARCHIVE>" -targetDTHome "<DT_HOME_NEW>"`

Backups of files are changed in `<DT_HOME_NEW>` and a migration log are created in a new directory in `<DT_HOME_NEW>/migration/backup/`.

5.3. Edit configuration files

dynatrace-migration lists files with configuration settings that may need to be migrated manually to the new release. Therefore, the applicable old files are copied with the suffix `.toBeMigrated`. These files are found in `<DT_HOME_NEW>`. **Do not reuse** the old files, because for AppMon 6.3 and later, changed boot bundles cause your Server to not start.

Here is a sample output:

```
There are some files left, which have to be migrated manually:

In <DT_HOME_NEW>:
 \dtanalysisserver.ini -> edit and integrate custom settings from old \dtanalysisserver.i
 \dtcollector.ini -> edit and integrate custom settings from old \dtcollector.ini.toBeMig
 \dtfrontendserver.ini -> edit and integrate custom settings from old \dtfrontendserver.i
 \dtserver.ini -> edit and integrate custom settings from old \dtserver.ini.toBeMigrated
 \server\selfmonitoring\dtselfmon.ini -> edit and integrate custom settings from old \ser

Do NOT re-use the old files, this may cause components to not start!
```

- `dtserver.ini`: If you are upgrading and using continuous transaction storage (see Prepare to Upgrade), do **not** migrate `-Dcom.dynatrace.diagnostics.memory.maxPurePathBufferSize=<X>` if present.
- `dtserver.ini`: If it contains an entry regarding SQLSERVER, follow this KB article.
- Migrate all `-Xmx`, `-Xms`, `-XX:PermSize`, and `-XX:MaxPermSize` entries.

See more at Set up the Server and Set up Collectors.

5.4. Register and Auto-start additional collector instances

Expand for information on instances

If you used Collector instances in the old installation:

- **Windows:** In the directory `<DT_HOME_NEW>`, for each Collector instance, using the same `<CollectorInstanceName>`s as in the old version, execute:

```
dtcollector -service install -instance <CollectorInstanceName>
```

- ***NIX:** In `/etc/init.d`, for each Collector instance, there is an old startup script named `dynaTraceCollector<NN>`, where `<NN>` is a number. For each such file:
  - Take note of the values for the variables `DT_OPTARGS` and `DT_INSTANCE`.
  - Replace it with the new version: `cp <DT_HOME_NEW>/init.d/dynaTraceCollector/etc/init.d/dynaTraceCollect`
  - Edit `/etc/init.d/dynaTraceCollector<NN>` and fill in the noted values for `DT_OPTARGS` and `DT_INSTANCE`.

6. Auto-start new services

***NIX**: With `chkconfig` or `update-rc.d`, make sure that the new Server, Collectors, Memory Analysis Server, and Frontend Server are started automatically with `/etc/init.d` shell scripts and that the permissions are correct.

7. Start the new Server components

If you configured a `DT_HOME` environment variable, update it to `<DT_HOME_NEW>`, and then start the AppMon Server and Frontend Server.

8. Client migration

If it is acceptable for your users to reconfigure their client themselves (proxy settings etc), skip this step.

Otherwise, compare old and new files and carry over your changes:

- After a first client start in each user home the old and new `dtclient.ini`
- Defaults for new users on a machine: `<DT_HOME_OLD>\dtclient.ini.template` and `<DT_HOME_NEW>\dtclient.ini.template`

Where is the Dynatrace user home?

- **Windows**: `C:\Users\<username>\.dynaTrace\Dynatrace 7.0\`
- ***NIX:** `~/.dynaTrace/Dynatrace 7.0/`

9. Start the new Client

In `<DT_HOME_NEW>` start `dtclient` or use the Webstart Client at `https://<servername>:8021`.

Be sure to connect to the right AppMon Server (hostname) in the **Settings > Dynatrace Server > Connectivity** pane.

10. Activate licenses for the new Server

If you require a proxy server to access the Internet, manually configure the Client proxy settings to access the online licensing eServices site.

To import the Server and UEM licenses, click **Settings > Dynatrace Server > License > Import** and follow the instructions.

11. Select server sizing

After the license import, you are presented with different sizing options (see Sizing Settings for more details). Select the Sizing that you identified during the Preparations. It is important to perform this step **before starting the new Collectors**, to avoid Out-of-Memory problems.

12. Apply update to new AppMon

Check the download page if there is an update available for AppMon 7.0, use the Client to install the Update and follow the instructions. This restarts and updates Collectors and Agents.

13. Connect the Performance Warehouse

In **Settings > Dynatrace Server > Performance Warehouse**, verify configuration for the old database, click **Test** and **Connect**.

If you encounter an error, please see Performance warehouse migration troubleshooting.

14. Configure session storage

If you want to migrate your stored sessions, choose **Settings > Dynatrace Server > Storage** and verify that the directory points to your session storage data.

15. Start new Collectors

Start the new Collectors, then choose **Settings > Dynatrace Server > Collectors** and verify that all Collectors are connected to the Server. If you applied an update(to the new Server), you should immediately restart the Collectors from this dialog box for the update to be applied.

The AppMon Server and Collector now accept Agent connections.

Next: Upgrade Agents

Upgrade agents

Looking for the 6.5 version of this page? Open in 6.5 documentation

On this page

Related pages

Component compatibility

Upgrading zOS Agents

Set up agents

Agent Development Kit (ADK)

Migration steps

Prepare to upgrade

Upgrade server, collectors, and clients

Final migration checks

Upgrading the Agents is **recommended** and required to receive updates. Restarting an agent here means usually to restart **the instrumented application**.

Old Agents are compatible and continue to function.

1. Bootstrapped Agents

Includes: Java, .NET, Host, Native ADK, PHP, CICS, and IMS

Make sure the Agents are **connected**, then **restart** to upgrade them automatically. They continue to use the `<DT_HOME_OLD>` directory for retrieving their configuration. You can switch them to `<DT_HOME_NEW>` by upgrading the Bootstrap Agents (see below).

1.1. Bootstrapped Web Server Agents

Includes: Web Server

In the correct order:

1. Restart the Web Server Agent **Service** and wait until it has reconnected.
2. Restart the Web Server Agent.

2. Non-bootstrapped Agents

Includes: Mobile App ADK

Switch your Mobile App to the new ADK and deploy the new version to upgrade. The compatibility statement also applies to Mobile ADK Agents.

3. Bootstrap Agents

For AppMon 6.5 and later, **Bootstrap** Agents with versions 5.6 and earlier are unsupported. Verify their version and if unsupported, upgrade the Agent's bootstrap component.

These features also require minimum versions:

- Restart Webserver Agent Services from the Client (Bootstrap >= 6.2)
- Disable automatic agent updates (Bootstrap >= 6.3)

4. z/OS Agents

z/OS Agents can be bootstrapped or non-bootstrapped. See Upgrading zOS Agents for more information.

Next: Final Migration Checks

Upgrading Bootstrap Agents

Looking for the 6.5 version of this page? Open in 6.5 documentation

On this page

Preparation

Bootstrap agent upgrade process

Uninstall the outdated agents

Related pages

License upgrade

zOS Agent troubleshooting

zDC

Migration steps

Prepare to upgrade

Upgrade server, collectors, and clients

Upgrade agents

Final migration checks

AppMon 6.5 and later no longer supports bootstrap agents earlier than 6.0. This page describes how to upgrade your pre-6.0 bootstrap agents to the newest version.

Preparation

1. Download a new agent installer for your platform from downloads.dynatrace.com.
2. Consider a backup of your current agent configuration directory.
3. Deploy and install it on your target platforms next your agents installation which is currently in use.

Bootstrap agent upgrade process

- Host Agent
  1. Stop the old host agent.
  2. Migrate the content in `/conf/dthostagent.ini` (mainly the name and the server) from your old to your new installation directory.
  3. Start the new host agent.
- Webserver Agent
  1. Stop the old web server agent.
  2. Migrate your webserver agent settings in `conf/dtwsagent.ini` from your old to your new installation directory.
  3. Start the new web server agent.
- Java Agent
  1. Reconfigure the path to the agent in you local application configuration (for example, from `-agentpath:/opt/dynatrace-<loldversion>/agent/lib64/libdtagent.so=name=Application_Monitoring` to `-agentpath:/opt/dynatrace-<latestver>/agent/lib64/libdtagent.so=Application_Monitoring`).

     You can also install to and configure a version-independent agent path (for example, /opt/dynatrace-agent/) and move the old version out of that location and install the new version into the same directory.
  2. Restart your application.
- .NET Agent
  - Restart your application after installing the new agents.
- zOS zLocal / zRemote agent
  - Upgrading the zLocal / zRemote agents will automatically upgrade the bootstrap agents. Please follow the instructions here.

Uninstall the outdated agents

- **Linux**: Delete the installation directory of the outdated agents.
- **Windows**: Uninstall outdated agents using Add / Remove programs.

Upgrading zOS Agents

Looking for the 6.5 version of this page? Open in 6.5 documentation

On this page

Requirements

Upgrade procedure

Related pages

License upgrade

zOS Agent troubleshooting

zDC

Migration steps

Prepare to upgrade

Upgrade server, collectors, and clients

Upgrade agents

Final migration checks

Requirements

Upgrade AppMon Servers, Collectors, Agents and Performance Warehouse

Before starting any migration action for your z/OS Agents, read the general Upgrade to 7.0 guide and create a detailed plan for the upgrade of your AppMon Servers, Collectors, other Agents, and the Performance Warehouse and other components. For less maintenance overhead coordinate these upgrades to coincide with application restarts.

Important

The z/OS Agent can't directly connect to an AppMon Collector without first going through a zRemote Agent. Using the zRemote Agent reduces CPU consumption on z/OS. If your current pre-6.2 z/OS Agent deployment omitted the zRemote Agent, you must install it as a part of your upgrade. See zRemote Agent for more information.

License

You need valid licenses for the AppMon Server. See License Upgrade for more information. Although z/OS Agents are licensed MSU-based, the AppMon Server enforces the number of current connected CICS and IMS regions. See Prepare to Upgrade and License Upgrade for more information.

Upgrade procedure

1. Upgrade the AppMon Server and Collector as described in Upgrade to version 7.0 prior to upgrading the mainframe components. While newer zOS Agents are generally not compatible with older AppMon Collectors or Servers, newer AppMon Collectors or Servers usually work with older zOS Agents.

2. Follow the standard Agent installation process. This consists of running the JCL to download the agent distribution file and running SMP/E to populate the target datasets for the new release. For more information, see Download and SMP/E install of the z/OS Agent Components.

3. Do one of the following:

- Stop your current zDC and update the zDC authorized libraries to the version 7.0. Before you restart the z/DC, you should ensure that the zDC parm `DTMSG_SMOSIZE` has been uncommented and that parm `DTMSG_QSIZE` has been deleted. These parameters control whether messages are queued in the zDC dataspace or in the zDC 64-bit shared memory object (SMO) pending their writing to the AppMon Agent.
  The use of dataspaces has been deprecated. If you customize the ZDCSYSIN sample contained in the 5.6 version, you can skip this step because the sample already incorporates this change. After the zDC has been started, the CICS and IMS agents should register within 5 minutes. **- OR -**
- Create a parallel 7.0 environment for the zDC and zLocal (formerly USS) Agent. Follow the directions as described in Installing the zDC. With a parallel 7.0 environment, you can continue running the old z/DC while you migrate individual CICS regions and IMS control regions over to the new zDC, or you can test with a few regions prior to upgrading the default zDC. If multiple zDCs are running, the one that specifies DEFAULT(YES) will be chosen unless an INITPARM parameter in the CICS or IMS system initialization parameters specifies that it must connect to a zDC with a particular name: INITPARM=(ZDTPLT6x='xxxx').

Notes

- There can be only one zDC per LPAR with the DEFAULT(YES) option.
- The SUBSYSTEM_ID must be unique to each zDC.
- The library used in the zDC STEPLIB must be authorized.

4. The bootstrap Agent automatically upgrades the zLocal Agent when it first connects to an upgraded AppMon Collector / Server. If you use the non-bootstrap version of the z/OS UNIX Agent, you must run COPYAGNT to copy the new version to the location in the z/OS UNIX filesystem where it executes. See the COPYAGNT step in Installing the zDC.

5. Because the AppMon new version zDC is compatible with older CICS and IMS Agents, you don't need to upgrade all CICS regions and IMS control regions at once. However, if you monitor transactions cross CICS regions or IMS control regions, those related regions should be at the same Agent version otherwise incomplete PurePaths will result. Consequently, upgrading CICS regions and IMS control regions should be done with consideration to application groupings. Plan on upgrading all of your CICS regions and IMS control regions as soon as possible to take advantage of the features that come with the new AppMon version.

- To upgrade your CICS regions, replace the DFHRPL concatenation of the old PDS (or its contents) with the new PDS (or its contents). For example: `// DD DISP=SHR,DSN=<hlq>.LZDT700.SZDTLOAD`
- To upgrade your IMS control regions, replace the STEPLIB PDS (or its contents) in the JCL that executes the IMS Agent injection program, ZDTIINST, with the new PDS (or its contents). `// DD DISP=SHR,DSN=<hlq>.LZDT700.SZDTAUTH`

> **Note**
>
> The library used in the STEPLIB must be authorized.

Please refer to Migration Troubleshooting for the AppMon component compatibility table.

Final migration checks

Looking for the 6.5 version of this page? Open in 6.5 documentation

On this page

1. Manual configuration migration

2. Confirm operation of the new AppMon Server

Related pages

Troubleshooting migration

Migration steps

Prepare to upgrade

Upgrade server, collectors, and clients

Upgrade agents

1. Manual configuration migration

1.1. Reapply modifications to built-in items: sensor packs

Modifications to out-of-the-box sensor packs in the **Server Settings > Sensor Packs > Edit** dialog (only possible in debug mode) are not migrated. Do not export and reimport these old sensor packs, but rather reapply the edits in the UI.

Check user plugins

In the AppMon Client: **Settings > Dynatrace Server**, if plugins installed by the user were not migrated, re-install them.

1.2. Reimport LDAP certificates

If an additional certificate for LDAP (e.g. CA or self-signed certificate) was imported to the AppMon keystore:

- `server/conf/jssecacerts`

don't copy the old keystores, to avoid security issues through potentially compromised public certificates.

Please re-import the previous certificate by logging on with an AppMon local user and click in the AppMon Client: **Settings > Dynatrace Server > Users > LDAP > Test Connection.**

2. Confirm operation of the new AppMon Server

To ensure your new Server operates correctly, verify the following:

- All Agents displayed in **Cockpits > Status Overview > Agent Overview**:
  - Are connected
  - Have data coming in: scroll over horizontally to verify that **Event Count** or **Class Load Count** (or both) change
  - Have the correct Agent version (7.0) and Update version if one was installed.
- PurePaths come in.
- Dashboards are properly displayed.
- Historical data is present if you migrated the Session Storage.
- Verify that the WebUI (default port 9911), Server website (default port 8020) and REST services are reachable and have correct certificates if those were customized.
- Verify that these services and connector work: Email, LDAP, Proxy, DC-RUM, Gomez
- Review the **Start Center > Administration** for errors or warnings

Congratulations!

Migration successful!

Frequently Asked Questions

Forum posts

Open Q&A

Related pages

Troubleshooting migration

Migration steps

Prepare to upgrade

Upgrade server, collectors, and clients

Upgrade agents

Final migration checks

Which agent version runs after the Server/Collector upgrade?

After the Server/Collector upgrade, the Agent bootstrap upgrades the actual Agent, but the old version Agents run until you restart your applications.

Due to a change of the license handling of .NET and IIS agents in AppMon 6.3 and later, it is strongly recommended to restart all .NET and IIS agents on a host after an upgrade from a version earlier than 6.3 to AppMon 6.3 or later.

Please have a restart plan/coordination for your application tiers.

You can find the Agent version (bootstrap and active part) information in the **Agents Overview** dashboard, accessed from the AppMon Client cockpit.

Will old Agents work with the new license version of the Server?

Yes. For example, if you have a 7.0 Server and 7.0 license active on that server, 6.5 Agents consume the license slots as with a 6.5 Server and license. However, be sure to check the general Agent compatibility.

Do I need to install new Agents?

No for most Agents, because they are bootstrapped and upgrade automatically with a restart. But check if your bootstrap version is still supported. See Upgrade Agents for more information.

Do I need to create a Migration archive (.dtma)?

No, but it is highly recommended for backup purposes. See the tool syntax for more information.

Is the Performance Warehouse database schema backward compatible?

No. Please use your database backup to go back.

What version is my Performance Warehouse database schema?

Connecting the new AppMon Server to the old Performance Warehouse database updates the AppMon schema. You can check the version of the DB database.

For example, in SQL Server:

```
SELECT MAX(version) FROM schema_version;
```

If there is no table `schema_version` in your schema (in AppMon 6.3 and earlier), use the following statement:

```
SELECT * FROM dynatrace70.dbo.version;
```

Is the data between Server start and Performance Warehouse connection + end of migration lost (after upgrade)?

Yes, but only the last 10 minutes. Older data is lost.

How much time does the Performance Warehouse migration take during an upgrade?

This depends on the version you are upgrading from and the amount and type of data stored in the database. In general it should be done in a few minutes, but can take up to one hour in worst case scenarios.

Is it possible to migrate old Performance Warehouse data to a new database vendor?

No, if you require this, please submit a request for enhancement.

Is there any effect while the Performance Warehouse migration is running?

No, except only the last 10 minutes of data collected while the migration is running is stored.

In case of a rollback, how do I restore the Performance Warehouse?

Be sure that AppMon is disconnected while restoration. For the restoration, refer to your datyabase vendor documentation on how to restore snapshots.

Will deactivating the license cause an outage?

Not immediately for regular licenses (ecxludes demo and POC). After deactivation, you have a grace period where the license still functions. See license deactivation for more information.

UEM volume deactivates immediately however and does not have a grace period, as the voucher with the remaining visit counts is already returned to eService.

If I need to roll back, how can I get an old license as the original was upgraded?

Keep in mind that roll back should be a last resort scenario, first open a support ticket for your upgrade problem. Then open a license support ticket to get an old license.

How do I handle multiple Collector instances?

Collector instances share one single installation directory and are started and installed with the `-instance` parameter.

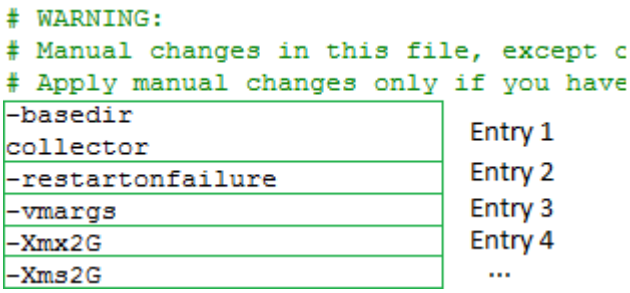
As these instances share certain files such as `dtcollector.ini`, you should shut down all instances when migrating files. The migration tool migrates all files for all collector instances by default, so you don’t need any special parameters. But you must make sure all instances are started from the new location using `init.d` scripts or installing new Windows services (as described in the guide).

How do I handle `.ini` and `.toBeMigrated` files

Using the file `dtcollector.ini` as an example, migration creates two files in your new installation directory:

- `<DT_HOME_NEW>/dtcollector.ini`: This is the new file from the installer for the version 7.0. It is used by the new collector. Any settings from your old installation are included in this file.
- `<DT_HOME_NEW>/dtcollector.ini.toBeMigrated` is a copy of your old `<DT_HOME_OLD>/dtcollector.ini` and stored here by the migration tool for convenience. This file potentially contains settings that you want to keep and integrate into the new file.

You can use a text comparison tool to clearly see the differences. Each entry begin with a "-" and ends right before the next "-", entries can be one-line or two-line. For example:



For each entry that is in `dtcollector.ini.toBeMigrated` and **not** in `dtcollector.ini`, decide if you ignore it or add it to `dtcollector.ini`.

Entries that you should migrate:

- Any entries that you added yourself, either in the file or through the "Debug Options" (Debug flags), mostly they should begin with `-Dcom.dynatrace`.
- Memory configuration lines, which begin with `-Xmx`, `-Xms`, `-XX:PermSize`, `-XX:MaxPermSize`.
- Memory and sizing entries: **-memory VALUE** and **-sizing VALUE**.

Troubleshooting migration

Looking for the 6.5 version of this page? Open in 6.5 documentation

Forum posts

Open Q&A

Related pages

Keystore Migration Troubleshooting

Migration steps

Prepare to upgrade

Upgrade server, collectors, and clients

Upgrade agents

Final migration checks

The migration archive is too large, migration is taking too long.

Case 1: Collectors: Large class cache

The class cache is found in `DT_HOME/collector/cache`. You can use the `-noclasscache` parameter, but keep in mind these remarks

Case 2: Many agentres files

Each update will install one in `DT_HOME/server/lib` and `DT_HOME/collector/lib`. These contain old agent versions that are required if agents of that version connect to the Server or Collectors. Some of those can be removed, but be sure to **carefully** follow these steps to not affect the monitored application:

1. Reduce the number of different Agents **versions** that connect to this installation. In the best case, update to the latest update and roll this out to Server, all Collectors and all Agents.
2. Make a list of all versions that any of your Agents are running on.
3. Shut down all components installed in this `DT_HOME` directory.
4. Do **not** delete `agentres.jar` (without version) or the subdirectory `minified`.
5. Back up and delete any files named `DT_HOME/server/lib/agentres-X.jar` and `DT_HOME/collector/lib/agentres-X.jar` for all versions X that are **not** in use on any of your Agents.
6. Continue with migration guide.

Collector or Server not starting?

See Keystore Migration Troubleshooting

Performance warehouse migration not successful?

See Performance warehouse migration troubleshooting.

PurePaths are missing after upgrade.

If you upgraded without migrating the class cache and are missing PurePaths or have PurePaths that contains `<unknown>` nodes, try to restart the instrumented applications. **Two** restarts of the instrumented applications are required to avoid this problem.

Memory settings in dtserver.ini and dtfrontendserver.ini are ignored and even removed

For the **server** '`-Xmx`' and '`-Xms`' VM arguments are ignored. Please select an appropriate sizing in **Settings > Dynatrace Server > Settings pane > Sizing**. See Sizing Settings for more details.

AppMon support

If you need further assistance, go to our support page.

Collecting support information

For issues with the upgrade or migration process and the migration tool, please provide the following information on support tickets

- A *support archive* with all logfiles if the affected components could be started and did connect - or a zip of the log folder of components that have problems
- The full migration archive (.dtma) if any was created. If no explicit location was specified, it is found it the target DT_HOME/migration/ directory. It can become quite large, if necessary we will provide an FTP location for upload
- All dtmigration.log *files* (source and target) from where the migration tool was executed (working directory)
- The versions of the source and target Dynatrace installation (eg. 6.2 -> 6.5)
- The version of the Migration Guide used (eg 6.3 or 5.6)

Performance warehouse migration troubleshooting

Error Message	DBMS	Affected version	Descriptio
SEVERE [FlywayRepositorySchema] error during performance warehouse migration: com.dynatrace.diagnostics.repository.flyway.FlywayRepositorySchema a:314 org.flywaydb.core.api.FlywayException: Found non-empty schema “DT_APM “ without metadata table! Use baseline() or set baselineOnMigrate to true to initialize the metadata table.	DB2 only	6.3 and later	For DB2, must have the name blanks in to fill up t name car correctly.
SEVERE [AbstractRepositoryAccessor] [PWH SCHEMA UPDATE] Migration V6.3.0.7__createIncidentProperties.ddl failed Error Code : -613 Message : The primary key, unique key, or table partitioning key identified by “INCIDENTPROPERTIES_PKEY” is too long or has too many columns.. SQLCODE=-613, SQLSTATE=54008, DRIVER=4.14.122 Location : ddls/db2/update/6.3.x/V6.3.0.7__createIncidentProperties.ddl (/ddls/db2/update/6.3.x/V6.3.0.7__createIncidentProperties.ddl) Line : 1 Statement : CREATE TABLE incidentproperties (uuid_lsb BIGINT NOT NULL, uuid_msb BIGINT NOT NULL, propkey VARCHAR(100) NOT NULL, value VARCHAR(1000) NOT NULL, CONSTRAINT incidentproperties_pkey PRIMARY KEY (uuid_lsb, uuid_msb, propkey, value))	DB2 only	6.3 and later	The Perfo requires a Performa administra page size since App message unique ke key identi “INCIDEN is too lon columns” this issue
SEVERE [AbstractRepositoryAccessor] cannot create performance warehouse schema: com.dynatrace.diagnostics.sdk.logging.ThrottlingLogger a:150 org.flywaydb.core.api.FlywayException: Found non-empty schema [MS\dtuser] without metadata table! Use baseline() or set baselineOnMigrate to true to initialize the metadata table.	SQL Server only	6.3 and later	Some spe the SQL s usage of the user r name. Th the two co backslash causes pr JDBC driv

The following table shows known PWH migration issues and the solutions. Please have a look at the error messages in the client and in the repository log.

If the migration did not finish successfully or the connection status shows “migration needed”, as a first

step, please check the Performance warehouse administration page for the requirements and needed database permissions. There may be an error message in the client, when you try to connect the Performance Warehouse (depending on your type of issue), which contains good hints for further analysis. Also, the repository log contains very helpful information. See **System Information > Server > Support Information > Log Files**.

Keystore Migration Troubleshooting

Looking for the 6.5 version of this page? Open in 6.5 documentation

Related pages

Troubleshooting migration

Certificate management

Advanced features - certificates, private keys and keystore

Symptoms

- Collector or Server is not starting.
- Jetty for DTANG, web services, or WebUI is not starting.

Also, one of the Logfiles in:

- `DT_HOME/log/collector/NAME/Collector.X.Y.log`
- `DT_HOME/log/server/Server.X.Y.log`
- `DT_HOME/log/server/FrontendServer.X.Y.log`

- `DT_CLIENT_HOME/log/client/Client.X.Y.log`
  Has one or several of the following strings:
  - The keystore file 'XXXX' is invalid
  - BrokenKeystoreFileException
  - javax.crypto.BadPaddingException: pad block corrupted
  - The subsequent `java.lang.NullPointerException` and `java.security.UnrecoverableKeyException` may differ a little, depending on the component (server, frontend server, collector, client):
    - [AbstractLifeCycle] FAILED SslContextFactory@47252b0d(/home/dynatrace/dynatrace-7.0.0/server/conf/jetty.jks,/home/dynatrace/dynatrace-7.0.0/server/conf/jetty.jks): java.lang.NullPointerException: org.eclipse.jetty.util.log.JavaUtilLog warn:71 java.lang.NullPointerException
      at org.eclipse.jetty.util.ssl.SslContextFactory.getKeyManagers(SslContextFactory.java:911)
    - [AbstractLifeCycle] FAILED SslContextFactory@1190d87(/home/dynatrace/dynatrace-7.0.0/server/conf/jetty_fe.jks,/home/dynatrace/dynatrace-7.0.0/server/conf/jetty_fe.jks): java.security.UnrecoverableKeyException: Password must not be null: org.eclipse.jetty.util.log.JavaUtilLog warn:71 java.security.UnrecoverableKeyException: Password must not be null

Solution 1

1. Download new version of the migration tool.
2. Recreate migration archive and re-try.

Solution 2

1. Shutdown all components using the affected installation directory.
2. Backup the full `<DT_HOME>/conf` folder.
3. Execute steps that apply.

Troubleshooting dt_pwdstore

This troubleshooting applies to `dt_pwdstore` files only.

In this case 'XXXX' in the exception is one of the following:

- `dt_pwdstore.bks/salt/key`

- `java.lang.NullPointerException at org.eclipse.jetty.util.ssl.SslContextFactory.getKeyManagers(SslContext`
- `java.security.UnrecoverableKeyException: Password must not be null`

1. Delete all `dt_pwdstore.*` and `.dt_pwdstore.*.bkp` files (the default `dt_pwdstore` is extracted at next startup) if you have a normal deployment.
2. Extract the `dt_pwdstore.zip` if you have a customized `dt_pwdstore`.
3. If still failing, delete all keystores, including `*.bks`, `*.salt`, `*.key`, and `*.bkp` keystore files in the folder.

Data lost:

- Nothing.
- All re-initialized stock passwords. Restore these by re-extracting `dt_pwdstore.zip`, or copy all `dt_pwdstore.*` files from a working component.
- *Step 3 only*: See sections of the specific component there data is lost.

Troubleshooting Collector

This troubleshooting applies to Collector only.

In this case 'XXXX' in the exception = `dt_co_keystore.bks/salt`.

The migration tool may copy potentially corrupt files from older deployments and leave needed ones behind.

1. If `.dt_co_keystore.*.bkp` files are present, restore them by removing `'.'` from the prefix and `'.bkp'` from the suffix.
2. Start the component.
3. If still failing, delete `*.bks`, `*.salt`, `*.key`, `*.bkp`.

Data lost:

- Collector proxy password.
- Private key and certificate of the collector.
- Accepted certificates from the AppMon server.

Troubleshooting Client

This troubleshooting applies to the AppMon Client only.

In this case ‘XXXX’ in the exception = `dt_cl_keystore.bks/salt`

Client keystores may be corrupted.

1. Delete `dt_keystore.*`, `.dt_keystore.*.bkp`, `dt_cl_keystore.*`.
2. In case `.dt_cl_keystore.*.bkp` files are present, restore them by removing `.` from the prefix and `.bkp` from the suffix.
3. Start the component. If still failing, delete `*.bks`, `*.salt`, `*.key`, `*.bkp`.

Data lost:

- Collector proxy password.
- Private key and certificate of the Collector.
- Accepted certificates from the AppMon server.

Troubleshooting Frontend Server

This troubleshooting applies to Frontend Server only.

In this case ‘XXXX’ in the exception = `dt_fe_keystore.bks/salt`.

The migration tool may copy potentially corrupt files from older deployments and leave needed ones behind.

1. Delete `dt_fe_keystore.*` if still present.
2. If `.dt_fe_keystore.*.bkp` files are present, restore them by removing `'.'` from the prefix and `'.bkp'` from the suffix. Otherwise, simply start the component.
3. If it’s still failing and `.dt_keystore.*.bkp` files are present, delete `dt_keystore.bks` and `dt_keystore.salt`, restore the `.bkp` files and restart.
4. If still failing, delete `*.bks`, `*.salt`, `*.key`, `*.bkp`.

Data lost:

- Private key and certificate of the frontend server.
- Accepted certificates from the backend server.

Troubleshooting Backend Server

This troubleshooting applies to Backend Server only.

In this case ‘XXXX’ in the exception = `dt_keystore.bks/salt`.

The migration tool may copy potentially corrupt files from older deployments and leave needed ones behind.

1. Delete `dt_keystore.*` if still present.
2. If `.dt_keystore.*.bkp` files are present, restore them by removing `'.'` from the prefix and `'.bkp'` from the suffix. Otherwise, start the component.
3. If still failing, delete `*.bks`, `*.salt`, `*.key`, `*.bkp`.

Data lost:

- Passwords like Gomez integration, PWH integration, SMTP integration, BT Export, Analysis Stream etc.
- Private key and certificate of the backend server.
- Accepted certificates from the memory analysis server.

Troubleshooting Memory Analysis Server

This troubleshooting applies to Memory Analysis Server only!

In this case ‘XXXX’ in the exception = `dt_al_keystore.bks/salt`.

The migration tool may copy potentially corrupt files from older deployments and leave needed ones behind.

1. Delete `dt_al_keystore.*` if still present.
2. If `.dt_al_keystore.*.bkp` files are present, restore them by removing `'.'` from the prefix and `'.bkp'` from the suffix, Otherwise simply start the component.
3. If it’s still failing and `.dt_keystore.*.bkp` files are present, delete `dt_keystore.bks` and `dt_keystore.salt`, restore the `.bkp` files restart.
4. If still failing, delete `*.bks`, `*.salt`, `*.key`, `*.bkp`.

Data lost:

- Private key and certificate of the memory analysis server.

Migration Tool

Related pages

Migrating server instances to separate installations

Migration tool details

Migration tool syntax

These pages provide detailed instructions and examples for using the migration tool when using the Upgrade and Migration guide to perform an AppMon installation migration or upgrade.

Click an entry below or in the Related pages to learn more about using the migration tool.

Migration tool details
Migration tool syntax
Migrating server instances to separate installations

Migration tool details

On this page

Overview

Migration Tool requirements

Use the Migration Tool

Examples

Related pages

Migration tool syntax

Troubleshooting migration

Migrating server instances to separate installations

Running multiple server instances on one system

Server.config.xml changes for server Instance removal

Upgrade server, collectors, and clients

Upgrade and migration guide

Overview

The AppMon Migration Tool (`dynatrace-migration`) is a command line tool that can backup and migrate configuration data from an existing version to a new AppMon version.

You can download the latest version here. The download is an executable, so there is no need to extract the jar file.

See the AppMon 2017 May Upgrade and Migration Guide for full instructions on upgrade or migration. Click here for previous versions of the Upgrade and Migration Guide. These pages also contain details on Performance Warehouse and Session Store handling.

By their very nature the Performance Warehouse DB and the session store in the file system need a different strategy for backup / migration (database / file-based backup).

Migration Tool requirements

- Java 7 or later is recommended, but Java 6x is also supported. AppMon includes a Java version in `<DT_HOME>/jre` that can be used if Java is not otherwise available.
- All source or target installation components such as the Server and Client must be shut down before running the Migration Tool. This is necessary because files like the Collector class cache are constantly changing, and files may also change while the tool runs. This can result in corrupt files.

Use the Migration Tool

Call the Migration Tool without command line parameters for a list and descriptions of the parameters you can use.

Examples

Here are some usage examples beyond the basic one or two-step services migration as described in the first two service migration steps in Upgrade server, collectors, and clients section.

Upgrade and migrate in one command

You can migrate using a single command if you can access the file systems of both old and new installations at the same time. Specify target locations for `sourceDTHome` and `targetDTHome`. This example migrates an AppMon full package installation on Windows:

```
java -jar dynatrace-migration.jar -migration -sourceDTHome "C:\Program Files\dynaTrace\Dynatrace 6.5" -targetDTHome "C:\Program Files\dynaTrace\Dynatrace 7.0"
```

Creating a backup archive

See AppMon Backup for how to use the migration tool to backup and restore critical files.

Upgrading with old Agent versions

If you have a need to keep old agent versions, it is possible during upgrade to keep the same agent version across application restarts. This command requires a target version of AppMon 6.3 or later, as well as bootstrap Agents 6.3 and later. The new bootstrap Agents must be connected to the old version Collectors before starting the Server upgrade.

```
java -jar dynatrace-migration.jar -migration sourceArchive "C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program Files\dynaTrace\Dynatrace 6.3" -upgradeAgents no
```

Agentres modification

Upgrading from AppMon 6.2 and earlier to AppMon 6.3 and later modifies the current version's `agentres.jar` to include the JavaScript Agent. This lets existing Webserver Agents inject the exact matching version. If the exact version match isn't found, the next higher version is injected. This is only relevant if there are Agents that have not been upgraded to AppMon 6.3 or later.

Use the `-noAgentResModification` flag to disable this modification, for instance:

```
java -jar dynatrace-migration.jar -migration -sourceDTHome "C:\Program Files\dynaTrace\dynaTrace 6.2" -noAgentResModification -targetDTHome "C:\Program Files\dynaTrace\Dynatrace 7.0
```

Upgrade and migrate without class cache

Data in the Collector class cache can be more than 10GB, which can take a significant amount of time and can pose a problem when transferring the file to the target location.

You can use the `-noclasscache` parameter to skip the class cache migration. However classes need to be newly discovered, therefore until the next Agent restart:

- HotSensor placement will not work.
- Class Browser will not show classes.
- Unknown PurePath nodes can happen with Agents earlier than 6.0.

Therefore the usage of this flag is *not recommended* and *by default* the Collector class caches are migrated.

This command migrates the configuration data from the archive, but not the class cache. `java -jar dynatrace-migration.jar -migration -sourceArchive "C:\migration\mymachine_<date>_<time>.dtma" -noclasscache -targetDTHome "C:\Program Files\dynaTrace\Dynatrace 7.0`

Select default and additional server and collector instances to migrate and upgrade migrate from archive

Upgrade and migrate all

This command example migrates and upgrades all default and additional AppMon Server and Collector instances from a specified archive. The command demonstrates the default behavior with for all parameters.

```
java -jar dynatrace-migration.jar -migration -sourceArchive "C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program Files\dynaTrace\Dynatrace 7.0"
```

Upgrade and migrate default server and collector instances only

```
java -jar dynatrace-migration.jar -migration -sourceArchive "C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program Files\dynaTrace\Dynatrace 7.0" -migrateInstances default
```

Upgrade and migrate default server instances

```
java -jar dynatrace-migration.jar -migration -sourceArchive "C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program Files\dynaTrace\Dynatrace 7.0" -migrateInstances defaultServer
```

Upgrade and migrate default collector instance

```
java -jar dynatrace-migration.jar -migration -sourceArchive "C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program Files\dynaTrace\Dynatrace 7.0" -migrateInstances defaultCollector
```

Upgrade and migrate additional instances only (not default server and collector)

```
java -jar dynatrace-migration.jar -migration -sourceArchive
"C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program
Files\dynaTrace\Dynatrace 7.0" -migrateInstances instances
```

Upgrade and migrate additional server instances (not default server)

```
java -jar dynatrace-migration.jar -migration -sourceArchive
"C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program
Files\dynaTrace\Dynatrace 7.0" -migrateInstances ServerInstances
```

Upgrade and migrate additional collector instances (not default collector)

```
java -jar dynatrace-migration.jar -migration -sourceArchive
"C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program
Files\dynaTrace\Dynatrace 7.0" -migrateInstances CollectorInstances
```

Upgrade and migrate, and specify alternative migration backup directory

```
java -jar dynatrace-migration.jar -migration -sourceArchive
"C:\migration\mymachine_<date>_<time>.dtma" -targetDTHome "C:\Program
Files\dynaTrace\Dynatrace 7.0" -backupDir "D:\migrationBackup"
```

Undo last migration step

```
java -jar dynatrace-migration.jar -undomigration -targetDTHome
"C:\Program Files\dynaTrace\Dynatrace 7.0"
```

Migration tool syntax

Related pages

Migration tool details

Troubleshooting migration

Upgrade and migration guide

```
java -jar dynatrace-migration.jar
--------------------------------------------------------------------------------
 Dynatrace Migration Tool       Copyright (C) 2004-2017 Dynatrace
--------------------------------------------------------------------------------
 Version 10.0.0.1258 built 2017-01-25T08:18:53.172-0500
  Startup 2017-03-30 14:40:19-0400

 -migration [{<option>}]
   Migrate, upgrade or backup an installation. Select an <option>:
     -sourceDTHome <directory> [-noClassCache]
            Dynatrace installation directory to migrate from.
            [-noClassCache]
            Instructs to not include the class cache (can require gigabytes).
     -sourceArchive <archivepath>
            Migration archive (.dtma) to migrate from.

    Select exactly one target option:
     [-targetArchiveDir <directory>]
            Target directory for the migration archive.
     [-targetArchive <filename>.dtma]
            Target path and filename for the migration archive.
     [-targetDTHome <directory> [-migrateInstances <instanceoption>]]
            Dynatrace installation to migrate to.

            -migrateInstances <instanceoption>
            Without -migrateInstances:
             All default and additional Server and Collector
             instances will be migrated.
            With <instanceoption>     will be migrated:
```

```
            with <instanceoption> ... will be migrated.
             default: only the default Server and Collector instance
             defaultServer: only the default Server instance
             defaultCollector: only the default Collector instance
             instances: only Server and Collector instances
             ServerInstances: only Server instances
             CollectorInstances: only Collector instances

            -makeServerInstanceDefault <instancename>
              EXPERIMENTAL Convert server instance to default instance
            -makeCollectorInstanceDefault <instancename>
              EXPERIMENTAL Convert collector instance to default instance
            -upgradeAgents [yes/no] Agents keep current version (target >= 6.3)
            -noAgentResModification Don't modify active agentres file

     [-backupDir <directory>]
            Optional: backup directory for files replaced during migration and
              to log migration steps and status
            If not present the backup directory is
                targetDTHome/migration/backup.

     -silent                 execute command without prompt.
     -skipDTHomeCheck        allow source or target directories that aren't Dyna
trace installations
     -skipRunningCheck       ignore running components
     -config <customConfig>  Use <customConfig> instead of default.
                             Mutually exclusive with [-migrateInstances <instanc
eoption>]]!
     -ignoreSpace            ignores required free space calculations
     -dropInvalidFilenames   drops files with invalid filesnames
     -ignoreNoReadAccess     ignore missing read access on sourceDTHome

 -undomigration [{<option>}]
   Where <option> includes:
     -targetDTHome <directory>
                             Dynatrace home directory for which you want to
                             revert the last migration step
     -silent                 execute command without prompt.

 -releaseNotes
 -version
```

Migrating server instances to separate installations

Related pages

Running multiple server instances on one system

Server.config.xml changes for server Instance removal

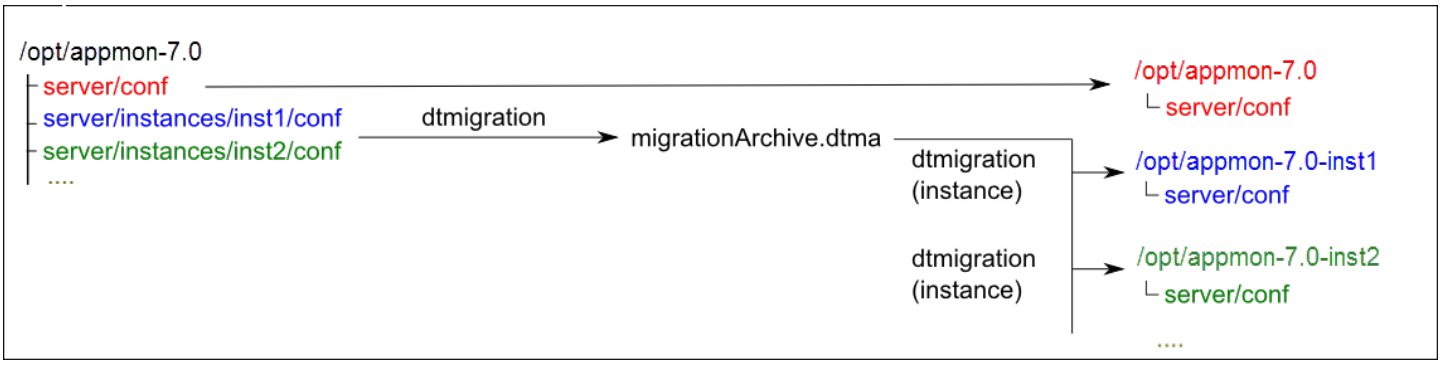
Upgrade server, collectors, and clients

Upgrade and migration guide

Overview

If you use AppMon Server instances that you created using the `dtserver -instance` command, you likely have two or more servers configured. The following steps let you to separate them into two separate installations, or distinct installation folders.

You should not perform an upgrade and separate the instances in a single step. First, split the instances into two distinct installations, for example `inst1` and `inst2`. Once you have split the installation instances, then upgrade each one separately. The following shows the workflow.



Windows requires a separate OS installation for each of the server installations, matching its sizing requirements. This can imply new hardware, virtual machines or Windows licenses. The `.msi` installer prohibits multiple installations on the same OS.

With separate server, differentiate between the following:

- The default server instance with most configuration files in `<DT_HOME>/server/conf`.

- The (zero to many) server instances with the most configuration files in `<DT_HOME>/server/instances/<INST_NAME>/conf`.

Perform the following before making any changes:

1. Download the latest Migration Tool.
2. Stop all components in all instances running from `<DT_HOME>`.
3. Back up all system before starting, especially the existing full installation directory using the following command:
   `java -jar dynatrace-migration.jar -migration -sourceDTHome "<DT_HOME>" -targetArchiveDir "<ARCHIVE_DIR>"`

Separation

Migrate the default server instance using the additional parameter `-migrateInstances default` when migrating from the migration archive to `<DT_HOME_NEW>` which is the target migration directory. See the AppMon 2017 May Upgrade and Migration Guide for more information.

For the remaining server instances, where `<DT_HOME_OLD>` is the original AppMon installation directory, for each AppMon Server instance, where `<INST_NAME>` denotes the instance name, perform:

1. Install a new AppMon Server using the installer on the target host.
2. Migrate all files, including files such as `.ini` files and agent resources shared between the default server instance and the server instance by executing the following command.
   `java -jar dynatrace-migration.jar -migration -sourceDTHome "<DT_HOME_OLD>" -targetDTHome "<DT_HOME_NEW>"` where `<DT_HOME_NEW>` is the selected installation folder, for example `/opt/dynatrace-7.0-instance1`.
3. Execute the following command to migrate the server instance files to the default server instance.
   `java -jar dynatrace-migration.jar -migration -sourceDTHome <DT_HOME_OLD>" -targetDTHome "<DT_HOME_NEW>" -migrateInstances serverinstances -makeServerInstanceDefault "<INST_NAME>"`.
4. Delete the `<DT_HOME_NEW>/server/instances` folder for cleanup and configuration verification.

Port offsets for the previous instance server remain in place.

Manual steps

1. All system profiles of the default instance and the migrated instance are migrated. From `<DT_HOME_NEW>/server/conf/profiles`, remove all `*.profile` files that are only in `<DT_HOME>/server/conf/profiles` and not also in `<DT_HOME>/server/instances/<INST_NAME>/conf/profiles`.
2. Dashboard files have server instance references included. Either open each dashboard in the AppMon Client and set correct data source, or replace the server instance name in XML files before starting the server.
3. For Linux only, update all `init.d` files. To do this, remove the variable values for `<DT_INSTANCE>` and `<DT_FE_INSTANCE>`. Keep the `-portoffset` parameter in the `<DT_OPTARGS>` and `<DT_FE_OPTARGS>` for new services (DTANG Adapter).

Apply general rules for multiple installations on the same system, for instance:

- different startup and shutdown scripts referencing the correct installation (init.d)
- separate TCP / UDP ports

Continue with the Upgrade and Migration Guide after the file migration for each instance.

Running multiple server instances on one system

Related pages

Server.config.xml changes for server Instance removal

Upgrade server, collectors, and clients

Upgrade and migration guide

Note

While running multiple server versions on one machine is possible, it is not recommended. Each server should run on a separate machine to provide optimum server performance and prevent port and other issues.

When different versions of AppMon are installed to the same machine, they have fully separate installation directories, for example `C:\Program Files\dynaTrace\Dynatrace 6.5` and `C:\Program Files\dynaTrace\Dynatrace 7.0`. Each installation instance must have different port assignments. However, because AppMon assigns ports during installation, the same default ports are potentially assigned for each of two separate installations on the same machine. One installation must use non-default port settings. AppMon writes the port settings during initial Server or Frontend Server startup. One of the installation instances must have different ports assigned.

To best resolve this potential issue, you must change the one of the installation instances to use different ports.

To do this, do the following to the second instance installed.

1. During installation, clear the **Start Server** option in the installer.
2. Do one of the following, depending on the version of AppMon being installed:

Then as a user with the necessary permissions, continue depending on the version in that installation directory, with an example offset of 100:

AppMon 6.3 and later* In the proper AppMon installation directory run the following:
`dtserver -portoffset 100`
`dtfrontendserver -portoffset 100`

AppMon versions prior to 6.3 The DTang port is best calculated (to have the same offset for all ports) for the server as `8040+offset` and for collector as `8042+offset`. So in `dtserver.ini` add the entries like this:

```… (other entries) …

-server
-Dcom.dynatrace.diagnostics.dtangport=8140 -Dcom.dynatrace.diagnostics.PortOffset=100

… (other entries) …
```

```
In `dtfrontendserver.ini` (with no DTang entry)

```... (other entries) ...

-server
-Dcom.dynatrace.diagnostics.PortOffset=100

... (other entries) ...
```
```

Server.config.xml changes for server Instance removal

Related pages

Running multiple server instances on one system

Upgrade server, collectors, and clients

Upgrade and migration guide

As part of Migrating Server Instances to Separate Installations, when moving the `<DT_HOME>/server/instances/INSTANCE_NAME/conf/server.config.xml` file to `<DT_HOME_NEW>/server/conf/server.config.xml`, the current migration tool makes the following changes. The changes here are only for reference and manual execution if necessary.

In `server.config.xml` : On the following XML-**elements**:

- `/dynatrace/serverconfig/collectorconfig/loggingconfig` the `path` attribute changes from `../../../log/collector/<INST_NAME>` to `../log/collector/dynaTrace Collector`.
- `/dynatrace/serverconfig/loggingconfig` the `path` attribute changes from `../../../log/server/<INST_NAME>` to `../log/server`
- `/dynatrace/serverconfig/settings/server` the attribute `instancename` is removed.
- `/dynatrace/serverconfig/settings/oopanalyzer` the `logpath` attribute changes from `../../../log/analysisserver` to `../log/analysisserver`.
- If the embedded repository is used: `dynatrace/repositoryconfig/repository/connection` the `embededdatapath` (sic!) attribute changes from `"instances/<INST_NAME>/repository"` to `"repository"`.