

Device Deduplication

Device Deduplication is a new feature available by installing two Accumulated Hotfixes using Tachyon Setup as described in [Tachyon Setup: Upload and apply a 1E Hotfix bundle](#). You can get these hotfixes from 1portal.force.com/s/tachyontopicdetail.

The hotfix for Tachyon Platform *Server* must be installed first, and hotfix for *Content Distribution* installed second. Installing each hotfix does not enable the feature until you have modified the Tachyon Coordinator configuration settings as described below. The *Server* hotfix resolves general duplication issues, and the *Content Distribution* hotfix resolves issues seen in the Nomad app.

 After applying the hotfixes, the Device Deduplication feature uses internal default settings, causing the process to run at 1AM UTC every Sunday, and only log duplicates found based on FQDN, without making any database changes.

The configuration steps described below for updating the Coordinator configuration file, specify False (default) for each of the configuration arguments. You are strongly advised that all configuration arguments - except for DeviceIdentifier - are set to False (default) for the first run or for whenever a new DeviceIdentifier value is being set. Results should then be analyzed to ensure that the DeviceIdentifier specified is correct for your environment.

On this page:

- [What is Device Deduplication](#)
- [Configuration](#)
 - [Modifying when Device Deduplication runs](#)
- [Arguments](#)
 - [Instance](#)
 - [Multiple Passes of Deduplication](#)
- [Experience Synchronization](#)
- [Logging and analysing results](#)

What is Device Deduplication

Several scenarios can lead to the same device being reported more than once within Tachyon, which can find their way into Experience and other applications. The main issue is duplicates having different TachyonGuid's. TachyonGuid is the primary distinct identifier of a device meaning that when calculating the total number of devices within the system it can easily show a count that is higher than the real world number. It also means that data associated with a device (for example, Experience performance data) will eventually be split amongst multiple TachyonGuids, affecting access to the device's data as well as affecting aggregations.

In most cases, deduplicating using the device FQDN should suffice, in that it is reasonable to assume that multiple devices with the same FQDN are the same device. However this is not true in all cases - for various reasons some organizations do not use FQDN to identify a device, they may uniquely identify devices based on Make, Model and Serial Number. For this reason the deduplication feature can be configured with one or more device attributes to uniquely identify a distinct device.

Configuration

Tachyon stores its Device Deduplication configuration in the `<INSTALLDIR>\Tachyon\Coordinator\Tachyon.Server.Coordinator.exe.config` file.

After any changes reboot the server or restart the **1E Tachyon Coordinator** service.

The time and frequency of when the Device Deduplication process runs is changed by adding and editing the **DeviceDeduplication** setting.

```
<module assemblyName="Tachyon.Server.Coordinator" providerName="Tachyon.Server.Coordinator.Scheduling.SchedulingModule">
  <settings>
    <add key="Crontab1" value="50 23 * * * LogDevicesSeenInTheLast7Days" />
    <add key="Crontab2" value="15 * * * * MaintainPolicyRulePartitioning" />
    <add key="Telemetry" value="40 23 * * 5 SendTelemetryStats" />
    <add key="DeviceDeduplication" value="0 1 * * 0 DeviceDeduplication {DeleteDevices:'false',
UpdateDeviceData:'false',DeleteOrphanedDeviceData:'false',DeviceIdentifier:['FQDN']}" />
  </settings>
</module>
```

The numbers that you see after **value** is a crontab schedule expression. The schedule `0 1 * * 0` means "at 01:00 hours, any day of the month, any month of the year, on weekday 0 (Sunday)". In other words, this configuration runs the Device Deduplication process at 1AM UTC every Sunday night.

You can use an online tool such as <https://crontab.guru/> to verify your crontab schedule expression.

 Please ensure there are no spaces in the **arguments** text (within the brackets).

Do not change the other crontab keys unless advised by 1E.

Modifying when Device Deduplication runs

By default, **Device Deduplication** runs at 1AM UTC every Sunday night. Another process called **Experience Synchronization** runs at 2AM UTC daily. It is very important that Experience Synchronization is not running when Device Deduplication starts, therefore if you change the time when either of these processes start, then you must ensure **Experience Synchronization** process starts **after** the **Device Deduplication** process start, and ideally there should be at least a 1 hour gap to avoid overlap. You may need a longer gap for very large systems.

Arguments

Running with all "Update..." and "Delete..." arguments set to false will make no database changes and is considered a diagnostic/practice run. Following a run such as this the DeviceDeduplication and DeviceDeduplicationLog tables could then be analyzed to figure out how many duplicates are in the environment based on the **DeviceIdentifier** input. This would provide a projected impact for a subsequent run with one or more of these arguments set to true. Alternatively, the user could then also choose to modify the **DeviceIdentifier** value to include more/less/different columns that would provide a more fine tuned target of what the user expects a truly unique device to be.

Argument	Default	Usage
Instance	Instance number of last run +1	This can optionally be added to the list of parameters in the app setting. By default, the orchestration will handle this automatically. If specifying, previous instance numbers can be found in the DeviceDeduplicationLog table. A use case for specifying this is retroactive execution, see Instance below section for more details.
DeleteDevices	False	Delete duplicate devices from both databases, and any management group associations for those devices.
UpdateDeviceData	False	Relink Tachyon Experience performance data from duplicate devices (that will be deleted) to the singular active device that will be kept.
DeleteOrphanedDeviceData	False	Remove Tachyon Experience performance data that is associated with a device that no longer exists.
DeviceIdentifier	['FQDN']	<p>This is an array where each item is wrapped in single quotes. Items are separated by a comma. When specifying multiple values within an item, a comma must also be used. See the Multiple Passes of Deduplication section below for more details of scenarios where more than one item is specified.</p> <p>Assuming default behavior of a single item: Values used can be any columns from the TachyonMaster.dbo.Device table. Any columns specified here will be used to create a single hash per device. Any devices that contain identical values across all columns specified will be assigned the same hash. Of these devices, the one that has the most recent LastConnUtc value will be marked as ToBeDeleted=0 whilst the others will be marked as ToBeDeleted=1. This is how we identify duplicates as well as the singular active device we want to keep.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p> If the column schema of the Device table changes between runs, then the DeviceDeduplication table will have to be backed up and deleted before the next run. This will generate a new version of the table in the next run with up-to-date device columns, which also means new columns can now be specified within the DeviceIdentifier value.</p> </div>

The following table describes the system behavior for every permutation of boolean arguments passed to DeviceDeduplication:

DeleteDevices	UpdateDeviceData	DeleteOrphanedDeviceData	Device Deduplication process behaviour
F	F	F	This is the default configuration (3 x False). Before you change these settings, you should run a diagnostic/practice run where no changes will be made to the devices or their data. Check DeviceDeduplication and DeviceDeduplicationLog table for results analysis.
T	F	F	Delete devices identified as duplicates, do not update or merge any existing performance data.
F	T	F	Do not delete devices identified as duplicates, reassign duplicate devices performance data to the single device that has been identified as the most recent instance of the device. *1
F	F	T	Do not delete devices identified as duplicates, do not update or merge any existing performance data. Delete any performance data that is associated with a TachyonGuid that no longer exists in the TachyonMaster Device table. This might be useful as a database cleanup exercise to remove "unparented data". *1 *2
T	T	F	Delete devices identified as duplicates and merge any existing performance data that was associated with duplicates.
F	T	T	Do not delete devices identified as duplicates, reassign duplicate devices performance data to the single device that has been identified as the most recent instance of the device. Finally, after relinking the data, delete any performance data that is associated with a TachyonGuid that no longer exists in the TachyonMaster Device table. *1
T	F	T	Delete devices identified as duplicates, do not update or merge any existing performance data. Delete any performance data that is associated with a TachyonGuid that no longer exists in the TachyonMaster Device table. *2
T	T	T	Delete devices identified as duplicates, reassign duplicate devices performance data to the single device that has been identified as the most recent instance of the device. Finally, after relinking the data, delete any performance data that is associated with a TachyonGuid that no longer exists in the TachyonMaster device table.

*1 - DeleteDevices: 'false' and either of the other settings is 'true' - This could lead to continued inflated device counts in the system.

*2 - UpdateDeviceData: 'false' and DeleteOrphanedDeviceData: 'true' - Be careful as this data will be deleted and cannot be recovered or merged at a later date even if the TachyonGuid that reported it comes online again.

Instance

Each run of Device Deduplication will result in entries in the **DeviceDeduplicationLog** table. All rows for the run will be assigned an instance number. The Instance number is also used to be able to share context between the TachyonMaster and TachyonExperience parts of the process.

Retroactive execution: An old instance can also be passed to the process via the app setting. When an old instance is provided duplicates will not be recalculated, instead the list of devices from that previous run will be used (from the **DeviceDeduplication** table). This will also be logged in the **DeviceDeduplicationLog** table, an entry that reads "This is a rerun of a previous instance. Previous history from the DeviceDeduplication table will be used for this instance". In an execution such as this any new value for **DeviceIdentifier** will be ignored however all other new arguments will be applied.

Multiple Passes of Deduplication

The device identifier argument is a comma separated list of items, each item itself being a comma separated list of device attributes. The order of items specified will be adhered to as the order of **passes** of Device Deduplication. This means that the first pass will calculate duplicate devices based on the arguments specified against the **Device** table. Every subsequent pass will further calculate duplicate devices based on its arguments specified against the **results of the previous pass**. There is no limit to the number of passes specified.

For example, the following passes would be run if the DeviceIdentifier is set to DeviceIdentifier: ['SerialNumber,SMBiosGuid','Manufacturer,Model,Location','FQDN',]:

Pass Number	Device attributes used	Identify duplicates based on...
1	SerialNumber SmbiosGuid	Devices in the TachyonMaster Device table that have the same SerialNumber AND SMBiosGuid.
2	Manufacturer Model Location	Devices identified as duplicates in pass 1 that also have the same Manufacturer, Model and Location.
3	FQDN	Devices identified as duplicates in pass 2 that also have the same FQDN.

The results of pass 3 alone are then used for the rest of the process and considered duplicates going forward.

Experience Synchronization

If any of the Boolean arguments are set to true, Device Deduplication may affect existing data in the system, whether through relinking to a new TachyonGuid or removing as orphaned data. For this reason, when the Device Deduplication process completes, it triggers a **full** Experience Synchronization. By default, the regular Experience Synchronization process is **incremental**, and runs at 2AM UTC daily.

The **full sync** includes a **full process** of the TachyonExperience SSAS cube, which is initiated regardless of the current ProcessMode option specified (GlobalSetting table in the TachyonExperience database). A **full process** is required because measures which typically Tachyon handles as additive can have historical data modified by the DeviceDeduplication process, which could otherwise lead to inaccurate counts and aggregations.

For large environments the **full process** of the cube is likely to be a long running background process, it will not impact data displayed in the UI until it has completed. If it conflicts with an ongoing Experience Synchronization process it will be skipped. Server resource demand is likely to increase during a cube process. Be aware that depending on the resources available, this could affect the query response time of the UI.

The latest Accumulated Hotfix for Tachyon Platform Server allows you to add the **ExperienceSync** setting to the Coordinator configuration file. You do not need to add or modify this unless advised by 1E.

```
<module assemblyName="Tachyon.Server.Coordinator" providerName="Tachyon.Server.Coordinator.Scheduling.SchedulingModule">
  <settings>
    <add key="Crontab1" value="50 23 * * * LogDevicesSeenInTheLast7Days" />
    <add key="Crontab2" value="15 * * * * MaintainPolicyRulePartitioning" />
    <add key="Telemetry" value="40 23 * * 5 SendTelemetryStats" />
    <add key="DeviceDeduplication" value="0 1 * * 0 DeviceDeduplication {DeleteDevices:'false', UpdateDeviceData:'false',DeleteOrphanedDeviceData:'false',DeviceIdentifier:['FQDN']}" />
    <add key="ExperienceSync" value="0 2 * * * ExperienceSynchronization" />
  </settings>
</module>
```

The numbers that you see after **value** is a crontab schedule expression. The schedule **0 2 * * *** means "at 02:00 hours, any day of the month, any month of the year, any day". In other words, this configuration runs the Experience Synchronization process at 2AM UTC daily.

You can use an online tool such as <https://crontab.guru/> to verify your crontab schedule expression.

Logging and analysing results

The Device Deduplication process creates logs in the **Tachyon.Coordinator.log** file.

The **DeviceDeduplicationLog** table - in the TachyonMaster database - logs more granular detail including the arguments used for the instance, details of each pass if more than one is specified, numbers of duplicates found, as well as rows affected in specific tables by various actions.

The **DeviceDeduplication** table - in the TachyonMaster database - stores a list of devices to be kept and their duplicates for the instance. The **ToBeDeleted** column is either 0 (false) for a device to be kept, and 1 (true) for a duplicate to be deleted. Kept and duplicate devices are associated by the **Hash** column.

With all arguments set to **false** you can review whether the specified **DeviceIdentifier** is able to uniquely identify devices and genuine duplicates. Once you are sure the **DeviceIdentifier** is suitable, you can then change other arguments from **false** to **true** according to your needs.

When the **DeleteDevices** argument is set to true, the **Device** table is updated for each device where older duplicates of it have been identified. Each device will have a timestamp set for **LastDeduplicationUtc** column, as well as an Experience event logged. The timestamp is updated when one or more new duplicates are found. There is also a **DeduplicationCount** column which represents the number of complete Device Deduplication processes where a device has been found to have duplicates. These values will only be set on the device identified as being the most recent device that will be kept, not its duplicates.

Devices with a **LastDeduplicationUtc** value can be found using the below SQL query.

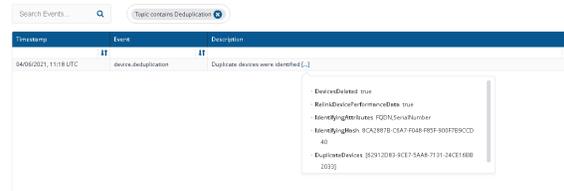
Genuine devices that have had duplicates identified

```
SELECT * FROM [TachyonMaster].[dbo].[Device] WHERE LastDeduplicationUtc IS NOT NULL and DeduplicationCount IS NOT NULL
```

The Experience event can be viewed within Tachyon Experience (Devices {Select a device} Logs).

The details of the log will show what attributes were used when it was identified as having duplicates, if multiple passes were specified they will be delimited with a '#' in the order of execution. The **IdentifyingHash** value can be used to query the **DeviceDeduplication** table **Hash** column, in the TachyonMaster database.

This will provide all device details of the device itself as well as all of its duplicate devices.



Timestamp	Name	Description
04/06/2021, 11:19 UTC	device-deduplication	Duplicate devices were identified [...] DeviceDeleted: true #IdentifyDevicesPerformedData: true IdentifyingHashes: F02N,Scss,Flamber IdentifyingHash: 8CA3897B-C5A7-F548-8F5F-80078ECCD -407 DuplicateDevices: 302912D83-9CE3-5A48-7131-24C214E8B 2033