Compere

Managing Assets with Compere

User Guide

THSENSE

Managing Assets with Compere : User Guide

Trademark Information

Copyright Information

Printed: January 2024

This edition is for software version Compere 1.0-7
Document ref.: M860-1

E: info@7thsense.one
W: 7thsense.one

7thSense Design Ltd

2 The Courtyard, Shoreham Road
Upper Beeding
Steyning
West Sussex
BN44 3TN
UK

T: +44 (0) 1903 812299

7thSense Design LLC

4207 Vineland Rd
Suite M1
Orlando, FL 32811
USA

T: +1 407 505 5200

Overview

In a Compere system using timelines, all timeline assets are first ingested into a database to become a pool of timeline resources. The database itself and the logistics mechanisms are, for the most part, installed behind the scenes and require no user intervention. The technical details of how it works may nevertheless help, especially when you want to change the way you store, distribute and manage groups of assets for different purposes. Maybe you want to move to network storage, or move to higher resolutions and section large media differently. If so go to:

- Asset Logistics App 16
- Configure Asset Logistics for Remote Servers 18

There is huge scope in the Compere Asset Database and it can adapt with you.

Ingesting

Anything that might be required on a timeline, such as video, audio, shaders, maps, Notch blocks, mesh models and so on, are not just copied from disk to disk in Compere, rather they are processed by a separate app called **MediaLogisticsApp**. The ingest process secures and adds value by identifying content to be added, extracting its metadata, and moving files to registered locations. In this way the assets become identifiable and accessible throughout the entire system.

Some of metadata is drawn from the asset files themselves, more can be added by user intervention.

The contents of the database are managed in the Compere Asset Management 9 panel.

Organising

The asset database records everything you need to know about assets and where they are located and distributed. It does not organise assets in the way you might want to see them together; that is left to Asset Management in Compere.

The **Asset Management** panel uses the available metadata for ingested assets to organise the *user view* of what is available. This includes virtual folders and subfolders, grouping, filtering and sorting as well as search. This is important, because once many assets have been ingested you want to make them easy to find and use, in the way you think about them for your application.

Assets are dragged directly out of the Asset Management panel onto timelines.

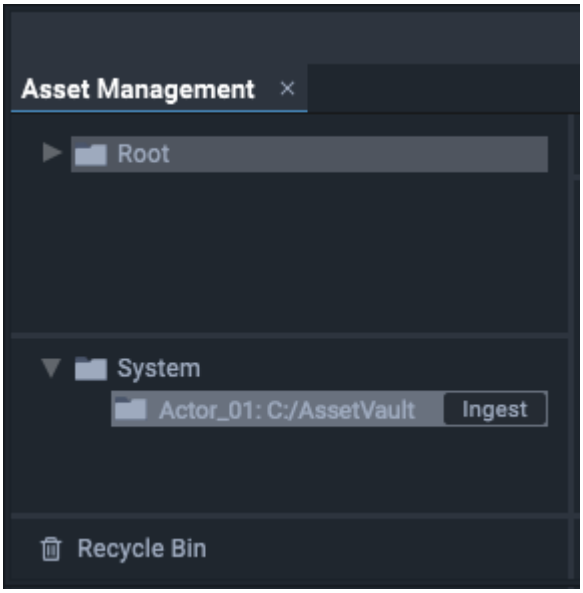
Ingest: Asset Logistics

Assets to be ingested into the database and distributed across the whole system, are placed into designated ingest ('watch') folders on the ingesting server. Assets can include:

- .7th sequences
- .7th images
- .wrp7 warp files
- .mp3/.wav/.ogg audio files
- Unreal Engine Generative files
- Notch blocks

The AssetLogistics app is configured by the ML-settings.xml file (see below).

Available localised vaults for distribution can be seen by the app and presented as 'System' folders in the Asset Management panel in Compere. Ingest can be in this way be manually pulled into specific local vaults.



Click the ingest button for the required vault destination. While ingest is taking place a downloading icon is shown in place of the button.

Once ingested, assets become available in the Root folder at the top. From here, assets can be organised, grouped and sorted.

Multiple Actors/Conjurers and the assets database

Clients read the distributed assets database; in this way they can see and locate each other's assets, for example, distributed sections of media. When a group of media sections is added to a project timeline, each Actor will play back its allocated section of that name.

Using a NAS

Network access storage can be used as a streaming source for Actor devices, but requires there to be one device running the Logistics app. This will provide the ingest process for the NAS, to monitor the ingest folders, and populate the database. NAS ingest ('watch') folders must not be monitored by more than one instance of the MediaLogisticsApp. When used as a streaming resource, there must be adequate bandwidth and speed for delivery, for example employing ST 2110 systems with high-speed switches and PTP.

Media distribution

Media can be ingested locally where required for best performance. The distributed database will always reveal to any device where all assets can be found.

Asset Logistics Settings

All the above is maintained in the Asset Logistics configuration file **ML-settings.xml** in C:\7thSense\Compere. Once set up for a system it will largely remain unchanged.

Here is an example settings file, where Sectioned Media [8] is distributed between a number of media servers:

```
<MediaLogisticsSettings
                        timebaseNumerator="60000"
                        timebaseDenominator="1001"
                        watchFolder="E:\Watch"
                        assetVault="E:\Vault"
                        assetVaultTags="local;section_4"
                        machineName="actor-04"
                        machineAddress="10.100.110.103"
                        databaseNodeAddresses="10.100.110.100;10.100.110.101;
                        10.100.110.102;10.100.110.104;10.100.110.105;
                        10.100.110.106;10.100.110.107;10.100.110.108;
                        10.100.110.109;10.100.110.110;10.100.110.111;
                        10.100.110.112;"/>
```

Here the watch and vault file locations are established, the source (local or a NAS, for example) and name of the required section set, details of this Actor, followed by a list of all **other** destination nodes (Actors, Conjurers) that participate in asset ingest and distribution activities (i.e. all devices with the distributed database except this one).

The vault tags are user metadata to be added on ingest. In this example, the vault to hold media section 4 (e.g. one carved part of a large movie sequence) is tagged as such. It has also been given a common tag ‘local’ shared by other assets, so that they can all be addressed by selecting the ‘local’ tag.

Sectioned Media Management

Very large canvases typically use sectioned (carved) media, so that the bandwidth per media server is optimised. In Actor, sectioned media needs to be ingested with section metadata, so that each set of sectioned frames retains identity and location (position) within the original media resolution.

A series of folders for the sections will typically have a common root name, as will the stem name of the frames they contain, but prior to ingest, each section-named subfolder also requires a **sidecar.json** file containing destination tags and positional properties.

```
{
  "metadata":{
    "sectionOriginX": "8192",
    "sectionOriginY": "2160"
  },
  "tags": [
    "section_4"
  ]
}
```

This will ensure that Asset Management can identify the **sections**, their respective distribution **locations**, and relative **positional** (origin) properties. Frame **size** is gathered on ingest from the native frame metadata.

The tag identifying the media section each Actor requires, is used by the server that is managing asset ingestion (see Configure Asset Logistics for Remote Servers [18]) to push the respective media sections to the designated section Actor servers.

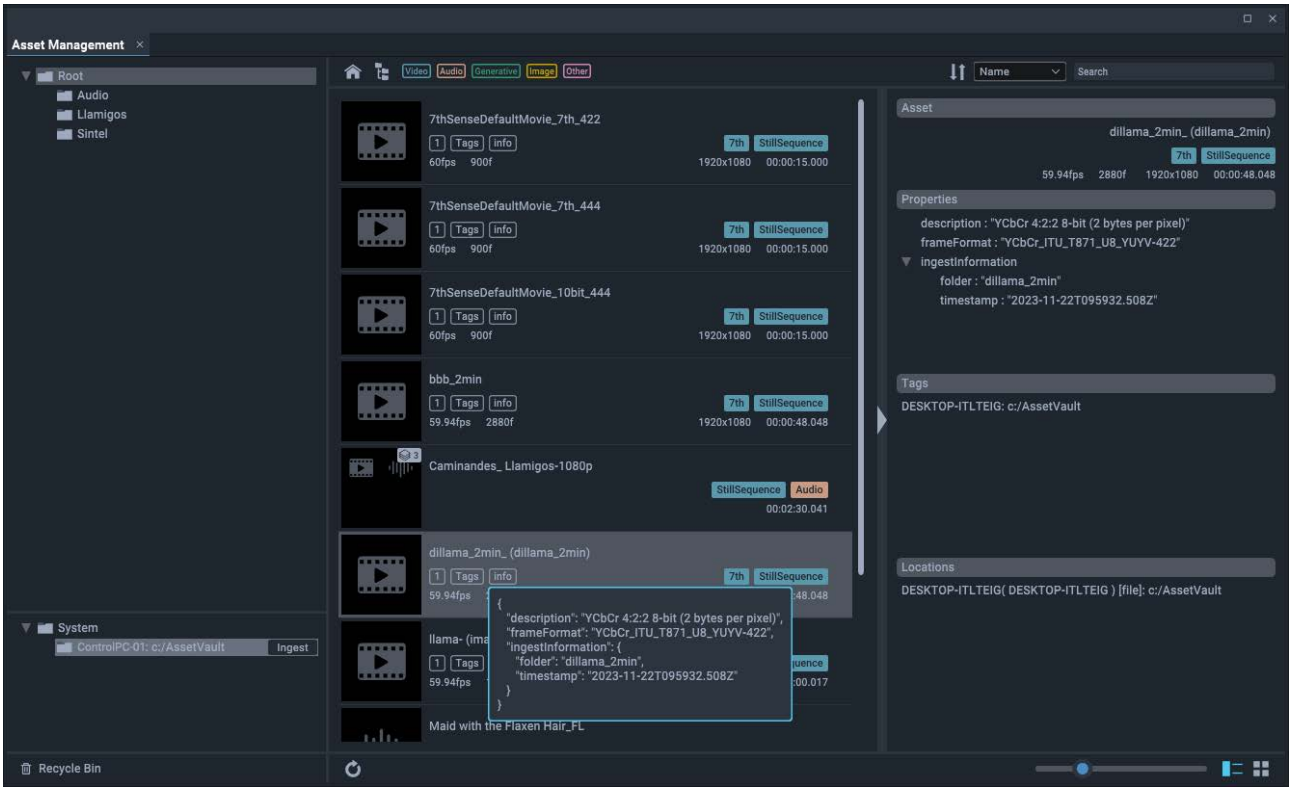
The result in the Asset Management [9] panel will be a parent group of assets, the children assets of which are all the sections. When the group (‘the movie’) is added to the timeline, each Actor recognises the name of ‘the movie’ and plays the section of that name that it has been given. An asset group ingested in this way will not split as if it were many things when added to a timeline, it is regarded as a single entity for the user.

Together, this will ensure all sections can be re-assembled, arranged and played back correctly and precisely.

Organise: Asset Management

Open the Asset Management panel to view and organise timeline assets that have been ingested, and manually ingest new assets to required vaults.

This panel addresses the underlying database of ingested assets and metadata. Assets, singly or as groups, can be dragged from here directly onto timelines. The illustration below shows that some assets in the Root folder have been grouped (the icon showing how many assets are in the group) and some are also in three subfolders.



Movie sequences that have been sectioned and distributed, will automatically be ingested as a group with the movie name. Do not ungroup these.

The right hand panel can be dragged out and shows all the details, which are also available in the list view or by clicking on the grey buttons for more detail about distributed locations, tags or property information (as shown in the illustration above). The number in the first button here shows that the asset is currently only in one location.

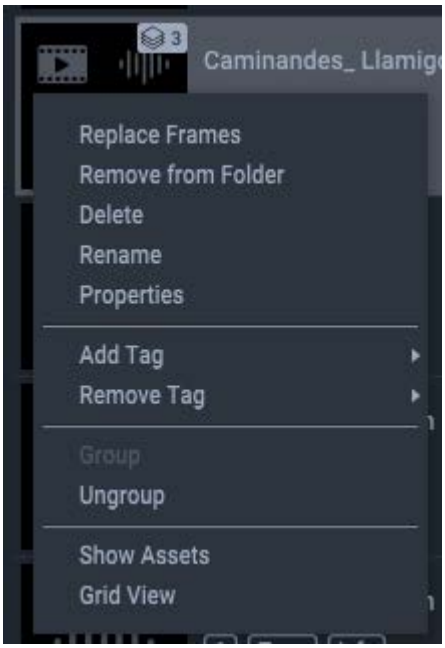
To begin with, all available assets will be listed together in the Root folder. From here, assets can be filtered by type, searched by name, tagged for destination, grouped, and organised into collections in subfolders.From the top of the panel, the folders can be navigated: ‘Home’ is top level, the folders icon includes or excludes items in subfolders. The category buttons filter the list out by asset type, and the whole asset database contents can be searched.

At the bottom:

Assets removed from the Asset Management panel go to the Recycle bin. Hover this to see the option to empty it (i.e. delete the assets altogether).

You can refresh the list (and reorder it), and change the view (tabular list, or thumbnail grid) using the icons, bottom right, or the right-click context menu on any asset listed, to change the view. Use the slider control to zoom the asset details and thumbnails.

Assets have a right-click context menu that allows assets to be grouped and ungrouped, removed from folders, deleted, and tagged. An asset group – for example of sectioned media comprising a whole – can be addressed with this menu as a group or individually.



Tags

Tags are used for selective media distribution, for example sending sectioned media to respective servers (see Configure Asset Logistics for Remote Servers 18), or identifying associated assets for user-defined purposes.

Ingesting assets

The Asset Logistics app monitors the designated 'Watch' folder, ready to ingest new timeline assets placed there when called on. Select a system device on the lower left section and click the Ingest button. Once ingested, these assets become available to any device through the Asset Manager.

Viewing assets

Select the Root folder in the Asset Management panel and click the ‘Refresh’ button if freshly ingested media does not appear straight away. View the assets shown in the panel as list or grid (thumbnail) format by right clicking any asset and choosing your view.

Organising assets

Folders

There are two ‘special’ folders in Asset Management: ‘Root’ and ‘Recycle Bin’. These can be renamed if required, but retain the special roles of being respectively the sum of all ingested assets, and assets extracted ready for complete deletion.

Creating and naming folders or subfolders and populating them does not move any files. It only categorises the associated records with metadata to better organise them. The Root folder on an Actor device remains the central and secure repository for ingested assets, however grouped, or associated in other folders. Any Actor device (node in the system) locates the physical assets on other Actor devices via the common shared database.

Add subfolders

Subfolders can be added or removed at any time: right-click the parent folder and select ‘New subfolder’.

Rename folders

Folders and subfolders can be renamed at any time: right-click and select ‘Rename’.

Remove folders

Right-click a folder to remove it. This places the selected assets or folders in the Recycle folder. Content from here can be added back into the root folder, but cannot be added directly to the timeline.

Remove from folder

A subfolder can be removed from a folder without affecting its contents in any other way.

Grouping

Grouping allows assets that belong together in use to be handled as a unit. The actual files are not moved around, but are given an association in the database. When a group is dragged onto a timeline, the members are divided into layers, not least because they can be different asset types

(e.g. video and multiple audio tracks). The exception is the ingested group of sectioned media, whereby each Actor recognises the asset name and plays its own version (section) of it, oblivious of the other section players.

Conditions:

- An asset can only belong in one group.
- Once in a group, only the whole group can be added to a timeline.
- To group assets, drag one onto another in the list, and a group is formed. Drag and add others in the same way onto the group header. Alternatively use Ctrl+click to make a selection, or Shift+click for a series, then right-click to 'Group'.
- The number of assets in a group is then shown.
- Right-click a group to 'Show assets', i.e. expand the group. Right-click again to Hide, i.e. collapse the group.
- To ungroup, right-click the group header and select 'Ungroup'.
- To join groups together, select both group headers and right-click to select 'Group'.
- To split a group into two groups, select assets for one group, right-click and select 'Split group'.

Deleting assets

Assets are deleted from the database permanently and the resource files deleted from the ingest folder, when they are deleted from the Recycle Bin.

Note: deleting very large assets can take a long time. If deleting to create disk space, check before adding new assets of similar size.

Frame Replacement

There will be times when corrections are made to subsets of frames in a movie sequence, and these need to replace existing frames on a Compere system. Replacement follows a similar process to the initial ingest, and requires specific instructions to match new frame details with old. Removing and reingesting many thousands of frames and then adding them back into timelines is not a realistic prospect for minor changes.

Frame replacement is a one-to-one, name-for-name action. It cannot introduce or remove frames or alter the duration or sequential naming of frames in any way.

Process overview

1. Un-distribute the frame sequence(s) from local devices, leaving only the ingest vault.
2. Capture the metadata and locations for existing frames in a fresh sidecar.json file to match against replacement frames of the same name. This can include sectional subfolders.
3. Add replacement frames and the sidecar.json to watch folders.
4. Edit or verify the watch folder names in the sidecar.json file.
5. Reingest the replacement frames.
6. Re-tag and distribute as originally configured.

Replacement step-by-step

The Asset Logistics (MediaLogisticsApp.exe) must be running, if it is not already running as a service.

1. Undistribute locally ingested assets subject to replacement by removing their tags. For example, select an asset group that has multiple locations such as ‘NAS’ (streaming) and ‘Local’ and remove the ‘Local’ tag from all at once. The Logistics app will delete locally distributed copies of the sequences.
2. Right-click the asset or asset group (or multi-click selected assets in a group) and select ‘Replace Frames’ to create a new file of metadata of the folders involved. This will raise a message ‘Copied to clipboard’. Open a text editor, paste this in and save it to your watch folder (as in the ingest process) as sidecar.json. Keep the file open to edit.

   Observe the structure of the json file. Only image sequence assets are recorded, and even a single asset sequence will have an asset group name, it will just be the same. Note that each set of replacement frames is applied to the original asset UUID; no new sequences are created.
3. Copy the replacement frames into a subfolder of the watch folder containing the sidecar.json (if sectioned replacements, then in respective section-named subfolders, if possible using the original ingest folder names).
4. Edit the sidecar.json to verify or complete the watch asset folder name(s). Do not edit the asset group details. Save and close it.

Example:


```
{
  "assetGroupUpdate": {
    "assetGroup": {
      "name": "MyBigMovie group",
      "uuid": "1275ded111502269f7a1b6413c966eb2"
    },
    "assetUpdates": [
      {
        "assetUpdate": {
          "asset": {
            "name": "MyBigMovie_ (MBM_x0_y0)",
            "uuid": "0bb88d9c621d34f06486915e53097625"
          },
          "folder": "MBM_section_01",
          "sectionIds": [
            "section_01"
          ]
        }
      },
      {
        "assetUpdate": {
          "asset": {
            "name": "MyBigMovie_ (MBM_x0_y2160)",
            "uuid": "0bb88d9c621d34f06486915e530998f0"
          },
          "folder": "MBM_section_02",
          "sectionIds": [
            "section_02"
          ]
        }
      },
      {
        "assetUpdate": {
          "asset": {
            "name": "MyBigMovie_ (MBM_x4096_y0)",
            "uuid": "0bb88d9c621d34f06486915e5309bba1"
          },
          "folder": "MBM_section_03",
          "sectionIds": [
            "section_03"
          ]
        }
      },
      {
        "assetUpdate": {
          "asset": {
            "name": "MyBigMovie_ (MBM_x4096_y2160)",
            "uuid": "0bb88d9c621d34f06486915e5309d7a2"
          },
          "folder": "MBM_section_04",
          "sectionIds": [
            "section_04"
          ]
        }
      }
    ]
  }
}
```


5. In Compere Asset Management, select the vault listed under System, and click Ingest.
6. Add destination tags back into each asset or asset group as a whole.

Checks and Errors

The replacement process will check for errors and reported them to the AssetLogistics log file.

Frame replacement will stop on the first error and any remaining frames for assets will be rejected back to the \RejectedFiles folder in the watch folder.

Frames that have been successfully replaced will have the original frames moved to an \Ejected subfolder of the asset folder in the \RejectedFiles folder within the watch folder.

Constraints checklist

- The asset group in the sidecar.json must be valid.
- All the assets in the sidecar.json must be valid.
- All the assets in the sidecar.json must belong to the asset group specified.
- All the folders in the sidecar.json must be present in the frame replacement folder (no more, no less).
- The names of the asset sections in the sidecar.json must match the names of the actual asset sections.
- Each replacement frame must match an existing frame of the same filename in the asset (it is not possible to insert missing frames or extend the length of the sequence).
- Each asset must be present on the ingest machine.
- Each asset must not be present on any other machine (i.e. if distributed, modify the tags such that the asset location count is 1, and only on the ingest machine).
- Each asset's first frame must be the same size and frame format as the first replacement frame for that asset.

Asset Logistics App

A preconfigured database using Apache CouchDB is optionally installed with Compere, and is essential for working with Actor servers. Installed alongside is FileZilla FTP server software for the distribution of assets throughout the whole system. There is no need to manually copy assets between devices and locations.

This section provides a more technical reference to the workings of the database into which all assets are ingested, creating metadata records used in their organisation. Metadata added is available in Compere's Asset Management panel, and is used to automate other processes such as the distribution of sectioned media.

The Asset Logistics app file may still be named 'MediaLogisticsApp', because its role widened in development. That name is used in examples here. On an Actor device, it is typically found in C:\7thSense\compere, along with a settings file 'ML-settings.xml'. It relies on an ingest ('**watch**') **folder** in a static location, into which new assets are copied prior to ingesting them into the database. The destination location is referred to as the **asset vault**.

Startup – Command Line Options

MediaLogisticsApp.exe has a number of settings that are configurable from the command line. The complete list of supported command-line parameters can be shown with the -h or –help switch, e.g.:

```
MediaLogisticsApp.exe -h
```

Logistics settings can also be stored in an XML settings file (**ML-settings.xml**) which (if used) should be in the same directory as the **MediaLogisticsApp.exe** executable.

The watch folder, asset vault, machine ID, machine name and machine address (hostname or specific IP) are the parameters which must *always* be specified, either on the command-line, e.g.:

```
MediaLogisticsApp.exe -i c:\assetWatch -d c:\assetVault --machine-name "controlPC-01" --
machine-address "10.100.101.10"
```

or in the **ML-settings.xml** file:

```
<?xml version="1.0" encoding="UTF-8"?>
<MediaLogisticsSettings
     watchFolder="c:\assetWatch"
     assetVault="c:\assetVault"
     machineName="controlPC-01"
     machineAddress="10.100.101.10"/>
```

Any setting(s) specified on the command-line take precedence over the same settings stored in the XML file.

Adding the -s switch on the command-line causes the current settings to be written to the specified file, e.g.:

```
MediaLogisticsApp.exe -i c:\assetWatch -d c:\assetVault --assetVaultTags
"local;section_1" --machine-name "controlPC-01" --machine-address "10.100.101.10" -
s lastSettings.xml
```

writes the following to the file **lastSettings.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<MediaLogisticsSettings
     watchFolder="c:\assetWatch"
     assetVault="c:\assetVault"
     assetVaultTags="local;section_1"
     machineName="controlPC-01"
     machineAddress="10.100.101.10"/>
```

This provides an easy mechanism to transfer selected settings to **ML-settings.xml** for inclusion the next time the Logistics app starts.

Note: Logistics does not monitor the **ML-settings.xml** for changes, it is only read on startup.

Configure Asset Logistics for Remote Servers

For the purposes of database replication (CouchDB only) and file distribution (FTP), Asset Logistics must know with which other nodes it can share database documents, and from where it can acquire missing asset files.

Configuration settings for servers for file distribution (FTP)

The `--machine-address` parameter (or `machineAddress` in **ML-settings.xml**) must be specified for each machine in the system. This is a mandatory setting, and if database replication is correctly set up (see below) this is sufficient for any machine to determine which assets are available across the system and how to download their files.

Note: 'Machine' refers to a device running the Logistics app. This may be an Actor, Conjurer or Juggler, or a standalone app running on a NAS, for example.

Adding tags to the asset vault to specify its role

The `–assetVaultTags` switch specifies a semi-colon separated list of tags to apply to the asset vault for file distribution roles, e.g.:

```
MediaLogisticsApp.exe -i c:\assetWatch -d c:\assetVault --assetVaultTags
"local;section_1" --machine-name "actor-01" --machine-address "10.100.101.10" -s
lastSettings.xml
```

or in ML-settings.xml:

```
<MediaLogisticsSettings
     watchFolder="c:\assetWatch"
     assetVault="c:\assetVault"
     assetVaultTags="local;section_1"
     machineName="actor-01"
     machineAddress="10.100.101.101"/>
```

The tags above specify that the vault will have local and section_1 tags. This indicates that the vault is local storage and should be used for assets which either have the tag local, or if the section is specified, local and section_1.

A NAS public vault would have the tag `nas`.

Configuration settings for servers for file streaming (SMB mapped-drive)

If a machine shares its files via SMB (e.g. a NAS), and there is sufficient bandwidth for Windows clients (Actors) to stream media directly from it for playback, this can be configured by adding the endpoints setting (JSON format) in **ML-settings.xml**. For instance:

```
<MediaLogisticsSettings
     watchFolder="/data/assetWatch"
     assetVault="/data/assetVault"
     machineName="netstore-01"
     machineAddress="nas"
```

```
     assetVaultTags="nas"
     endpoints='[ { "path": "N:/assetVault", "protocol": "mapped-drive" } ]'/>
```

For now, Actors only support the mapped-drive protocol, but it is envisaged that this could expand to http, smb, ftp (all using the machine address to access the host machine, and now listing ftp explicitly rather than implicitly).

Configuration settings for servers for database replication

Pushing changes to other nodes

The --db-nodes switch specifies a semi-colon separated list of remote hostnames or IP addresses for database replication for pushing changes to, e.g.:

```
MediaLogisticsApp.exe -i c:\assetWatch -d c:\assetVault --machine-name "controlPC-01" --
machine-address "10.100.101.10" --db-nodes "192.168.0.217;ubuntu-
desktop;192.168.0.249;laptop1" -s lastSettings.xml
```

or in ML-settings.xml:

```
<MediaLogisticsSettings
     watchFolder="c:\assetWatch"
     assetVault="c:\assetVault"
     machineName="controlPC-01"
     machineAddress="10.100.101.10"
     databaseNodeAddresses="192.168.0.217;ubuntu-desktop;192.168.0.249; laptop1"/>
```

Pulling changes from other nodes

The --db-nodes-from switch specifies a semi-colon separated list of remote hostnames or IP addresses for database replication from which to pull changes, e.g.:

```
MediaLogisticsApp.exe -i c:\assetWatch -d c:\assetVault --machine-name "controlPC-01" --
machine-address "10.100.101.10" --db-nodes-from "192.168.0.217;ubuntu-
desktop;192.168.0.249;laptop1" -s lastSettings.xml
```

or in ML-settings.xml:

```
<MediaLogisticsSettings
     watchFolder="c:\assetWatch"
     assetVault="c:\assetVault"
     machineName="controlPC-01"
     machineAddress="10.100.101.10"
     databaseNodeAddressesFrom="192.168.0.217;ubuntu-desktop;192.168.0.249; laptop1"/>
```

Explanation

As mentioned above, when Compere starts MediaLogistics, it only sets the ‘watch folder’ and ‘asset vault’. For MediaLogisticsApp instances that are started by Compere, ensure that:

- machineName="name"
- machineAddress="address"
- databaseNodeAddresses="semi;colon;separated;hostnames;or;ip;addresses"

and/or

- databaseNodeAddressesFrom="semi;colon;separated;hostnames;or;ip;addresses"

are present in the **ML-settings.xml** file.

Note that a "machine ID" is required for MediaLogisticsApp to use – Compere will automatically generate a machine-id.txt containing a UUID on first startup; MediaLogistics will read the same file to get the machineID – if this file is not present, MediaLogisticsApp will create it.

The remote nodes specified by those parameters must have:

- CouchDB running on them for the purposes of database replication (changes occurring on this node are either pushed to the remote nodes, or remote nodes pull from this node, depending on configuration) and
- FileZilla FTP server setup and correctly configured for the purposes of asset file distribution (any files missing on this node are pulled from the remote nodes).

Example

Typically, all nodes will replicate to all other nodes in a fully-connected topology, but other topologies are possible (e.g. star topology) if required, allowing the use of thin clients which do not have their own CouchDB instance or FTP server running on them.

For instance, to fully connect four nodes (IP address 192.168.0.1, 192.168.0.2, 192.168.0.3, 192.168.0.4) for the purposes of database and file replication, and also connect a control PC (hostname laptop1) for the purposes of database replication, but not file replication could have the following MediaLogistics settings:

Control PC (laptop1)

```
<MediaLogisticsSettings
     watchFolder="<watch>"
     assetVault="<vault>"
     assetVaultTags=""
     machineName="Control PC laptop1"
     machineAddress="laptop1"
     databaseNodeAddresses="192.168.0.1"
     databaseNodeAddressesFrom="192.168.0.1"/>
```

Node 1 (192.168.0.1)

```
<MediaLogisticsSettings
     watchFolder="<watch>"
     assetVault="<vault>"
     assetVaultTags="local;section_1"
     machineName="actor-01"
     machineAddress="192.168.0.1"
     databaseNodeAddresses="192.168.0.2;192.168.0.3;192.168.0.4;nas"
     databaseNodeAddressesFrom="192.168.0.2;192.168.0.3;192.168.0.4;nas"/>
```

Node 2 (192.168.0.2)

```
<MediaLogisticsSettings
     watchFolder="<watch>"
     assetVault="<vault>"
     assetVaultTags="local;section_2"
     machineName="actor-02"
     machineAddress="192.168.0.2"
     databaseNodeAddresses="192.168.0.1;192.168.0.3;192.168.0.4;nas"
     databaseNodeAddressesFrom="192.168.0.1;192.168.0.3;192.168.0.4;nas"/>
```

Node 3 (192.168.0.3)

```
<MediaLogisticsSettings
     watchFolder="<watch>"
     assetVault="<vault>"
     assetVaultTags="local;section_3"
     machineName="actor-03"
     machineAddress="192.168.0.3"
     databaseNodeAddresses="192.168.0.1;192.168.0.2;192.168.0.4;nas"
     databaseNodeAddressesFrom="192.168.0.1;192.168.0.2;192.168.0.4;nas"/>
```

Node 4 (192.168.0.4)

```
<MediaLogisticsSettings
     watchFolder="<watch>"
     assetVault="<vault>"
     assetVaultTags="local;section_4"
     machineName="actor-04"
     machineAddress="192.168.0.4"
     databaseNodeAddresses="192.168.0.1;192.168.0.2;192.168.0.3;nas"
     databaseNodeAddressesFrom="192.168.0.1;192.168.0.2;192.168.0.3;nas"/>
```

NAS

```
<MediaLogisticsSettings
     watchFolder="/data/Watch"
     assetVault="/data/Vault"
     machineName="netstore-01"
     machineAddress="nas"
     assetVaultTags="nas"
     endpoints='[ { "path": "N:/Vault", "protocol": "mapped-drive" } ]'
     databaseNodeAddresses="192.168.0.1;192.168.0.2;192.168.0.3;192.168.0.4"
     databaseNodeAddressesFrom="192.168.0.1;192.168.0.2;192.168.0.3; 192.168.0.4"/>
```

Note that **laptop1** in this example, is the only node with responsibility for pulling/pushing database changes to 192.168.0.1 (Node1); this is purely to simplify the configuration, it would be equally valid for laptop1 to be included in all of the databaseNodeAddresses for the other nodes.

Document Information

Date	Document edition	Software version	Revision Details	Author/Editor
November 2023	1	Compere 1.0-7 Media Logistics 1.0.0	New release	Andie Davidson

E: info@7thsense.one
W: 7thsense.one

7thSense Design Ltd

2 The Courtyard, Shoreham Road
Upper Beeding
Steyning
West Sussex
BN44 3TN
UK

T: +44 (0) 1903 812299

7thSense LLC

4207 Vineland Rd
Suite M1
Orlando, FL 32811
USA

T: +1 407 505 5200