

TMSS10BatchLoader

TMSS10BatchLoader creates [Customer](#), [Media](#) and [Repositories](#) by loading a Configuration file into the TapeTrack Database.

Synopsis

```
TMSS10BatchLoader [-c configuration file] [-S logonstring]
```

Options

- -c The config file argument specifies the path to the Batch Configuration File.
- -S Logon string argument sets the [Server logon information](#).










If an argument value starts with the value `FI:`, the value for that attribute will be read from the file name specified immediately after it. For example, if a file named `ACME_volumes` has the content `ACME_*. *`, specifying `FI:/etc/ACME_volumes` will have the same effect as `-V ACME_*. *`. Using `FI:` to refer to a file for an attribute value works for all attributes.

If the pattern being matched starts with `*LIST:` then rather than matching the literal value, each record in the file specified as `*LIST:filename` will be tested. If one matches the match will succeed.

2021/06/16 02:14 · Scott Cunliffe

[include page](#)

Editable Fields

- [Customer-ID](#)
- [Customer Description](#)
- [Media-ID](#)
- [Media Description](#)
- [Repository-ID](#)
- [Repository Description](#)
- [Repository Icon](#)
 -  Rack (value = 0)
 -  Transport (value = 1)
 -  Vault (value = 2)
 -  Library (value = 3)
 -  Legal Hold (value = 4)
 -  Disaster Recovery (value = 5)
 -  Destruction (value = 6)
 -  Ordered (value = 7)
 -  Scratch (value = 8)
- [Next Repository](#)
- [Auto Container flag](#)

Technical Support

The TapeTrack Software is commercially supported by a full time help desk staff.

If you are experiencing problems or want some advice on how to configure or use the product please see the [Accessing Technical Support](#) page.

Exit Statuses

1. **zero** Program has ended successfully.
2. **non-zero** Program has not ended successfully.

Environment

Files

stderr: Diagnostic messages.

Example

Configuration File

Configuration file to create two [Customers](#) with Attributes:

Customer 0001

- [Customer-ID](#): US01, [Customer Description](#): New York Data Center.
- [Media-ID](#): LTO, [Media Description](#): LTO Cartridge.
- [Repository-ID](#): LIBR, [Repository Description](#): Library, [Repository Type](#): 1, [Next Repository](#): OFFS.
- [Repository-ID](#): OFFS, [Repository Description](#): Iron Mountain, [Repository Type](#): 2.
- [Repository-ID](#): DEST, [Repository Description](#): Destroyed, [Repository Type](#): 6.
- [Repository-ID](#): HOLD, [Repository Description](#): Legal Hold, [Repository Type](#): 4 .

Customer 0002

- [Customer-ID](#): US02, [Customer Description](#): Los Angeles Data Center.
- [Media-ID](#): 3490, [Media Description](#): IBM 3490 Cartridge.
- [Repository-ID](#): LIBR, [Repository Description](#): Library, [Repository Type](#): 1, [Next Repository](#): OFFS.
- [Repository-ID](#): OFFS, [Repository Description](#): Iron Mountain, [Repository Type](#): 2 and [Next Repository](#): LIBR.

[Download Sample Configuration file.](#)

```
setup=  
{  
  customers=
```

```

(
  {
    id="US01"; description="New York Data Center";
    media=
    (
      {
        id="LT0"; description="LT0 Cartridge";
        repositories=
        (
          { id="LIBR"; description="Library"; type=3;
next="OFFS"; },
          { id="OFFS"; description="Iron Mountain"; type=2;
next="LIBR"; },
          { id="DEST"; description="Destroyed"; type=6; },
          { id="HOLD"; description="Legal hold"; type=4; }
        )
      },
      {
        id="3490"; description="IBM 3490 Cartridge";
        repositories=
        (
          { id="LIBR"; description="Library"; type=3;
next="OFFS"; },
          { id="OFFS"; description="Iron Mountain"; type=2;
next="LIBR"; }
        )
      }
    )
  },
  {
    id="US02"; description="Los Angeles Data Center";
    media=
    (
      {
        id="LT0"; description="LT0 Cartridge";
        repositories=
        (
          { id="LIBR"; description="Library"; type=3; },
          { id="OFFS"; description="Iron Mountain"; type=2; }
        )
      },
      {
        id="3490"; description="IBM 3490 Cartridge";
        repositories=
        (
          { id="LIBR"; description="Library"; type=3; },
          { id="OFFS"; description="Iron Mountain"; type=2; }
        )
      }
    )
  }
}

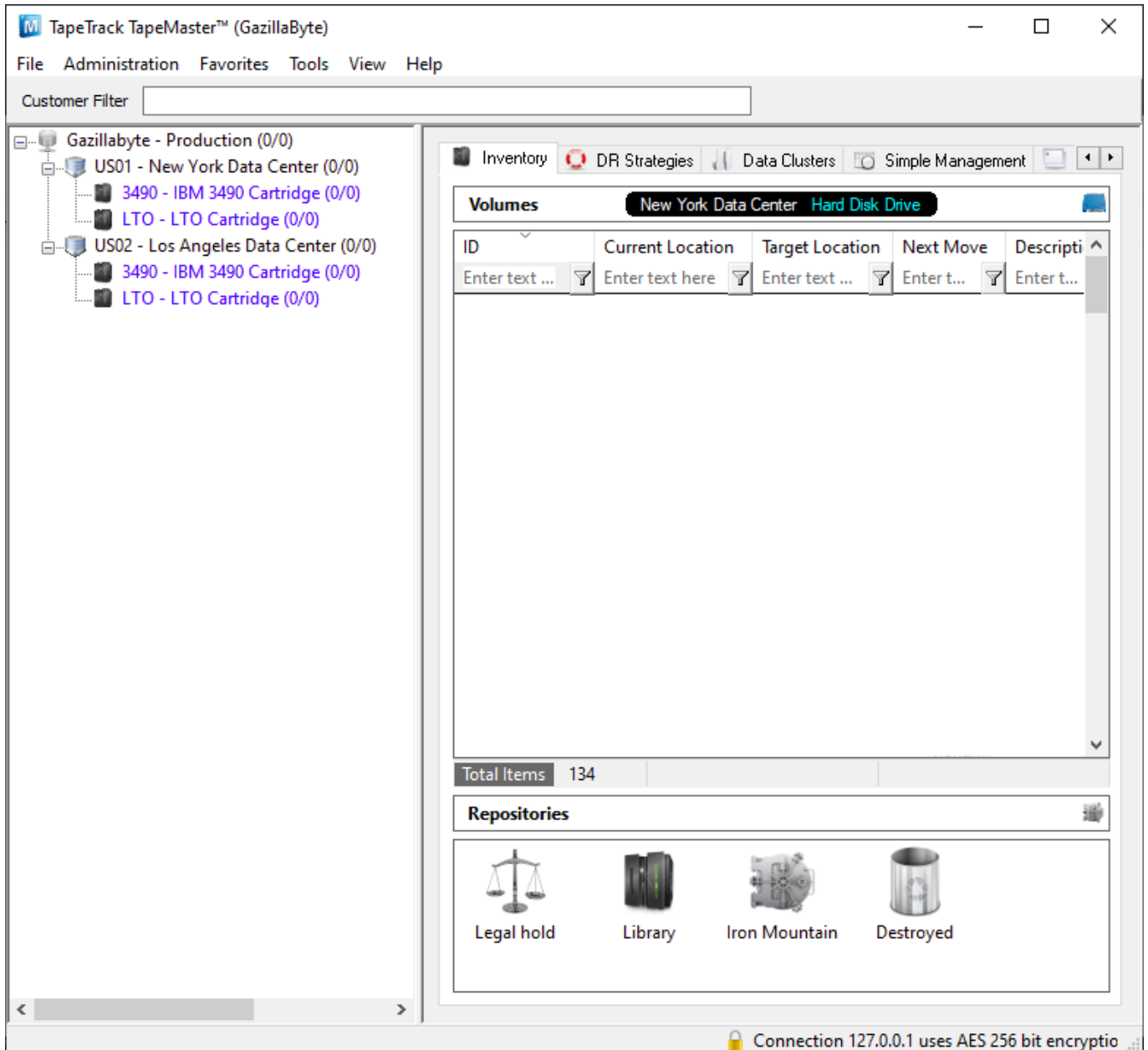
```

```
);  
};
```

Command line syntax

```
TMSS10BatchLoader -S user:-password@localhost -c master.cfg
```

Output Results



Configuration Files - Minimising Duplicate Code

When creating multiple [Customers](#) with identical [Media Types](#) and/or [Repositories](#) it is possible to

create a configuration file with these Attributes and include it in the master configuration file to save duplication of code blocks.

To create two customers, US01 and US02, with:

- **Media-ID:** LTO, **Media Description:** LTO Cartridge.
- **Media-ID:** 3490, **Media Description:** IBM 3490 Cartridge.
- **Repository-ID:** LIBR, **Repository Description:** Library, **Repository Type:** 1, **Next Repository:** OFFS.
- **Repository-ID:** OFFS, **Repository Description:** Iron Mountain, **Repository Type:** 2 and **Next Repository:** LIBR.

A configuration file may “include” the contents of another file using an include directive.

This directive has the effect of in-lining the contents of the named file at the point of inclusion.



An include directive must appear on its own line in the input. It has the form:

```
@include "filename"
```

Any backslashes or double quotes in the filename must be escaped as '\\\' and '\\"', respectively.

Configuration Files

master.cfg

setup=

```
{
  customers=
  (
    {
      id="US01"; description="New York Data Center";
      @include "include.cfg"
    },
    {
      id="US02"; description="Los Angeles Data Center";
      @include "include.cfg"
    }
  );
};
```

include.cfg

```
media=
(
  {
    id="LT0"; description="LT0 Cartridge";

    repositories=
    (
      { id="LIBR"; description="Library"; type=3;
next="OFFS"; },
      { id="OFFS"; description="Iron Mountain"; type=2;
next="LIBR"; }
    )
  },
  {
    id="3490"; description="IBM 3490 Cartridge";

    repositories=
    (
      { id="LIBR"; description="Library"; type=3;
next="OFFS"; },
      { id="OFFS"; description="Iron Mountain"; type=2;
next="LIBR"; }
    )
  }
)
```

Command line syntax

```
TMSS10BatchLoader -S user:-password@localhost -c master.cfg
```

Output Results

The screenshot shows the TapeTrack TapeMaster™ (Gazillabyte) web interface. At the top, there is a menu bar with 'File', 'Administration', 'Favorites', 'Tools', 'View', and 'Help'. Below the menu is a 'Customer Filter' input field. The left sidebar displays a tree view of the production environment:

- Gazillabyte - Production (0/0)
 - US01 - New York Data Center (0/0)
 - 3490 - IBM 3490 Cartridge (0/0)
 - LTO - LTO Cartridge (0/0)
 - US02 - Los Angeles Data Center (0/0)
 - 3490 - IBM 3490 Cartridge (0/0)
 - LTO - LTO Cartridge (0/0)

The main content area has several tabs: 'Inventory', 'DR Strategies', 'Data Clusters', 'Simple Management', and 'Uploaded'. The 'Inventory' tab is active, showing a 'Volumes' section with a filter for 'New York Data Center' and 'LTO Cartridge'. Below this is a table with the following columns: ID, Current Location, Target Location, Flags, Move Time, and Due in. The table is currently empty. Below the table, it shows 'Total Items: 0'. The 'Repositories' section displays two icons: 'Library' and 'Iron Mountain'.

Changelog

Cannot load rss feed.

From: <https://rtfm.tapetrack.com/> - **TapeTrack Documentation**

Permanent link: <https://rtfm.tapetrack.com/cli/tmss10batchloader?rev=1737497272>

Last update: **2025/01/21 22:07**

