TMSS10BatchLoader

Synopsis

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

Description

Creates Customer, Media and Repositories by loading a configuration file into the TapeTrack Database.

Options

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

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.

<sxh> setup= {

```
 customers=
 (
    {
        id="US01"; description="New York Data Center";
```

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

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

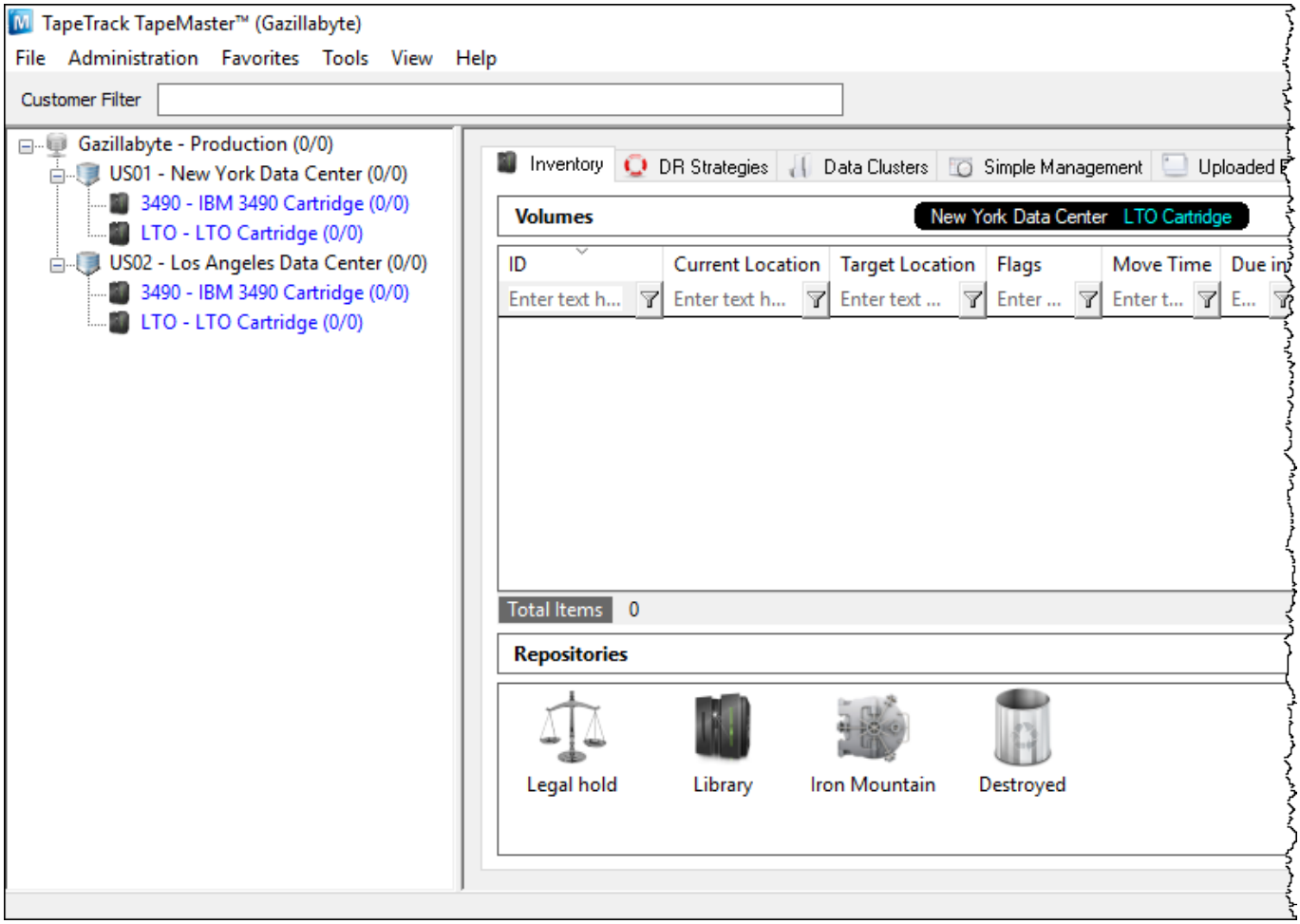
```
               repositories=
               (
                   { id="LIBR"; description="Library"; type=3; },
                   { id="OFFS"; description="Iron Mountain"; type=2; }
               )
           }
        )
    }
 );
```

}; </sxh>

Command line syntax

<sxh>TMSS10BatchLoader -S user:-password@localhost -c master.cfg</sxh>

Output Results



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 inlining 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

include.cfg

<sxh>

```
        media=
        (
           {
               id="LTO"; description="LTO 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"; }
               )
           }
        )
```

</sxh>

master.cfg

<sxh> setup= {

```
 customers=
 (
    {
        id="US01"; description="New York Data Center";
        @include "include.cfg"
```

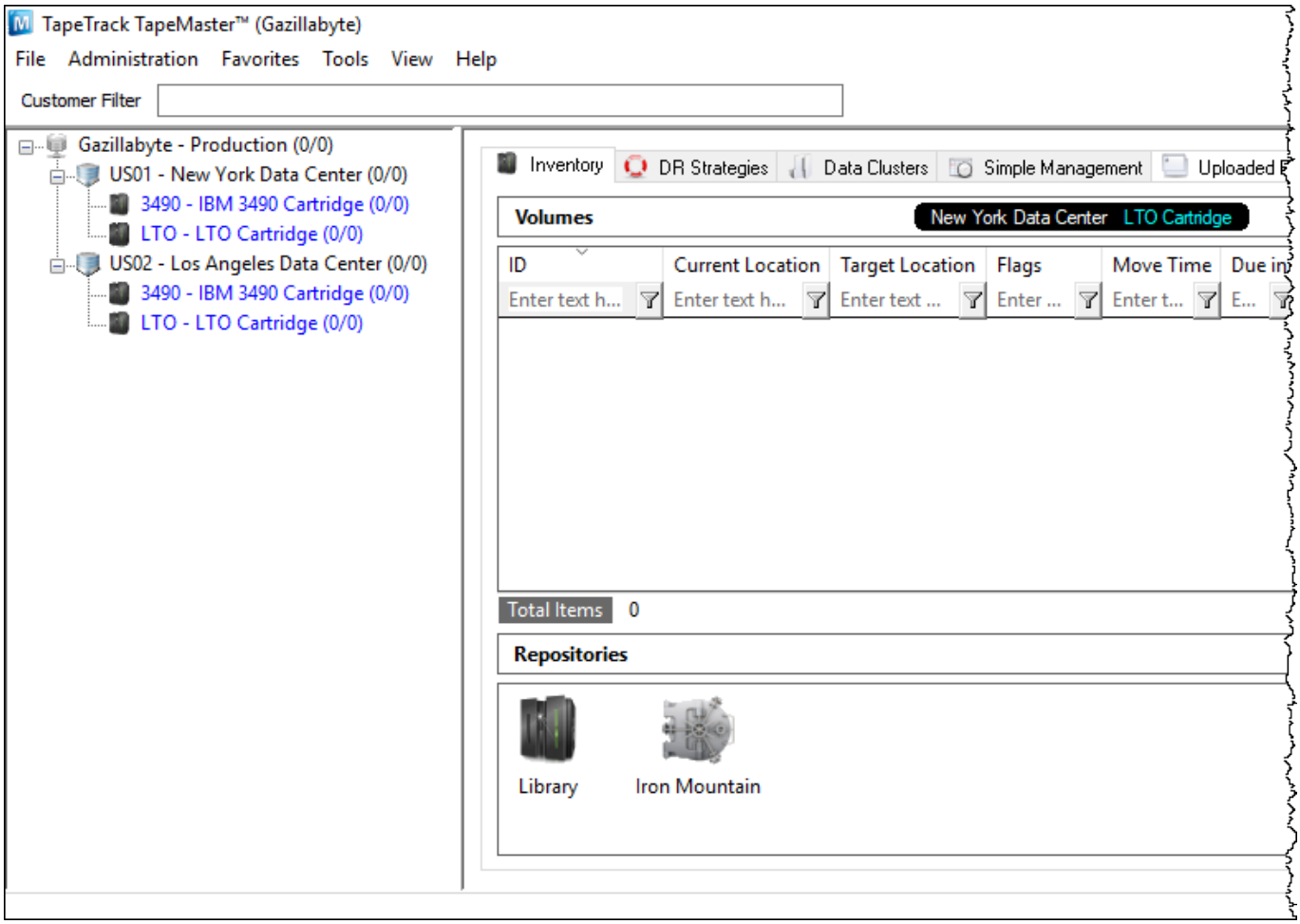
```
    },
    {
        id="US02"; description="Los Angeles Data Center";
           @include "include.cfg"
    }
);
```

}; </sxh>

Command line syntax

<sxh>TMSS10BatchLoader -S user:-password@localhost -c master.cfg</sxh>

Output Results



From:
https://docs.tapetrack.com/ - **TapeTrack Documentation**

Permanent link:
https://docs.tapetrack.com/cli/tmss10batchloader?rev=1551087281

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