Seed Data

CloudTest version

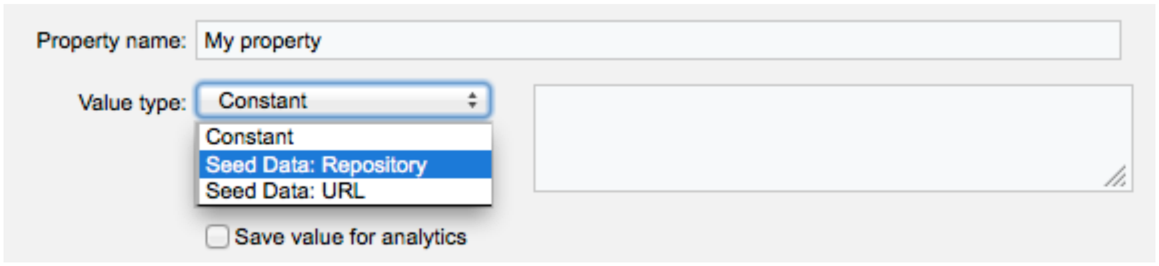
This document applies to CloudTest build 5483 and later.

Introduction

This feature allows sets of seed data values in tabular/CSV form to be extracted from a Seed Data object in the Repository or from a CSV file accessible through some URL and distributed to various Custom Properties in a Composition. A typical use case would be a set of login IDs for a test.

Basic procedure

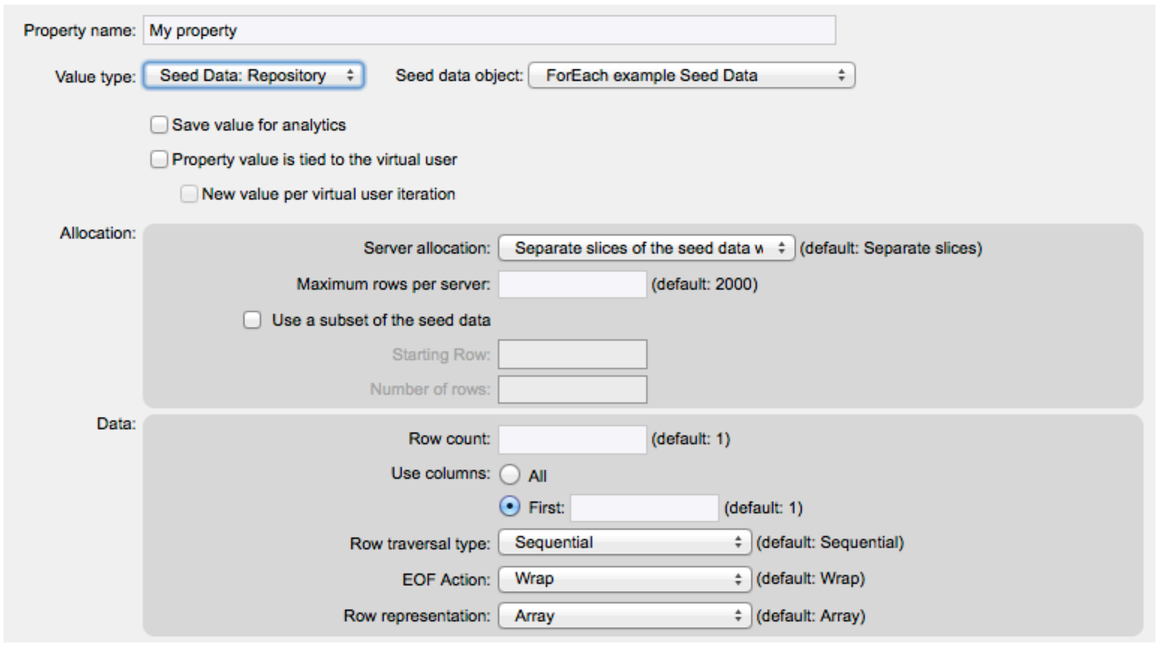
For any Custom Property defined at the Track level or below, you can specify that the value of the Custom Property is to be populated from seed data at runtime:



Property name: My property

Value type: Constant

Save value for analytics



Property name: My property

Value type: Seed Data: Repository   Seed data object: ForEach example Seed Data

Save value for analytics

Property value is tied to the virtual user

New value per virtual user iteration

Allocation:

Server allocation: Separate slices of the seed data w (default: Separate slices)

Maximum rows per server: (default: 2000)

Use a subset of the seed data

Starting Row:

Number of rows:

Data:

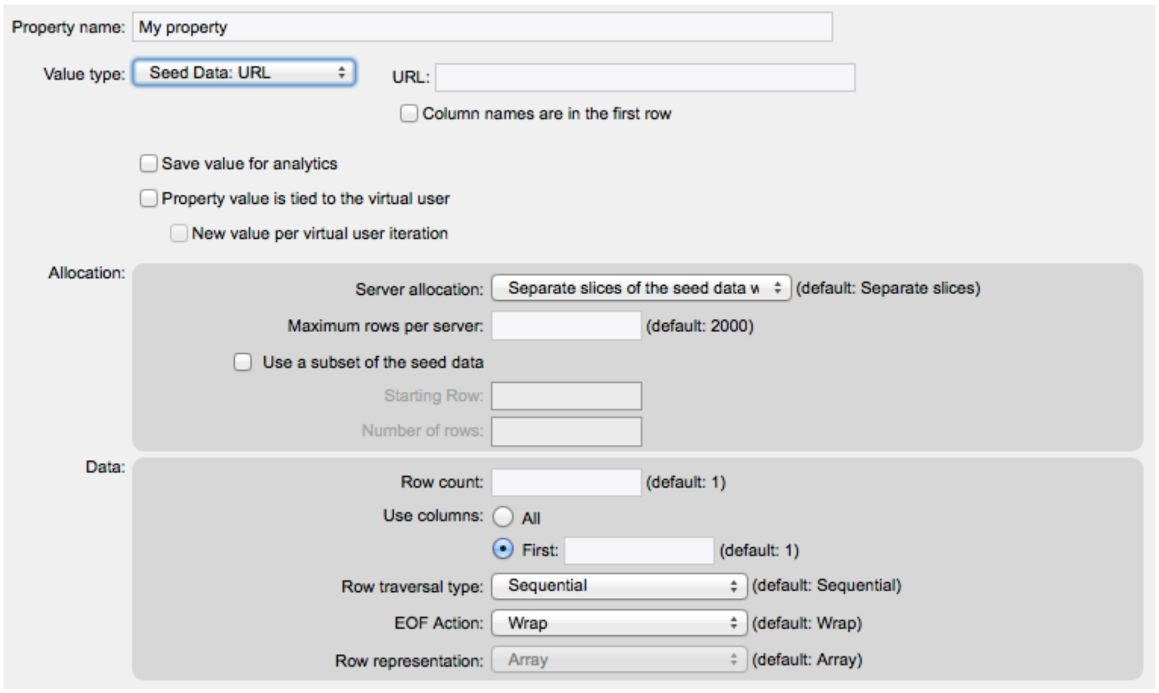
Row count: (default: 1)

Use columns: All [✓ First:] (default: 1)

Row traversal type: Sequential (default: Sequential)

EOF Action: Wrap (default: Wrap)

Row representation: Array (default: Array)



Property name: My property

Value type: Seed Data: URL

URL:

☐ Column names are in the first row

☐ Save value for analytics

☐ Property value is tied to the virtual user

☐ New value per virtual user iteration

Allocation:

Server allocation: Separate slices of the seed data w (default: Separate slices)

Maximum rows per server: (default: 2000)

☐ Use a subset of the seed data

Starting Row:

Number of rows:

Data:

Row count: (default: 1)

Use columns: ○ All

● First: (default: 1)

Row traversal type: Sequential (default: Sequential)

EOF Action: Wrap (default: Wrap)

Row representation: Array (default: Array)

When the Composition is played, each time an instance of the item containing the Custom Property is played, the value of the Custom Property will be filled in with a value from the specified Seed Data source.

The Custom Property can then be used anywhere within the Composition that a Custom Property can be used, such as ISSEs, Validation, Repeats, and so forth.

How seed data is distributed

A Composition may contain many different Custom Properties referring to multiple sources of seed data. When the Composition is loaded, the seed data requirements within the Composition are analyzed and the seed data is read from the sources (Repository objects and/or URLs).

To prevent the reading of seed data from becoming a performance issue during the play of the Composition, all of the seed data is read and distributed when the Composition is loaded. For each seed data source, a “slice” (chunk) of rows of seed data is allocated to each server on which the Composition will play. Each server then manages and reads from its slice of seed data during the test. Each server keeps its slice of seed data in memory.

In a multi-server Composition, only one of the Maestro servers reads the seed data, and it then distributes the slices to all of the other Maestro servers involved with the Composition.

Since the seed data is only read during Composition load, and not during Composition play, Composition play is not affected by the time it takes to read seed data from the source. In addition, since only a single Maestro server reads the seed data at load time, the server hosting the data (in the case of URL-based seed data) will not be subject to reads from multiple servers, thus lessening the performance needs for the hosting server.

The size of the slice of data sent to each server depends upon the settings of the Custom Properties and the size (number of rows) of the seed data source. The slice size is controlled by the "*Maximum rows per server*", "*Starting row*", and "*Number of rows*" settings that are specified. If the number of rows of seed data available from the source is not enough to support the specified number of rows that would result from those settings, then as many rows as possible are distributed.

The "*Server allocation*" setting controls how the data is divided among the servers in a multi-server Composition. If the option "*Separate slices of the seed data will be sent to each server*" is chosen, the data is divided evenly among all of the servers, and each server gets a unique slice of the data. On the other hand, if the option "*The same single slice of seed data will be sent to all servers*" is chosen, then a copy of same entire slice of seed data is sent all servers, and each server will be using the same seed data.

For example, for a Composition that plays on 5 servers with the option "*Separate slices of the seed data will be sent to each server*", if the "*Maximum rows per server*" is specified as 1,000, then if there are at least 5,000 rows in the source then 1,000 rows of seed data will be distributed to each server. However, if the source only contains 600 rows, then only 120 rows would be distributed to each server (600/5). If there are not enough rows of seed data to distribute at least one row to each server, the load will fail.

If instead, the option "*The same single slice of seed data will be sent to all servers*" is chosen, then all 5 servers would share the same single set of seed data (either 5,000 or 600 rows in the above examples).

If a Composition contains references to multiple seed data sources, each server will have a separate, independent slice for each such seed data source.

If multiple Custom Properties refer to the same seed data source, they will all extract data from the same single slice (each will get different unique rows from that slice).

If multiple Custom Properties refer to the same seed data source but specify different values for "*Maximum rows per server*", the highest specified value will be used.

If multiple Custom Properties refer to the same seed data source, all references must specify the same "*Starting row*" and "*Number of rows*" settings, if those settings are specified.

No security (such as Basic Authentication or other user ID or passwords) is supported when accessing seed data via URL. However, Repository Seed Data objects can be encrypted with a password. At Composition load time, you will be prompted to enter any such necessary passwords.

EOF Action

The "EOF Action" setting specifies what should happen if any server reaches the end of it's slice of seed data during play of the Composition. The possible settings are:

- "*Wrap*"

  The server will start over at the beginning of its slice of seed data.

- "*Null*"

  The values of the Custom Properties will be set to nulls. If according to the settings the property value is to be an array, it will be set to an array of nulls of the appropriate dimension and size.

- "*Error*"

  The item containing the Custom Property will fail. Error handling will proceed according to the regular "Failure action" settings of the item, it's parents, and the Composition.

- "*Stop Track and Drain*"

  The current instance of the item's parent Track on the current server (and everything in it) is stopped. No new repeats or parallel repeat renewals will occur for that Track on this server. Other instances of the Track on this server that already exist will continue to their normal end, but no new instances will be started on this server.

  In other words, if using the recommended approach of having each Track represent a "Virtual User", and if using "Parallel Repeat Renewal" to cause each Virtual User to be automatically replaced when it ends, then this setting will cause no new Virtual Users to be created on this server when its seed data runs out.

- "*Stop Composition*"

  The entire Composition (on all servers) will be stopped immediately (as if the "Stop" button had been pressed).

If multiple Custom Properties refer to the same seed data source, each such reference can have it's own EOF Action, they do not all need to be the same.

Row traversal options

The "*Row traversal type*" setting specifies how each server traverses through its slice of the seed data.

- "*Sequential*"

  The server traverses through it's set of data sequentially. The "*EOF Action*" setting applies and indicates what to do if the end of the server's set of data is reached.

- "*Random with replacement*"

  Whenever a new value is needed to fill in a new property value, one is chosen randomly from the server's set of data. The "*EOF Action*" setting does not apply, since each selection is just made randomly from the entire set of data for that server, and therefore the server can never "run out" of values.

- "*Random without replacement*"

  Whenever a new value is needed, one is chosen at random and marked as used. Thus a unique value is chosen each time from the server's set of data. Since in this case the server can run out of data, the "*EOF Action*" setting applies.

Assignment of seed data values to individual properties

When the Composition is playing, any time that a new item starts playing, such as a Clip, Chain, or Message, if that item contains Custom Properties that reference seed data, the server on which the item is playing extracts the next available row from the slice of seed data that server has been allocated and sets the Custom Property's value. Depending upon the settings, this may result in multiple rows being extracted for each Custom Property value.

The "*Column count*" setting specifies the number of columns to retrieve from the seed data. If the setting is one, the property is set to a single value. If the setting is greater than one, the property value type depends upon the option chosen for the "*Row representation*" setting.

If "*Array*" is chosen the property is set to an array value, with one array member for each column in the seed data. If there are fewer columns in the seed data than specified in "*Column count*", then the extra values in the array are set to null. The array will always be the size specified by "*Column count*".

If "*Struct*" is chosen, the property is set to a struct value, with one member for each column in the seed data. The "*Struct*" option can only be used when accessing seed data via URL and the "*Column names are in the first row*" option is chosen, or when a Repository seed data object is being used.

An example of the use of multiple columns would be where the seed data contains username and password pairs. In this case the seed data might contain two columns, the first column being the username and the second column being the corresponding password. Each use of the data needs access to both values as a pair, so "*Column count*" is set to 2. The value assigned to each instance of the Custom Property will be either an array or a struct with the two values.

The "*Row count*" setting specifies the number of rows of seed data to distribute to each individual instance of the Custom Property.

If the value is greater than one, then the Custom Property's value will be an array value, one entry in the array for each row.

If both "*Column count*" and "*Row count*" are greater than one, then the property value will be either a two-dimensional array, where the first index is the row and the second index is the column (if "*Array*" was chosen), or an array of structs (if "*Struct*" was chosen).

Leading and trailing white space are trimmed from the seed data values.

Tying seed data values to the virtual user

In Cloud Test, an instance of a Track is generally used to represent a "virtual user". Parallel repeat renewal of the Track is generally used to represent "iterations" of a virtual user.

If there are multiple references to the same seed data source from different Custom Properties within the same virtual user, each such Custom Property will get a different value from the seed data.

For example, if Custom Properties defined in two different Messages in the same Clip both refer to the same seed data source, each will get it's own value from the seed data.

If the "*Property value is tied to the virtual user*" option is selected, then all Custom Properties within the same virtual user that reference the same seed data source will get the same single seed data value. (In the example above, the two Custom Properties in the two Messages will get the same value.) In addition, the same value will be used for all iterations of the virtual user (each renewed parallel repeat of the Track).

If the "*New value per virtual user iteration*" option is selected, then each iteration of the virtual user will get a new value. In the example above, the two Custom Properties in the two Messages will get the same value within each iteration of the virtual user, but each iteration will have a different value.

Discussion of random distributions

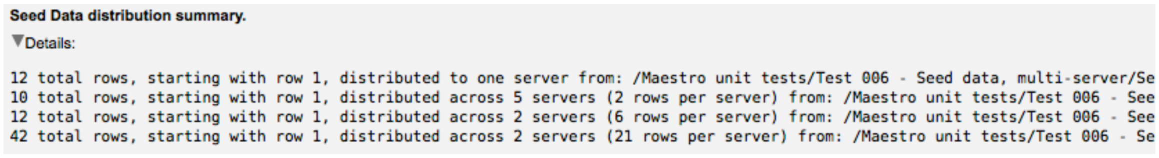
To cause every instance of a property to get a random value, then set "*Server allocation*" to "*Separate slices of the seed data will be sent to each server*" in order to cause different sets (slices) of the data to be sent to each server. Each server then selects randomly from within it's slice.

To cause every server to randomly select from the entire set of data, when it is not required that every value be used exactly once across all servers, then set "*Server allocation*" to "*The same single slice of seed data will be sent to all servers*" to cause the entire set of data to be sent to all servers. All of the servers will then be randomly selecting from the entire range of values.

Result events

Various events are logged to the Result that indicate how the seed data was allocated and used. These events can be viewed in the "Result Details" and "Event Log" Dashboard widgets. All such events are logged at the "Composition" level and thus will be present even when the Composition is played in "Load" mode.

When play starts, a "Seed Data distribution summary" event is logged that shows a consolidated summary of the slices distributed to each server involved in the Composition:

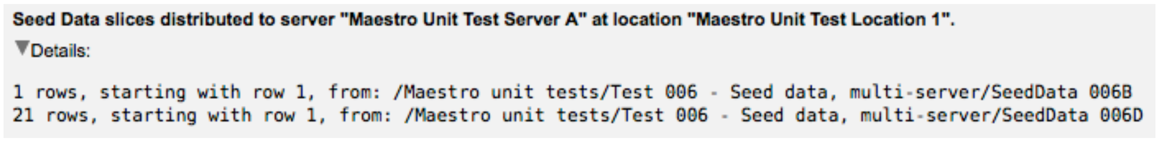


Seed Data distribution summary.

▼Details:

```
12 total rows, starting with row 1, distributed to one server from: /Maestro unit tests/Test 006 - Seed data, multi-server/Se
10 total rows, starting with row 1, distributed across 5 servers (2 rows per server) from: /Maestro unit tests/Test 006 - See
12 total rows, starting with row 1, distributed across 2 servers (6 rows per server) from: /Maestro unit tests/Test 006 - See
42 total rows, starting with row 1, distributed across 2 servers (21 rows per server) from: /Maestro unit tests/Test 006 - Se
```

When play starts on each server involved in the Composition, a "Seed Data slices distributed" event is logged for the server that shows information about the slices distributed to that server:

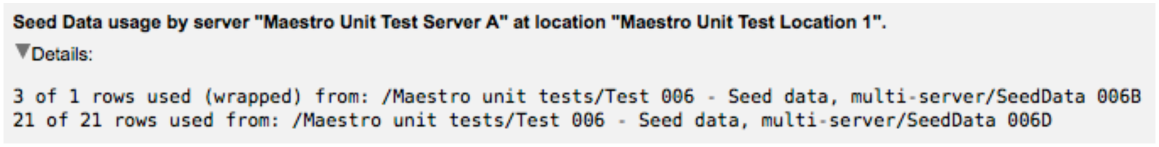


Seed Data slices distributed to server "Maestro Unit Test Server A" at location "Maestro Unit Test Location 1".

▼Details:

```
1 rows, starting with row 1, from: /Maestro unit tests/Test 006 - Seed data, multi-server/SeedData 006B
21 rows, starting with row 1, from: /Maestro unit tests/Test 006 - Seed data, multi-server/SeedData 006D
```

When play completes on each server involved in the Composition, a "Seed Data usage" event is logged that shows how much data that server used from the slice, as well as whether it was necessary to wrap back to the start of the data (if allowed to do so by the Custom Property settings). In this example, the first slice wrapped twice as the slice size was a single row and a total of 3 rows were allocated:



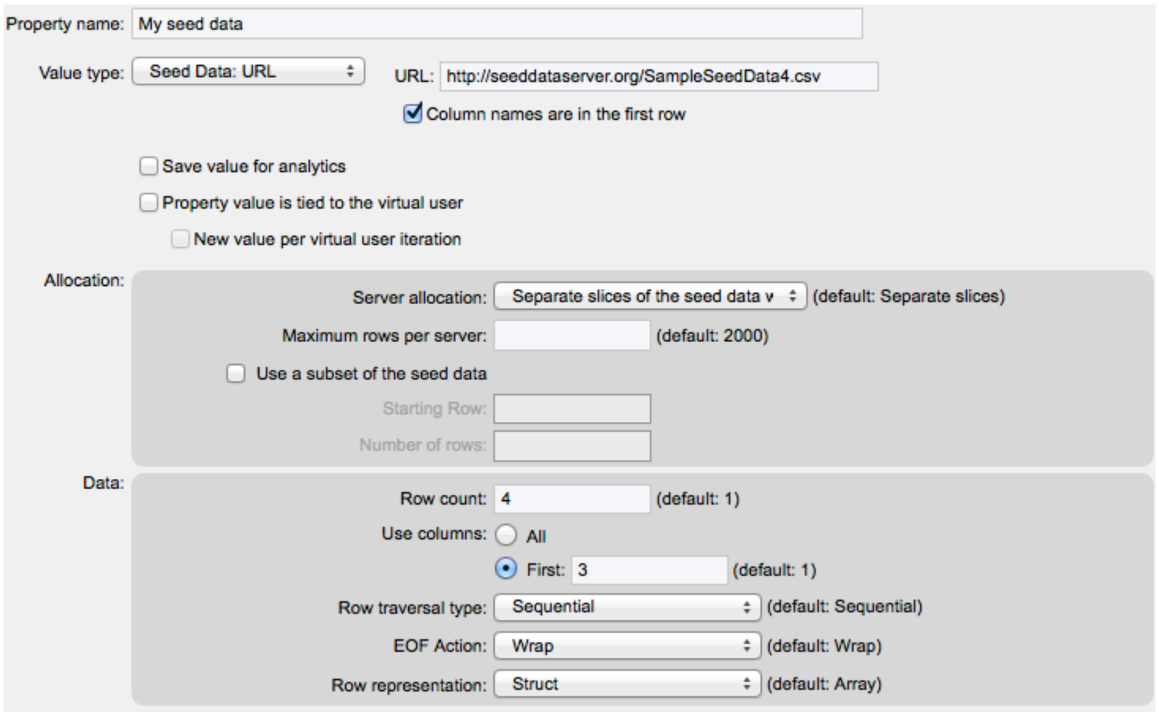
Seed Data usage by server "Maestro Unit Test Server A" at location "Maestro Unit Test Location 1".

▼Details:

```
3 of 1 rows used (wrapped) from: /Maestro unit tests/Test 006 - Seed data, multi-server/SeedData 006B
21 of 21 rows used from: /Maestro unit tests/Test 006 - Seed data, multi-server/SeedData 006D
```

Example

In this example, the Composition contains a Track named "Track 1" with a single Custom Property named "My seed data" whose value comes from seed data via URL with the following settings:



Property name: My seed data

Value type: Seed Data: URL URL: http://seeddataserver.org/SampleSeedData4.csv

[✓ Column names are in the first row]

Save value for analytics

Property value is tied to the virtual user

New value per virtual user iteration

Allocation:

Server allocation: Separate slices of the seed data v (default: Separate slices)

Maximum rows per server: (default: 2000)

Use a subset of the seed data

Starting Row:

Number of rows:

Data:

Row count: 4 (default: 1)

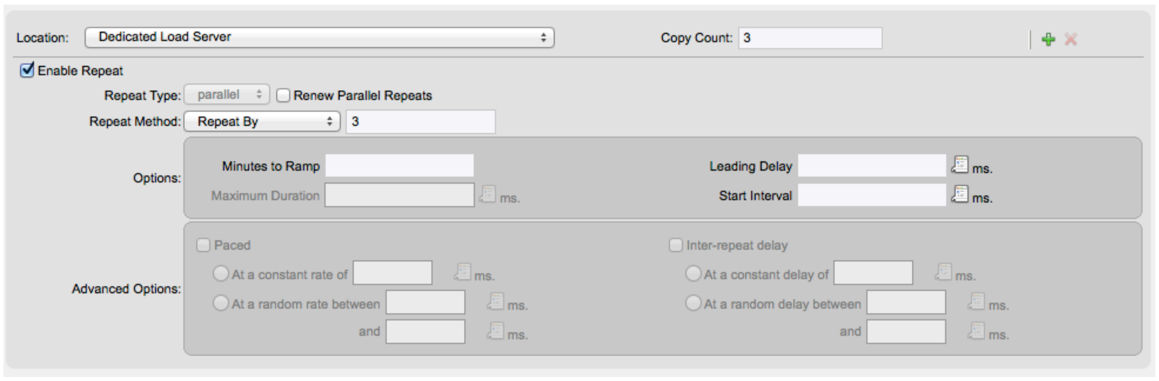
Use columns: All [✓ First: 3] (default: 1)

Row traversal type: Sequential (default: Sequential)

EOF Action: Wrap (default: Wrap)

Row representation: Struct (default: Array)

The Track has 3 virtual users (repeats) and is copied to 3 servers. The virtual users do not iterate, so "Parallel repeat renewal" was not selected.



Location: Dedicated Load Server Copy Count: 3

[✓ Enable Repeat]

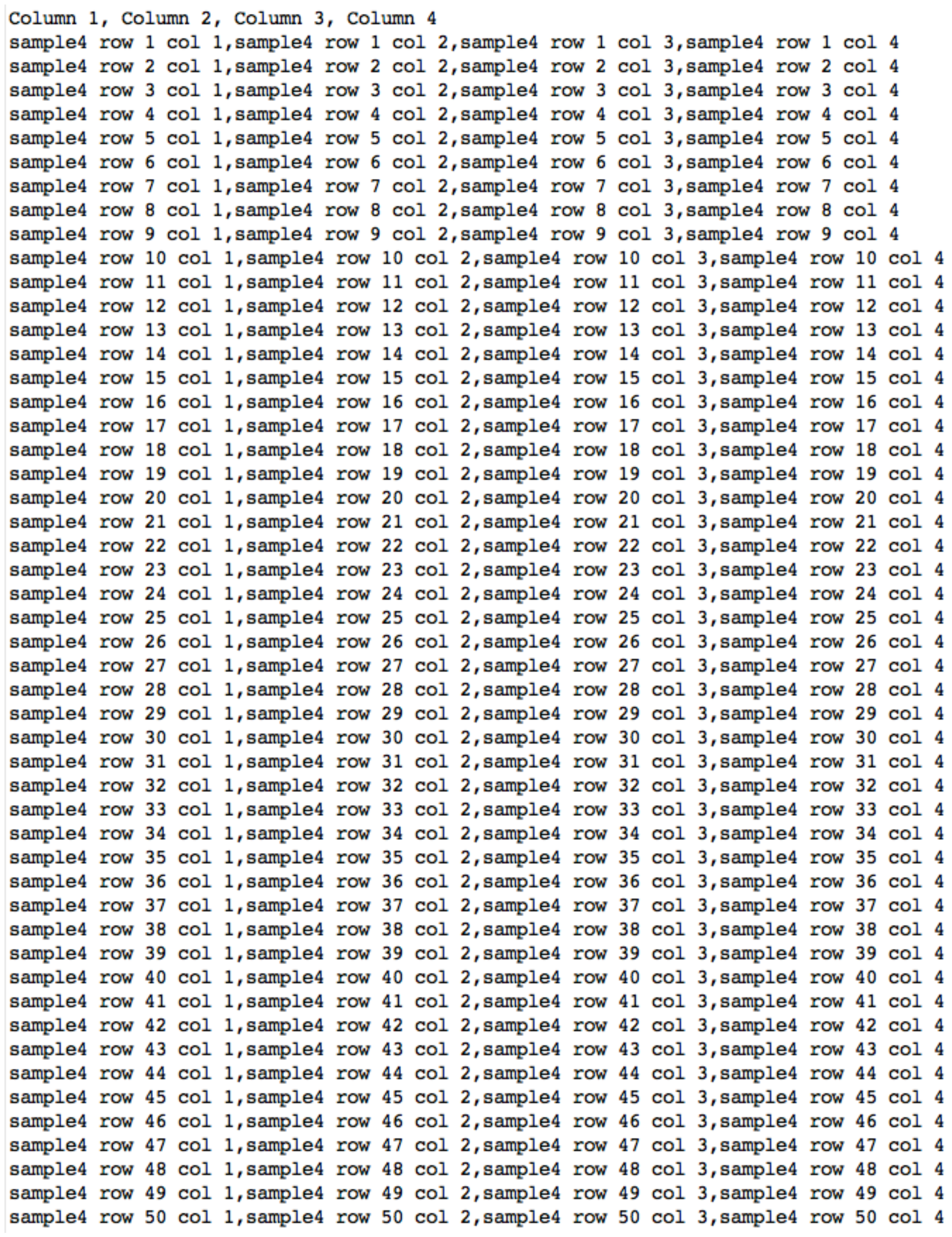
Repeat Type: parallel Renew Parallel Repeats

Repeat Method: Repeat By 3

Options: Minutes to Ramp Leading Delay ms. Maximum Duration ms. Start Interval ms.

Advanced Options: Paced At a constant rate of ms. At a random rate between ms. and ms. Inter-repeat delay At a constant delay of ms. At a random delay between ms. and ms.

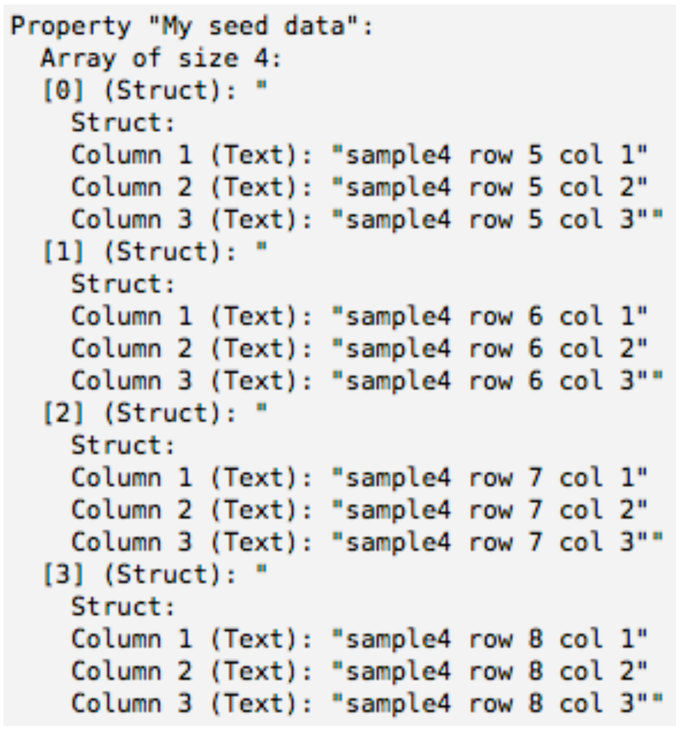
The referenced Seed Data file has the following contents:



```
Column 1, Column 2, Column 3, Column 4
sample4 row 1 col 1,sample4 row 1 col 2,sample4 row 1 col 3,sample4 row 1 col 4
sample4 row 2 col 1,sample4 row 2 col 2,sample4 row 2 col 3,sample4 row 2 col 4
sample4 row 3 col 1,sample4 row 3 col 2,sample4 row 3 col 3,sample4 row 3 col 4
sample4 row 4 col 1,sample4 row 4 col 2,sample4 row 4 col 3,sample4 row 4 col 4
sample4 row 5 col 1,sample4 row 5 col 2,sample4 row 5 col 3,sample4 row 5 col 4
sample4 row 6 col 1,sample4 row 6 col 2,sample4 row 6 col 3,sample4 row 6 col 4
sample4 row 7 col 1,sample4 row 7 col 2,sample4 row 7 col 3,sample4 row 7 col 4
sample4 row 8 col 1,sample4 row 8 col 2,sample4 row 8 col 3,sample4 row 8 col 4
sample4 row 9 col 1,sample4 row 9 col 2,sample4 row 9 col 3,sample4 row 9 col 4
sample4 row 10 col 1,sample4 row 10 col 2,sample4 row 10 col 3,sample4 row 10 col 4
sample4 row 11 col 1,sample4 row 11 col 2,sample4 row 11 col 3,sample4 row 11 col 4
sample4 row 12 col 1,sample4 row 12 col 2,sample4 row 12 col 3,sample4 row 12 col 4
sample4 row 13 col 1,sample4 row 13 col 2,sample4 row 13 col 3,sample4 row 13 col 4
sample4 row 14 col 1,sample4 row 14 col 2,sample4 row 14 col 3,sample4 row 14 col 4
sample4 row 15 col 1,sample4 row 15 col 2,sample4 row 15 col 3,sample4 row 15 col 4
sample4 row 16 col 1,sample4 row 16 col 2,sample4 row 16 col 3,sample4 row 16 col 4
sample4 row 17 col 1,sample4 row 17 col 2,sample4 row 17 col 3,sample4 row 17 col 4
sample4 row 18 col 1,sample4 row 18 col 2,sample4 row 18 col 3,sample4 row 18 col 4
sample4 row 19 col 1,sample4 row 19 col 2,sample4 row 19 col 3,sample4 row 19 col 4
sample4 row 20 col 1,sample4 row 20 col 2,sample4 row 20 col 3,sample4 row 20 col 4
sample4 row 21 col 1,sample4 row 21 col 2,sample4 row 21 col 3,sample4 row 21 col 4
sample4 row 22 col 1,sample4 row 22 col 2,sample4 row 22 col 3,sample4 row 22 col 4
sample4 row 23 col 1,sample4 row 23 col 2,sample4 row 23 col 3,sample4 row 23 col 4
sample4 row 24 col 1,sample4 row 24 col 2,sample4 row 24 col 3,sample4 row 24 col 4
sample4 row 25 col 1,sample4 row 25 col 2,sample4 row 25 col 3,sample4 row 25 col 4
sample4 row 26 col 1,sample4 row 26 col 2,sample4 row 26 col 3,sample4 row 26 col 4
sample4 row 27 col 1,sample4 row 27 col 2,sample4 row 27 col 3,sample4 row 27 col 4
sample4 row 28 col 1,sample4 row 28 col 2,sample4 row 28 col 3,sample4 row 28 col 4
sample4 row 29 col 1,sample4 row 29 col 2,sample4 row 29 col 3,sample4 row 29 col 4
sample4 row 30 col 1,sample4 row 30 col 2,sample4 row 30 col 3,sample4 row 30 col 4
sample4 row 31 col 1,sample4 row 31 col 2,sample4 row 31 col 3,sample4 row 31 col 4
sample4 row 32 col 1,sample4 row 32 col 2,sample4 row 32 col 3,sample4 row 32 col 4
sample4 row 33 col 1,sample4 row 33 col 2,sample4 row 33 col 3,sample4 row 33 col 4
sample4 row 34 col 1,sample4 row 34 col 2,sample4 row 34 col 3,sample4 row 34 col 4
sample4 row 35 col 1,sample4 row 35 col 2,sample4 row 35 col 3,sample4 row 35 col 4
sample4 row 36 col 1,sample4 row 36 col 2,sample4 row 36 col 3,sample4 row 36 col 4
sample4 row 37 col 1,sample4 row 37 col 2,sample4 row 37 col 3,sample4 row 37 col 4
sample4 row 38 col 1,sample4 row 38 col 2,sample4 row 38 col 3,sample4 row 38 col 4
sample4 row 39 col 1,sample4 row 39 col 2,sample4 row 39 col 3,sample4 row 39 col 4
sample4 row 40 col 1,sample4 row 40 col 2,sample4 row 40 col 3,sample4 row 40 col 4
sample4 row 41 col 1,sample4 row 41 col 2,sample4 row 41 col 3,sample4 row 41 col 4
sample4 row 42 col 1,sample4 row 42 col 2,sample4 row 42 col 3,sample4 row 42 col 4
sample4 row 43 col 1,sample4 row 43 col 2,sample4 row 43 col 3,sample4 row 43 col 4
sample4 row 44 col 1,sample4 row 44 col 2,sample4 row 44 col 3,sample4 row 44 col 4
sample4 row 45 col 1,sample4 row 45 col 2,sample4 row 45 col 3,sample4 row 45 col 4
sample4 row 46 col 1,sample4 row 46 col 2,sample4 row 46 col 3,sample4 row 46 col 4
sample4 row 47 col 1,sample4 row 47 col 2,sample4 row 47 col 3,sample4 row 47 col 4
sample4 row 48 col 1,sample4 row 48 col 2,sample4 row 48 col 3,sample4 row 48 col 4
sample4 row 49 col 1,sample4 row 49 col 2,sample4 row 49 col 3,sample4 row 49 col 4
sample4 row 50 col 1,sample4 row 50 col 2,sample4 row 50 col 3,sample4 row 50 col 4
```

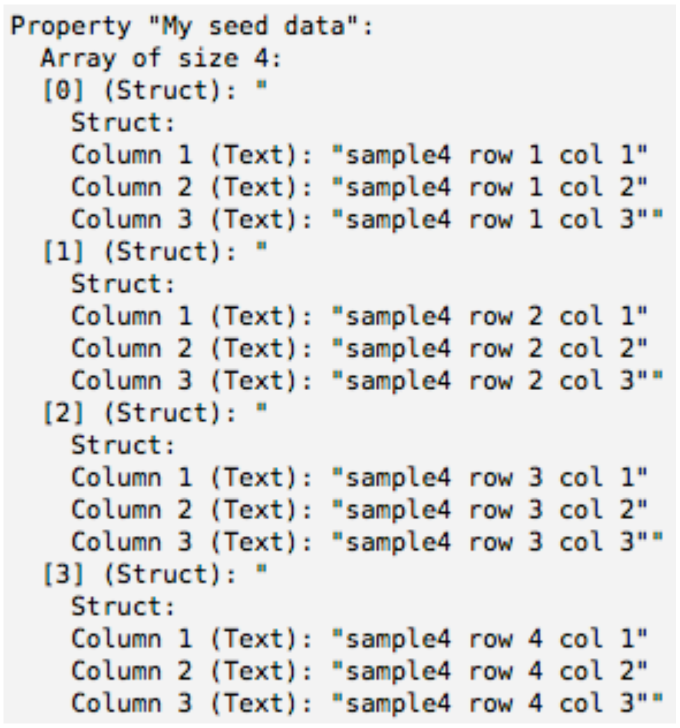
When the Composition plays, the following instances of the Track are given the following values for the Custom Property “My seed data”:

Track: "Track 1 (copy 1 of 3)" [0]



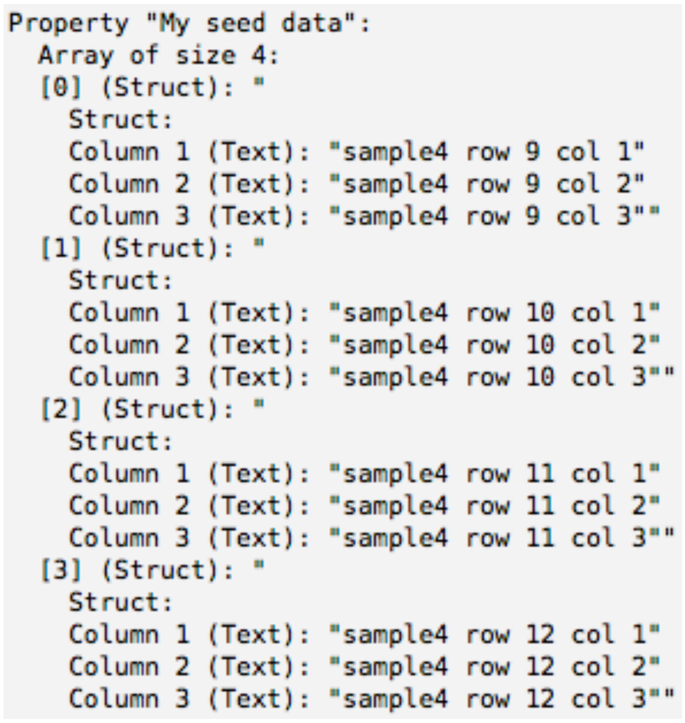
```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 5 col 1"
    Column 2 (Text): "sample4 row 5 col 2"
    Column 3 (Text): "sample4 row 5 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 6 col 1"
    Column 2 (Text): "sample4 row 6 col 2"
    Column 3 (Text): "sample4 row 6 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 7 col 1"
    Column 2 (Text): "sample4 row 7 col 2"
    Column 3 (Text): "sample4 row 7 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 8 col 1"
    Column 2 (Text): "sample4 row 8 col 2"
    Column 3 (Text): "sample4 row 8 col 3""
```

Track: "Track 1 (copy 1 of 3)" [1]



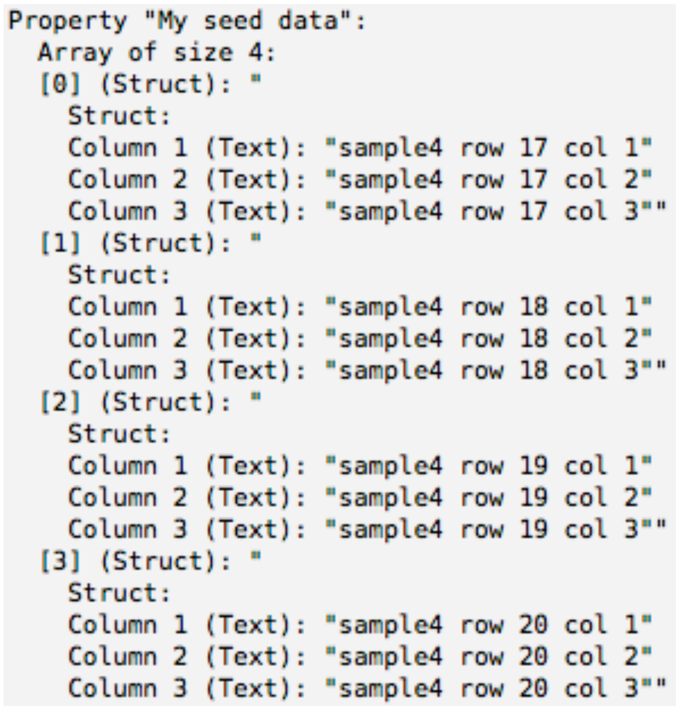
```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 1 col 1"
    Column 2 (Text): "sample4 row 1 col 2"
    Column 3 (Text): "sample4 row 1 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 2 col 1"
    Column 2 (Text): "sample4 row 2 col 2"
    Column 3 (Text): "sample4 row 2 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 3 col 1"
    Column 2 (Text): "sample4 row 3 col 2"
    Column 3 (Text): "sample4 row 3 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 4 col 1"
    Column 2 (Text): "sample4 row 4 col 2"
    Column 3 (Text): "sample4 row 4 col 3""
```

Track: "Track 1 (copy 1 of 3)" [2]



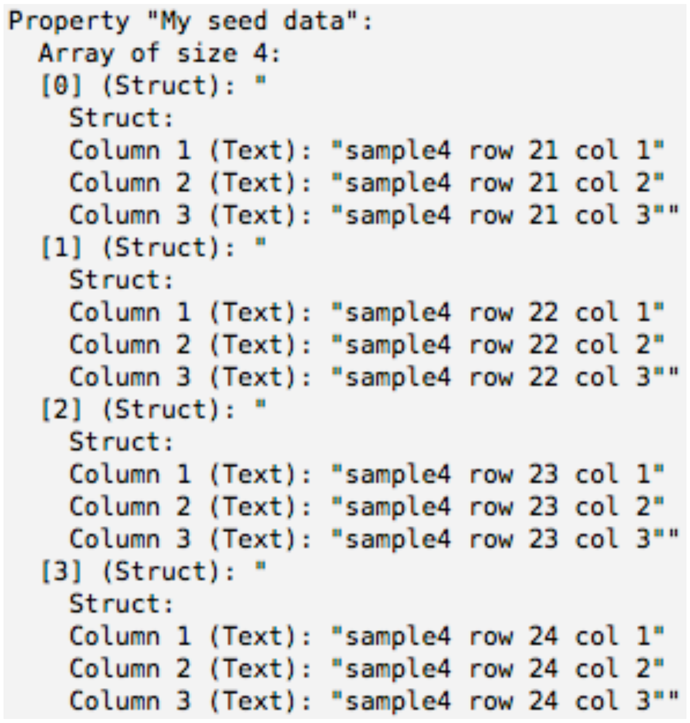
```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 9 col 1"
    Column 2 (Text): "sample4 row 9 col 2"
    Column 3 (Text): "sample4 row 9 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 10 col 1"
    Column 2 (Text): "sample4 row 10 col 2"
    Column 3 (Text): "sample4 row 10 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 11 col 1"
    Column 2 (Text): "sample4 row 11 col 2"
    Column 3 (Text): "sample4 row 11 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 12 col 1"
    Column 2 (Text): "sample4 row 12 col 2"
    Column 3 (Text): "sample4 row 12 col 3""
```

Track: "Track 1 (copy 2 of 3)" [0]



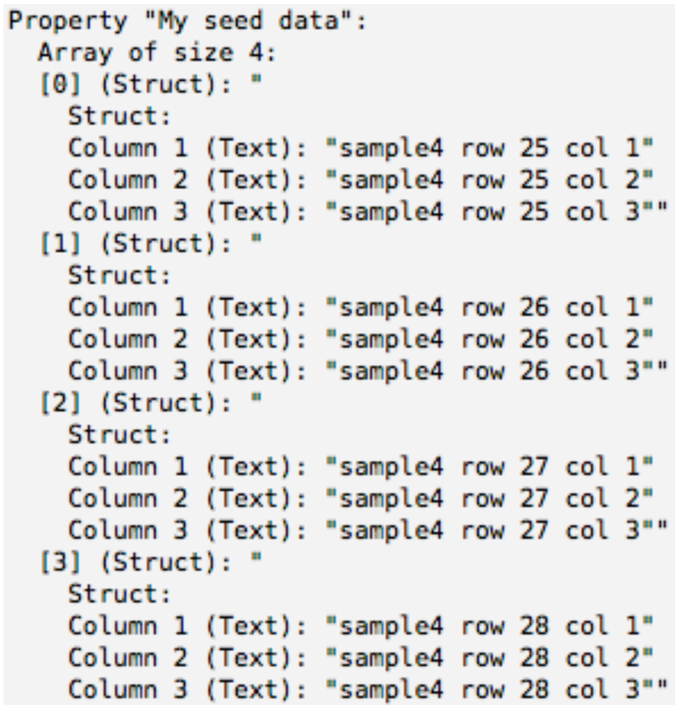
```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 17 col 1"
    Column 2 (Text): "sample4 row 17 col 2"
    Column 3 (Text): "sample4 row 17 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 18 col 1"
    Column 2 (Text): "sample4 row 18 col 2"
    Column 3 (Text): "sample4 row 18 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 19 col 1"
    Column 2 (Text): "sample4 row 19 col 2"
    Column 3 (Text): "sample4 row 19 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 20 col 1"
    Column 2 (Text): "sample4 row 20 col 2"
    Column 3 (Text): "sample4 row 20 col 3""
```

Track: "Track 1 (copy 2 of 3)" [1]



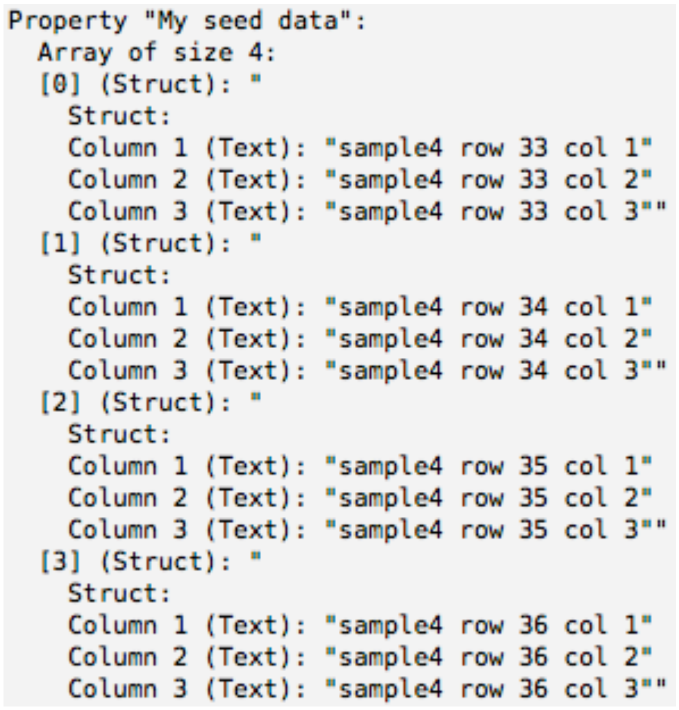
```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 21 col 1"
    Column 2 (Text): "sample4 row 21 col 2"
    Column 3 (Text): "sample4 row 21 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 22 col 1"
    Column 2 (Text): "sample4 row 22 col 2"
    Column 3 (Text): "sample4 row 22 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 23 col 1"
    Column 2 (Text): "sample4 row 23 col 2"
    Column 3 (Text): "sample4 row 23 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 24 col 1"
    Column 2 (Text): "sample4 row 24 col 2"
    Column 3 (Text): "sample4 row 24 col 3""
```

Track: "Track 1 (copy 2 of 3)" [2]



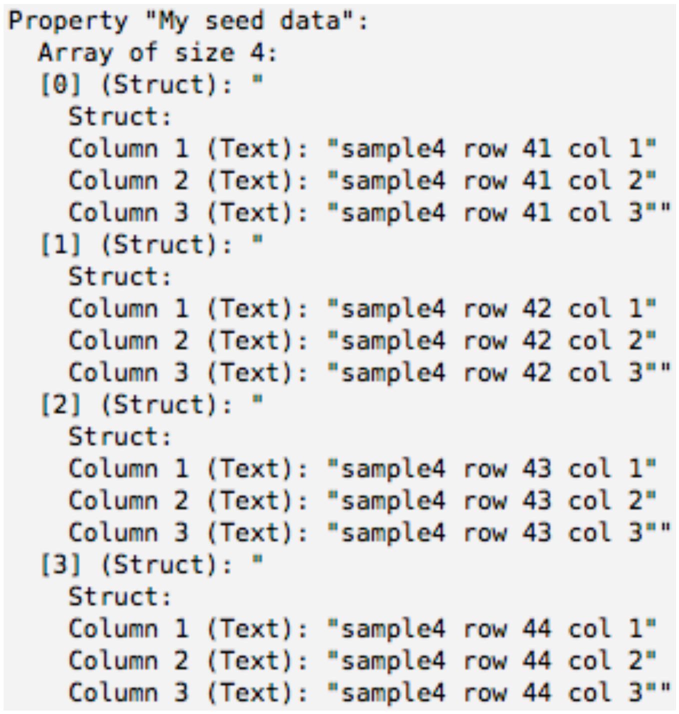
```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 25 col 1"
    Column 2 (Text): "sample4 row 25 col 2"
    Column 3 (Text): "sample4 row 25 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 26 col 1"
    Column 2 (Text): "sample4 row 26 col 2"
    Column 3 (Text): "sample4 row 26 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 27 col 1"
    Column 2 (Text): "sample4 row 27 col 2"
    Column 3 (Text): "sample4 row 27 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 28 col 1"
    Column 2 (Text): "sample4 row 28 col 2"
    Column 3 (Text): "sample4 row 28 col 3""
```

Track: "Track 1 (copy 3 of 3)" [0]



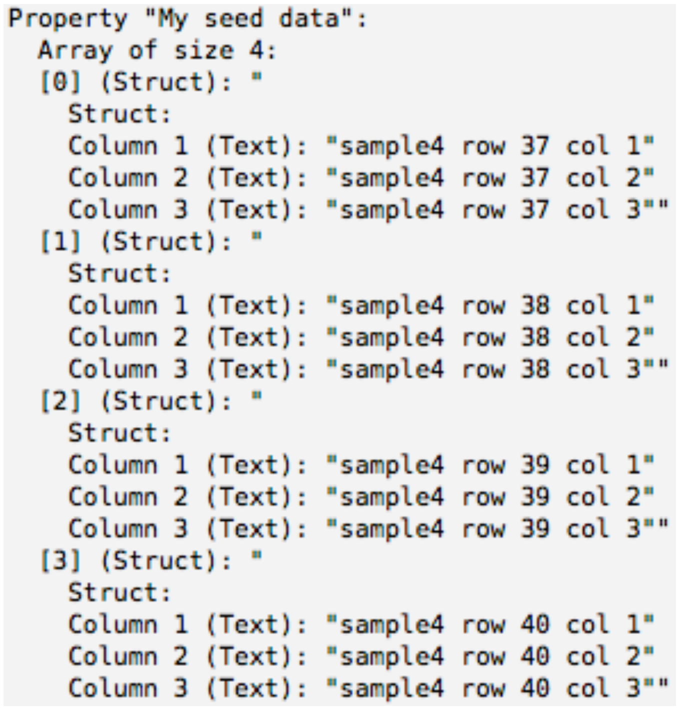
```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 33 col 1"
    Column 2 (Text): "sample4 row 33 col 2"
    Column 3 (Text): "sample4 row 33 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 34 col 1"
    Column 2 (Text): "sample4 row 34 col 2"
    Column 3 (Text): "sample4 row 34 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 35 col 1"
    Column 2 (Text): "sample4 row 35 col 2"
    Column 3 (Text): "sample4 row 35 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 36 col 1"
    Column 2 (Text): "sample4 row 36 col 2"
    Column 3 (Text): "sample4 row 36 col 3""
```

Track: "Track 1 (copy 3 of 3)" [1]



```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 41 col 1"
    Column 2 (Text): "sample4 row 41 col 2"
    Column 3 (Text): "sample4 row 41 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 42 col 1"
    Column 2 (Text): "sample4 row 42 col 2"
    Column 3 (Text): "sample4 row 42 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 43 col 1"
    Column 2 (Text): "sample4 row 43 col 2"
    Column 3 (Text): "sample4 row 43 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 44 col 1"
    Column 2 (Text): "sample4 row 44 col 2"
    Column 3 (Text): "sample4 row 44 col 3""
```

Track: "Track 1 (copy 3 of 3)" [2]



```
Property "My seed data":
  Array of size 4:
  [0] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 37 col 1"
    Column 2 (Text): "sample4 row 37 col 2"
    Column 3 (Text): "sample4 row 37 col 3""
  [1] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 38 col 1"
    Column 2 (Text): "sample4 row 38 col 2"
    Column 3 (Text): "sample4 row 38 col 3""
  [2] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 39 col 1"
    Column 2 (Text): "sample4 row 39 col 2"
    Column 3 (Text): "sample4 row 39 col 3""
  [3] (Struct): "
    Struct:
    Column 1 (Text): "sample4 row 40 col 1"
    Column 2 (Text): "sample4 row 40 col 2"
    Column 3 (Text): "sample4 row 40 col 3""
```