

# So sind die File Index Dateien organisiert

Als interessierter NetWorker Administrator wissen Sie natürlich, wo beim NetWorker Server die File Indexe der einzelnen Clients gespeichert werden. Hierbei ist die Verzeichnis-Struktur immer gleich:

## Unter Linux

<code>/nsr</code>	Der Ursprung der NetWorker Dateiablage
<code>/index</code>	Kennung für den Client File Index Pfad
<code>/clientname</code>	Jeder Client hat sein eigenes Unterverzeichnis
<code>/db6</code>	Kennung für einen Index ab der NW Version 6.0
<code>/####0000</code>	Eine ‚zeitliche Unterteilung‘

Denken Sie daran, daß der NetWorker in der Regel auf einem separaten Volume installiert wird. Bei der Installation wird jedoch ein Link auf `/nsr` erstellt, damit jedes System gleich aussieht.

## Unter Windows

<code>... \nsr</code>	Der Ursprung der NetWorker Dateiablage
<code>\index</code>	Kennung für den Client File Index Pfad
<code>\clientname</code>	Jeder Client hat sein eigenes Unterverzeichnis
<code>\db6</code>	Kennung für einen Index ab der NW Version 6.0
<code>\####0000</code>	Eine ‚zeitliche Unterteilung‘

Im tiefsten Unterverzeichnis befinden sich dann die Client File Index Dateien Hier finden Sie für jede Sicherung im Zustand *browsable* **genau 3 Dateien**, also z.B.,

```
E:\>tree /F E:\nsr\index\19-nwserver.eval.local\db6
Folder PATH listing for volume PROGRAMS
Volume serial number is 78E6-4666
E:\NSR\INDEX\19-NWSERVER.EVAL.LOCAL\DB6
|   v6hdr
|   v6hdr.lck
|   v6journal
|
+---61460000
    6146293d.k0
    6146293d.k1
    6146293d.rec
    6146455d.k0
    6146455d.k1
    6146455d.rec
```

E:\>

Nur die Datei `*.rec` enthält die eigentlichen Daten (die Records). Bei den `*.k*` Dateien handelt es sich um interne Indexe, die allein zur schnelleren Bearbeitung erstellt werden. Sie können allerdings aus den Record-Dateien wieder aufgebaut werden. Das geschieht sogar *on-the-fly*. Deshalb werden beim Backup der Indexe allein die `*.rec` Dateien gesichert.

Woraus besteht jetzt aber die ‚zeitliche Unterteilung‘? - Auch das läßt sich ganz einfach nachvollziehen.

Betrachten wir die Dateien der letzten Seite, dann stellt der geübte Administrator relativ leicht fest, daß die eigentlichen Dateinamen nichts anderes repräsentieren als die hexadezimale Darstellung der Epoch-, also der UNIX/Linux Standardzeit. Bei ihr werden einfach die Sekunden ab dem 01.01.1970 hochgezählt:

```
61460000
  6146293d.k0
  6146293d.k1
  6146293d.rec
  6146455d.k0
  6146455d.k1
  6146455d.rec
```

E:\>

Den Zusammenhang hierzu liefere ich gleich und auf der nächsten Seite.

Die ‚zeitliche Unterteilung‘ oder Gruppierung erfolgt dadurch, daß die letzten 4 Zeichen der Zeit auf ‚0000‘ gesetzt werden. Beim Sichern ermittelt der NetWorker nun, ob es bereits ein solches Verzeichnis mit diesem Namen gibt:

- Wenn ja, werden die neuen File Index Dateien hier abgelegt.
- Wenn nein, erstellt der NetWorker zuerst noch neues Verzeichnis mit diesem Namen.

So vermeidet die Software sehr große, flache Strukturen, denn spätestens nach FFFF Sekunden, also nach 65536 Sekunden erstellt der NetWorker bei einem Backup ein neues Verzeichnis für den File-Index dieses Clients. Das entspricht einem Intervall von genau 18 Stunden.

Den Zusammenhang des Verzeichnisnamens zu einer Sicherung erkennen Sie allerdings erst, wenn Sie sich die lange SSID einer Sicherung ansehen. Hier das Beispiel für die o.a. Dateien:

```
E:\>mminfo -q "name=D:\,savetime>09/01/2021" //
-r "name(10),level,sumsize(10),ssid(64)" -ot
name      lvl      size  ssid
D:\       full  200112092 0ab4221c-00000006-b146293d-6146293d-00500c00-25442829
D:\       incr    6680 f735b550-00000006-b046455d-6146455d-00510c00-25442829
```

E:\>

Das Geheimnis steckt in der 4. Zahlengruppe. Sie repräsentiert nichts anders als den UNIX-Zeitstempel - sie wird im Medienindex in der Spalte *nsavetime* gespeichert.

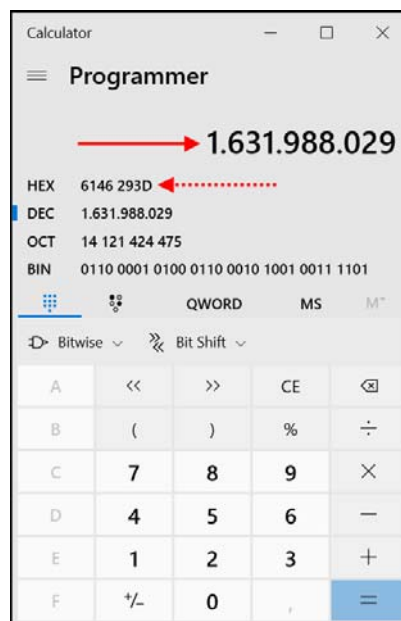
Untersuchen wir zur weiteren Betrachtung den oberen Save Set (den **full**). Dort finden wir die hexadezimale Zahl 6146293d.

Benutzen wir jetzt die lange SSID, um die anderen Zeitstempel abzufragen:

```
E:\>mminfo -q "ssid=0ab4221c-00000006-b146293d-6146293d-00500c00-25442829" //
-r "savetime(25),nsavetime,ssid,cloneid"
date      time      save time  ssid      clone id
18.09.2021 20:00:29 1631988029 2974165309 1631989086

E:>
```

Machen wir die Gegenprobe mit dem Windows Calculator - er bestätigt das Ergebnis:



Jetzt müssen Sie nur noch einen Epoch Rechner bemühen. Ihn finden Sie im Internet z.B. hier:

<https://www.epochconvert.com>

Geben Sie dort diese Zahl ein und klicken Sie auf *Timestamp to Human date*, erscheint dies Ergebnis:

### Convert epoch to human-readable date and vice versa

1631988029 Timestamp to Human date [\[batch convert\]](#)

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **seconds**:

**GMT** : Saturday, September 18, 2021 6:00:29 PM

**Your time zone** : Saturday, September 18, 2021 8:00:29 PM GMT+02:00 DST

**Relative** : 6 days ago

Vergleichen Sie die Zeit mit der o.a. *savetime*.

Allerdings wäre bei diesem Epoch Rechner die Umrechnung in eine Integerzahl gar nicht nötig gewesen - bei ihm können Sie sogar einen hexadezimalen Wert verwenden:

### Convert epoch to human-readable date and vice versa

6146293d 1 Timestamp to Human date 2 [\[batch convert\]](#)

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Converting [hexadecimal timestamp](#) to decimal: 1631988029

Assuming that this timestamp is in **seconds**:

**GMT** : Saturday, September 18, 2021 6:00:29 PM

**Your time zone** : Saturday, September 18, 2021 8:00:29 PM [GMT+02:00 DST](#)

**Relative** : 6 days ago

Vergleichen Sie die Zeit wieder mit der o.a. *savetime*.