

# Tajemnice bconsole czyli pierwsze kroki z tekstową konsolą Bacula cz.1

28 listopad 2010 autor: **gani**

Praca w tekstowej konsoli bconsole może wymagać nieco wprawy szczególnie dla początkujących użytkowników Bacula. Pierwsza część artykułu przedstawia podstawowe komendy wybierania informacji przy pomocy bconsole i kierowana jest właśnie do osób zaczynających poznawanie konsoli Bacula.

Tekstowa konsola bconsole to podstawowe narzędzie administracyjne Bacula. Wyposażona jest w szereg komend, przy pomocy których można dokonywać operacji listowania danych oraz zarządzać innymi serwisami Bacula. Niniejsza pierwsza część artykułu poświęcona jest operacjom wybierania różnego typu informacji z bazy danych.

## Informacje o wykonanych zadaniach

Jako "zadanie" przyjąłem nazywać wszystkie operacje, które możliwe są do zdefiniowania przy pomocy zasobu **Job** (m.in. backup, przywracanie danych, weryfikacja, kopia backupu, zadanie administracyjne, migracja backupu).

Do listowania informacji w bconsole służy komenda **list**. Do wyświetlenia wszystkich zadań do komendy list dodaje się parametr **jobs**:

```
list jobs
```

```
*list jobs
```

JobId	Name	StartTime	Type	Level	JobFiles	JobBytes	JobStatus
1	Katalog Domowy	2010-11-20 17:02:49	B	F	398	7864563	T
2	Poczta	2010-11-20 17:03:24	B	F	3	56095	T
3	Katalog Domowy	2010-11-20 17:03:28	B	I	1	55296	T
4	Poczta	2010-11-20 17:03:34	B	F	3	56805	T
5	Katalog Domowy	2010-11-20 17:03:38	B	I	3	276480	T
6	Poczta	2010-11-20 17:03:45	B	F	3	57515	T
7	Katalog Domowy	2010-11-20 17:03:49	B	I	2	165888	T
8	Poczta	2010-11-20 17:03:56	B	F	3	58225	T
9	Katalog Domowy	2010-11-20 17:03:59	B	I	2	165888	T
10	Poczta	2010-11-20 17:04:06	B	F	3	58935	T
11	Katalog Domowy	2010-11-20 17:04:10	B	I	2	497664	T
12	Poczta	2010-11-20 17:04:17	B	F	3	59645	T
13	Katalog Domowy	2010-11-20 17:04:21	B	I	4	774144	T
14	Poczta	2010-11-20 17:04:28	B	F	3	60355	T
15	Katalog Domowy	2010-11-20 17:04:32	B	I	1	55296	T
16	Poczta	2010-11-20 17:04:39	B	F	3	61065	T
17	Katalog Domowy	2010-11-20 17:04:43	B	I	4	442368	T
18	Poczta	2010-11-20 17:04:50	B	F	3	61775	T
19	Katalog Domowy	2010-11-20 17:04:54	B	I	3	331776	T
20	Poczta	2010-11-20 17:05:01	B	F	3	62485	T
21	Katalog Domowy	2010-11-20 17:05:05	B	I	2	165888	T

*Ilustracja 1: Przykładowe wyjście komendy listowania zadań.*

Powyższy listing zawiera listę 21 zadań. Nie jest to duża ilość i zapewne użytkownik w krótkim czasie jest w stanie ją przekroczyć. Gdy ilość zadań spowoduje, że przy komendzie ich listowania nie będą one już mieścić się na ekranie, to może oczywiście użyć przewijania w górę tekstowej konsoli (zazwyczaj skrót SHIFT+PAGE UP) lub przesuwac suwakiem w emulatorze terminala. Na pewno jest to jakieś rozwiązanie. Co jednak zrobić, gdy liczba wykonanych zadań przekroczy 1000? Wtedy ani nie będą mieścić się one w buforze tekstowej konsoli ani w zakresie suwaka emulatora terminala. Można w tym celu wykorzystać któreś z systemowych narzędzi jak choćby **more** lub **less**. W tym celu w systemowej linii komend (NIE w bconsole) można wydać polecenie jak poniżej:

```
echo "list jobs" | bconsole | more
```

lub

```
echo "list jobs" | bconsole | less
```

Komendy te spowodują wysłanie zapytania o listę zadań do konsoli bconsole, a następnie poprzez standardowe wyjście wynik zapytania zostanie przekierowany do programu zewnętrznej przeglądarki plików (np. more lub less).

Jeśli już wiadomo, że wynik przekazywanych komend do bconsole zwracany jest na standardowe wyjście to można wnioskować, że wyjście to można przekierować do każdego innego programu, który operuje na standardowym wyjściu (grep, awk, head, tail itp.), czy też przekierować wyjście bconsole do pliku. Co więcej, tego typu wywołanie konsoli Bacula można użyć do budowania własnych skryptów.

Na koniec dodam, że wywołanie te nie ogranicza się do listowania zadań poprzez komendę **list jobs** lecz można w nim wykorzystać dowolną komendę bconsole.

Aby ograniczyć ilość wyświetlanych zadań z komendy **list jobs** do jednego konkretnego zadania można użyć następującego wywołania w konsoli Bacula:

```
list jobid=13
```

```
*list jobid=13
+-----+-----+-----+-----+-----+-----+-----+-----+
| JobId | Name           | StartTime           | Type | Level | JobFiles | JobBytes | JobStatus |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 13    | Katalog Domowy | 2010-11-20 17:04:21 | B    | I     | 4        | 774144   | T         |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

**Ilustracja 2:** Listing wybranego zadania.

Uzyskana powyżej lista wykonanych zadań poprzez komendę **list jobs** dostarcza kilka ważniejszych informacji o zadaniach. Nie są to jednak wszystkie informacje, jakie można uzyskać z konsoli bconsole. Do wyświetlenia bardziej szczegółowych informacji o wybranym zadaniu służy komenda:

```
l1list jobid=13
```

gdzie jako wartość jobid trzeba wstawić identyfikator żadanego zadania (tutaj wstawiłem identyfikator o wartości 13).

```
*l1ist jobid=13
  JobId: 13
    Job: Katalog_Domowy.2010-11-20_17.04.19_28
    Name: Katalog Domowy
  PurgedFiles: 0
    Type: B
    Level: I
  ClientId: 1
    Name: hardstar-fd
  JobStatus: T
  SchedTime: 2010-11-20 17:04:19
  StartTime: 2010-11-20 17:04:21
  EndTime: 2010-11-20 17:04:22
  RealEndTime: 2010-11-20 17:04:22
  JobTDate: 1290269062
  VolSessionId: 13
  VolSessionTime: 1290268929
  JobFiles: 4
  JobErrors: 0
  JobMissingFiles: 0
  PoolId: 1
  PoolName: Default
  PriorJobId: 0
  FileSetId: 1
  FileSet: domowy
```

**Ilustracja 3:** Szczegółowy listing wybranego zadania.

Można również użyć komendy **l1ist** z parametrem **jobs** np:

```
l1ist jobs
```

```
JobId: 20
Job: Poczta.2010-11-20_17.04.55_46
Name: Poczta
PurgedFiles: 0
Type: B
Level: F
ClientId: 1
ClientName: hardstar-fd
JobStatus: T
SchedTime: 2010-11-20 17:04:55
StartTime: 2010-11-20 17:05:01
EndTime: 2010-11-20 17:05:02
RealEndTime: 2010-11-20 17:05:02
JobTDate: 1290269102
VolSessionId: 20
VolSessionTime: 1290268929
JobFiles: 3
JobErrors: 0
JobMissingFiles: 0
PoolId: 2
PoolName: File
PriorJobId: 0
FileSetId: 2
FileSet: poczta

JobId: 21
Job: Katalog_Domowy.2010-11-20_17.05.03_58
Name: Katalog Domowy
PurgedFiles: 0
Type: B
Level: I
ClientId: 1
ClientName: hardstar-fd
JobStatus: T
SchedTime: 2010-11-20 17:05:03
StartTime: 2010-11-20 17:05:05
EndTime: 2010-11-20 17:05:06
RealEndTime: 2010-11-20 17:05:06
JobTDate: 1290269106
VolSessionId: 21
VolSessionTime: 1290268929
JobFiles: 2
JobErrors: 0
JobMissingFiles: 0
PoolId: 1
PoolName: Default
PriorJobId: 0
FileSetId: 1
FileSet: domowy
```

**Ilustracja 4:** Szczegółowe informacje o wszystkich zadaniach.

lecz wtedy na ekranie zostaną wyświetlone szczegółowe informacje wszystkich wykonanych zadań, co przy kilkunastu zadaniach daje sporej długości listing. Nie jest to ani czytelny ani wygodny listing do analizowania. Z tego powodu polecam używać **llist jobid=ID** zamiast **llist jobs**.

Ostatnia komenda wyciągająca informacje o zadaniach jest swego rodzaju podsumowaniem ilości wykonanych zadań, ilości zapisanych plików i ilości zapisanych bajtów dla każdego zasobu typu **Job**. Dodatkowo wszystkie te wartości są zsumowane i podane jako całkowita ilość wykonanych zadań, plików i bajtów. Komenda wyciągająca te informacje z bazy danych to:

```
list jobtotals
```

```
*list jobtotals
+-----+-----+-----+-----+
| Jobs | Files | Bytes   | Job          |
+-----+-----+-----+-----+
| 11   | 422   | 10795251 | Katalog Domowy |
| 10   | 30    | 592900   | Poczta         |
+-----+-----+-----+-----+

+-----+-----+-----+-----+
| Jobs | Files | Bytes   |
+-----+-----+-----+-----+
| 21   | 452   | 11388151 |
+-----+-----+-----+-----+
```

*Ilustracja 5: Podsumowanie wykonanych zadań.*

## Informacje o woluminach

Kolejny zestaw komend, jakie zostaną przedstawione, służy do wybierania informacji o woluminach. Również tutaj użyta zostanie komenda **list**. Tym razem przełącznikiem będzie **media** lub **volumes** w następujący sposób:

```
list media
```

lub

```
list volumes
```

```
*list media
Pool: Default
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| MediaId | VolumeName          | VolStatus | Enabled | VolBytes | VolFiles | VolRetention | Recycle | Slot | InChanger | MediaType | LastWritten |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 3       | Plik-incremental-000 | Append   | 1       | 3107413 | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:05:06 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Pool: File
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| MediaId | VolumeName          | VolStatus | Enabled | VolBytes | VolFiles | VolRetention | Recycle | Slot | InChanger | MediaType | LastWritten |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1       | Plik-full-000      | Used     | 1       | 23463064 | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:03:17 |
| 2       | Plik-full-001      | Used     | 1       | 113849   | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:03:24 |
| 4       | Plik-full-002      | Used     | 1       | 114559   | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:03:35 |
| 5       | Plik-full-003      | Used     | 1       | 115269   | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:03:46 |
| 6       | Plik-full-004      | Used     | 1       | 115979   | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:03:56 |
| 7       | Plik-full-005      | Used     | 1       | 116689   | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:04:06 |
| 8       | Plik-full-006      | Used     | 1       | 60575    | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:04:17 |
| 9       | Plik-full-007      | Used     | 1       | 61285    | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:04:29 |
| 10      | Plik-full-008      | Used     | 1       | 61995    | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:04:40 |
| 11      | Plik-full-009      | Used     | 1       | 62705    | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:04:51 |
| 12      | Plik-full-010      | Used     | 1       | 63415    | 0         | 31536000    | 1       | 0    | 0         | File      | 2010-11-20 17:05:02 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Pool: Scratch
No results to list.
```

*Ilustracja 6: Lista wszystkich woluminów.*

Parametry **media** i **volumes** w połączeniu z komendą **list** znaczą to samo i można ich używać zamiennie. Z tego powodu nie ma znaczenia którego przełącznika użyjemy, gdyż otrzymany listing będzie dokładnie taki sam.

Podobnie jak w przypadku listy zadań, tak i tutaj istnieje możliwość wylistowania informacji o jednym tylko woluminie poprzez:

```
list media=nazwa_woluminu
```

lub

```
list volume=nazwa_woluminu
```

```
*list volume=Plik-full-003
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| MediaId | VolumeName | VolStatus | Enabled | VolBytes | VolFiles | VolRetention | Recycle | Slot | InChanger | MediaType | LastWritten |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 5       | Plik-full-003 | Used      | 1       | 115289   | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:03:46 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

**Ilustracja 7:** Listing wybranego wolumina.

Również parametry **media** i **volume** mogą występować zamiennie podobnie jak w przypadku listy wszystkich woluminów.

Aby filtrować wyniki komendy **list media** (**list volumes**) do wyświetlenia woluminów określonej puli woluminów, trzeba podać dodatkowy parametr **pool**, w którym podaje się nazwę puli, której woluminy chce się wyświetlić. Np.:

```
list media pool=nazwa_puli
```

lub

```
list volumes pool=nazwa_puli
```

```
*list media pool=File
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| MediaId | VolumeName | VolStatus | Enabled | VolBytes | VolFiles | VolRetention | Recycle | Slot | InChanger | MediaType | LastWritten |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1       | Plik-full-000 | Used      | 1       | 23463364 | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:03:17 |
| 2       | Plik-full-001 | Used      | 1       | 113848   | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:03:24 |
| 4       | Plik-full-002 | Used      | 1       | 114558   | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:03:35 |
| 5       | Plik-full-003 | Used      | 1       | 115289   | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:03:46 |
| 6       | Plik-full-004 | Used      | 1       | 115979   | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:03:56 |
| 7       | Plik-full-005 | Used      | 1       | 116689   | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:04:06 |
| 8       | Plik-full-006 | Used      | 1       | 60675    | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:04:17 |
| 9       | Plik-full-007 | Used      | 1       | 61285    | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:04:29 |
| 10      | Plik-full-008 | Used      | 1       | 61995    | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:04:40 |
| 11      | Plik-full-009 | Used      | 1       | 62705    | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:04:51 |
| 12      | Plik-full-010 | Used      | 1       | 63415    | 0        | 31536000    | 1       | 0   | 0         | File      | 2010-11-20 17:05:02 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

**Ilustracja 8:** Ograniczona do jednej puli lista woluminów.

Aby uzyskać szczegółowe informacje o jednym tylko woluminie, można posłużyć się komendą **llist** w następujący sposób:

```
llist media=nazwa_woluminu
```

lub

```
llist volume=nazwa_woluminu
```

```
*l1ist volume=Plik-full-003
  MediaId: 5
  VolumeName: Plik-full-003
  Slot: 0
  PoolId: 2
  MediaType: File
  FirstWritten: 2010-11-20 17:03:45
  LastWritten: 2010-11-20 17:03:46
  LabelDate: 2010-11-20 17:03:45
  VolJobs: 1
  VolFiles: 0
  VolBlocks: 1
  VolMounts: 1
  VolBytes: 115269
  VolErrors: 0
  VolWrites: 1
VolCapacityBytes: 0
  VolStatus: Used
  Enabled: 1
  Recycle: 1
  VolRetention: 31536000
VolUseDuration: 0
  MaxVolJobs: 1
  MaxVolFiles: 0
  MaxVolBytes: 0
  InChanger: 0
  EndFile: 0
  EndBlock: 115268
  VolParts: 0
  LabelType: 0
  StorageId: 1
  DeviceId: 0
  LocationId: 0
  RecycleCount: 0
  InitialWrite: 0
  ScratchPoolId: 0
  RecyclePoolId: 0
  Comment: NULL
```

**Ilustracja 9:** Szczegółowe informacje o wybranym woluminie.

W komendzie **l1ist volumes** (lub **l1ist media**) można też ograniczać wynik do określonej puli woluminów poprzez:

```
l1ist volumes pool=nazwa_puli
```

lub

```
l1ist media pool=nazwa_puli
```

## Informacje o pulach woluminów

Część informacji o pulach woluminów można było zauważyć w poprzednim podrozdziale, gdzie komenda `list media` (lub `list volumes`) pokazała zawartość puli woluminów czyli informacje o tym, jakie woluminy znajdują się w jakich pulach.

Komenda **list pools** daje możliwość wylistowania nieco więcej informacji na temat puli woluminów.

```
list pools
```

```
*list pools
+-----+-----+-----+-----+-----+-----+
| PoolId | Name   | NumVols | MaxVols | PoolType | LabelFormat |
+-----+-----+-----+-----+-----+-----+
| 1      | Default | 1       | 0       | Backup   | Plik-incremental-  
| 2      | File   | 11      | 100     | Backup   | Plik-full-  
| 3      | Scratch | 0       | 0       | Backup   | *
+-----+-----+-----+-----+-----+-----+
```

*Ilustracja 10:* Lista puli woluminów.

Komenda **llist pool=nazwa\_puli** udostępnia szerszych informacji na temat jednej konkretnej puli woluminów.

```
llist pool=nazwa_puli
```

```
*llist pool=File
    PoolId: 2
    Name: File
    NumVols: 11
    MaxVols: 100
    UseOnce: 0
    UseCatalog: 1
    AcceptAnyVolume: 0
    VolRetention: 31536000
    VolUseDuration: 0
    MaxVolJobs: 1
    MaxVolBytes: 0
    AutoPrune: 1
    Recycle: 1
    PoolType: Backup
    LabelFormat: Plik-full-  
    Enabled: 1
    ScratchPoolId: 0
    RecyclePoolId: 0
    LabelType: 0
```

*Ilustracja 11:* Szczegółowe informacje wybranej puli woluminów.

Tak jak w przypadku szczegółowego listowania informacji o zadaniu, tak i tutaj można wywołać komendę **llist** z parametrem **pools** lecz bez przypisanej do niej wartości. Wynikiem będzie uzyskanie szczegółowych informacji o wszystkich pulach woluminów.

```
llist pools
```



```

MaxVolJobs: 0
MaxVolBytes: 0
  AutoPrune: 1
    Recycle: 1
  PoolType: Backup
  LabelFormat: Plik-incremental- $\{\text{Kanter}+\}$ :p/3/0/r}
  Enabled: 1
ScratchPoolId: 0
RecyclePoolId: 0
  LabelType: 0

  PoolId: 2
    Name: File
    NumVols: 11
    MaxVols: 100
    UseOnce: 0
    UseCatalog: 1
AcceptAnyVolume: 0
  VolRetention: 31536000
  VolUseDuration: 0
    MaxVolJobs: 1
    MaxVolBytes: 0
    AutoPrune: 1
    Recycle: 1
    PoolType: Backup
    LabelFormat: Plik-full- $\{\text{Kanter}2+\}$ :p/3/0/r}
    Enabled: 1
ScratchPoolId: 0
RecyclePoolId: 0
  LabelType: 0

  PoolId: 3
    Name: Scratch
    NumVols: 0
    MaxVols: 0
    UseOnce: 0
    UseCatalog: 1
AcceptAnyVolume: 0
  VolRetention: 31536000
  VolUseDuration: 0
    MaxVolJobs: 0
    MaxVolBytes: 0
    AutoPrune: 1
    Recycle: 1
    PoolType: Backup
  LabelFormat: *
  Enabled: 1
ScratchPoolId: 0
RecyclePoolId: 0
  LabelType: 0

```

**Ilustracja 12:** Szczegółowe informacje o wszystkich pulach woluminów.

## Lista plików zadania

W bazie danych Bacula, oprócz informacji o wykonanych zadaniach, woluminach czy pulach woluminów, przetrzymywane są również wszystkie nazwy zapisywanych plików wraz z ich lokalizacjami. Je również można wylistować w tekstowej konsoli Bacula, a służy do tego komenda:

```
list files jobid=identyfikator_zadania
```

```
*list files jobid=2
+-----+
| Filename |
+-----+
| /var/spool/mail/ |
| /var/spool/mail/rrrr |
| /var/spool/mail/root |
+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| JobId | Name | StartTime | Type | Level | JobFiles | JobBytes | JobStatus |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | Poczta | 2010-11-20 17:03:24 | B | F | 3 | 56095 | T |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

**Ilustracja 13:** Lista plików wybranego zadania.

Jako, że jedno zadanie niejednokrotnie może zawierać setki, tysiące, setki tysięcy czy więcej plików, warto zapisać je bezpośrednio do pliku lub użyć przekierowania standardowego wyjścia bconsole z systemowej linii komend do innego programu. Jak tego dokonać, zaprezentowałem przy okazji omawiania listowania zadań (**list jobs**). Tutaj przedstawię jak zapisywać listingi wykonanych komend do pliku wprost z tekstowej konsoli Bacula.

Do przekierowania wyjścia wyników komend służy komenda:

```
@output lokalizacja_pliku
```

np.

```
@output /tmp/pliki.txt
```

Po wywołaniu tej komendy, wynik każdej następnej komendy trafi do zdefiniowanego pliku (w tym wypadku to pliki.txt).

Aby wyłączyć "zrzucanie" wyników komend do pliku, wystarczy wywołać **@output** bez parametru, np:

```
@output
```

Zrzucenie listy zapisanych plików zadania o identyfikatorze 12 do pliku *pliki.txt* może wyglądać następująco:

```
@output /tmp/pliki.txt
list files jobid=12
@output
```

## Podsumowanie

To tyle jeśli chodzi o wprowadzenie do wybierania informacji z bazy danych przy użyciu tekstowej konsoli Bacula lub poprzez systemową linię komend. W części drugiej artykułu zaprezentuję w jaki sposób wybierać dane z bazy danych przy użyciu zapytań SQL oraz jak tworzyć własne zestawy zapytań SQL.