Powiększanie baz i urządzeń

Typ: procedura **Dotyczy: Sybase ASE 15.7 / 16.0 (SAP ASE)** **Czas: ~30 min**

Efekt: baza danych ma więcej miejsca do zapisu, uzyskanego z device'a, który sam wcześniej został powiększony na poziomie systemu plików.

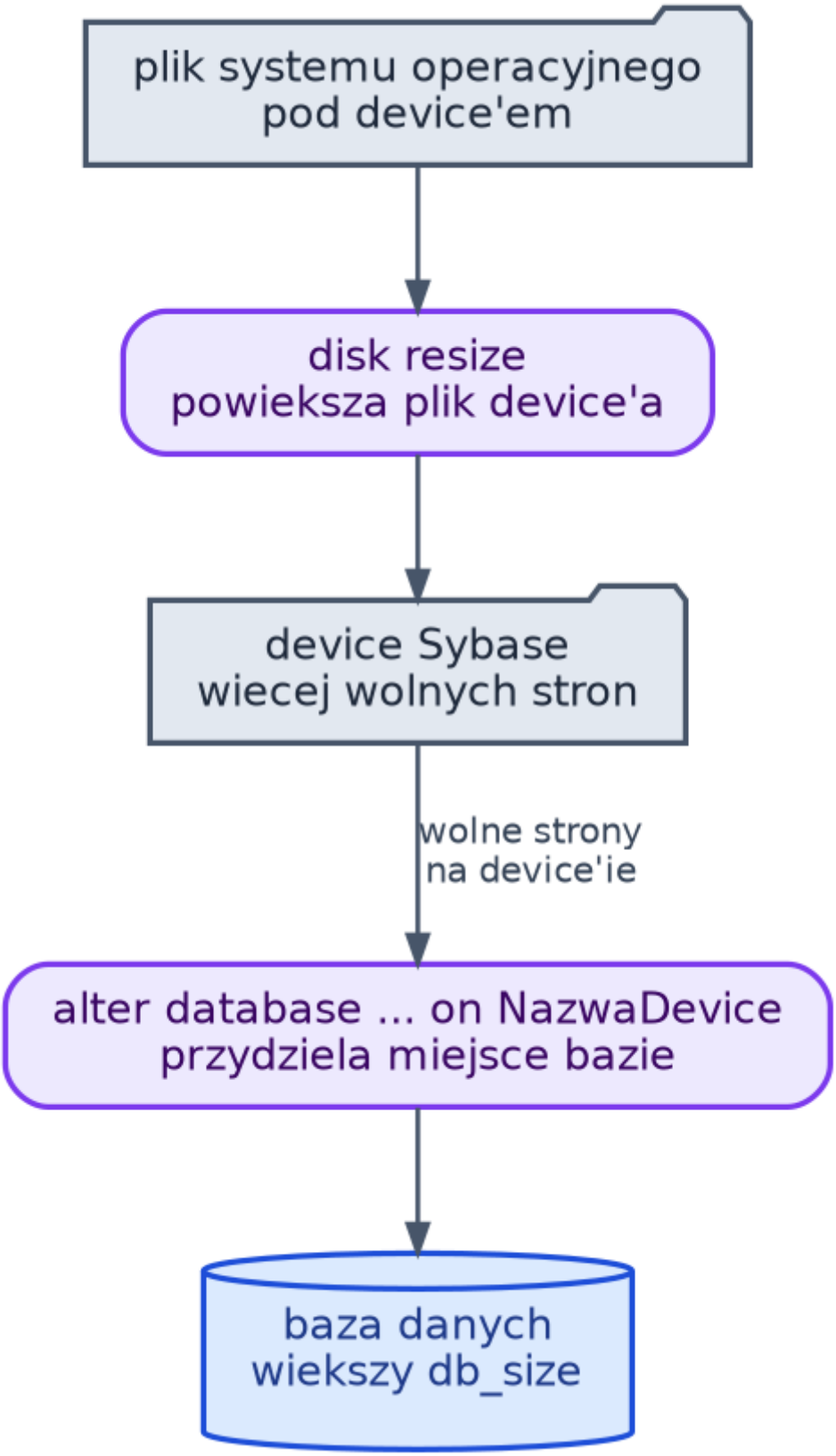
Skróty użyte na tej stronie (2)

Skrót	Co znaczy
ASE	Adaptive Server Enterprise: nazwa silnika Sybase od 1996 roku, dziś rozwijanego przez SAP jako SAP ASE
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE

Zanim zaczniemy

> **Z ŻYCIA WZIĘTE**
>
> Archiwum płac PLACE_ARCH na syb01 rośnie po każdym roku obrachunkowym, kiedy dział kadr wgrywa kolejny pakiet danych historycznych. Sprawdzamy sp_helpdevice i widzimy, że device dataProdukcja1 ma zaledwie kilkaset megabajtów wolnego miejsca, za mało na tegoroczny import. Powiększamy najpierw sam plik device'a poleceniem disk resize, a dopiero potem przydzielamy nowe miejsce bazie poleceniem alter database, bo pomylenie kolejności tych dwóch kroków nic by nie dało. Import danych z tego roku kończy się bez błędu o pełnym segmencie.

Powiększenie bazy w Sybase to zawsze dwuetapowa operacja: najpierw device musi mieć fizycznie więcej miejsca (albo trzeba dołożyć nowy device), dopiero potem to miejsce przydzielamy konkretnej bazie danych poleceniem `alter database`. Pojęcia device, baza i segment są opisane na stronie Architektura Sybase ASE: serwer, device, baza, segment — bez nich łatwo pomylić kolejność kroków poniżej.



Disk resize powieksza plik device'a, dopiero alter database przydziela to miejsce bazie (kliknięcie otwiera pełny rozmiar)

Wymaganie	Jak sprawdzić
Wolne miejsce na dysku pod plikiem device'a	`df -h` na systemie operacyjnym, w katalogu z plikiem `.dat`
Aktualny rozmiar device'a i bazy	`sp_helpdevice` i `sp_helpdb NazwaBazy`
Backup bazy master wykonany niedawno	zmiana rozmiaru bazy jest zapisywana w `master`

Uważaj: `disk resize` powiększa plik device'a na dysku i nie da się go łatwo zmniejszyć z powrotem — zbyt duża wartość rezerwuje miejsce na systemie plików, którego przez długi czas nikt nie wykorzysta. Operacje na bazie `master` wymagają szczególnej ostrożności, bo błąd w tej bazie może uniemożliwić start całego serwera.

Przebieg

1. Sprawdzenie aktualnej wielkości plików device

```
sp_helpdevice
go
```

```
 device_name     physical_name                                 description             size
free_kbytes
 --------------- ------------------------------------------- ---------------------- -------
-----------
 dataProdukcja1   /opt/sybase/produkcja/data1/dataProd1.dat  physical disk           326000
1200
 dataProdukcja2   /opt/sybase/produkcja/data2/dataProd2.dat  physical disk           100166
860

(2 rows affected)
```

Notujemy aktualny rozmiar (`size`, w kilobajtach) każdego device'a, który zamierzamy powiększyć — będzie potrzebny do wyliczenia nowej wartości w kroku 3.

2. Sprawdzenie ostatnich powiększeń i rozłożenia bazy po device'ach

```
sp_helpdb NazwaBazy
go
```

```
 name       db_size     owner  dbid  created      status
 ---------- ----------- ------ ----- ------------ -------------------
 produkcja  426166.0 MB sa     6     Mar 18 2021  trunc log on chkpt

 device_fragments      size        usage           free kbytes
 --------------------- ----------- --------------- -----------
 dataProdukcja1         326000.0 MB data only       1200
 dataProdukcja2         100166.0 MB data only       860

(2 rows affected)
```

Widzimy, na ilu i jakich device'ach rozłożona jest baza — to decyduje, które z nich będziemy powiększać w kroku 3, żeby przydzielić bazie nową przestrzeń.

3. Powiększenie plików urządzeń (device) na poziomie systemu plików

```
use master
go
disk resize name = 'NazwaDevice', size = 'RozmiarMB'
go
```

> (gdzie `RozmiarMB` to docelowy, całkowity rozmiar device'a po powiększeniu w megabajtach, nie sama wielkość przyrostu — Sybase liczy tę wartość jako nową sumę, nie jako dodatek do obecnej)

Powiększamy fizyczny plik device'a na systemie plików do nowej, wskazanej wielkości. Operację powtarzamy dla każdego device'a z osobna, jednym poleceniem na jeden device.

```
Device 'dataProdukcja1' resized to 400000000 kilobytes.
```

Komunikat potwierdza nowy, całkowity rozmiar w kilobajtach, nie przyrost — dobra chwila, żeby zestawić go z wartością odnotowaną w kroku 1.

4. Powiększenie bazy danych o nowo dostępne miejsce na device

```
use master
go
alter database NazwaBazy on NazwaDevice = 'RozmiarMB'
go
```

> (gdzie `RozmiarMB` w tym poleceniu to wielkość przyrostu, który baza ma dostać z tego device'a — w odróżnieniu od `disk resize`, tutaj podajemy o ile ma być więcej, nie docelowy rozmiar całkowity)

Przydzielamy bazie danych dodatkowe miejsce zwolnione na device'ie w kroku 3. To dopiero ten krok faktycznie zwiększa `db_size` widoczny w `sp_helpdb` — samo powiększenie device'a w kroku 3 tylko przygotowało dla niego miejsce.

```
Extending database by 102400 pages (800.0 megabytes) on disk dataProdukcja1
Allocating 800 allocation unit(s) out of 800 units (allocation page 44200192).
Allocating 800 allocation unit(s) out of 800 units (allocation page 44220672).
100 percent complete.
```

Przy większych przyrostach polecenie może przydzielać przestrzeń partiami po kilkaset jednostek alokacji naraz i wypisywać kolejne linie postępu, zanim zwróci komunikat końcowy.

Gdy powiększamy bazę na kilku device'ach naraz, powtarzamy `alter database` dla każdego z nich osobno:

```
use master
go
alter database NazwaBazy on NazwaDevice1 = 'RozmiarMB'
alter database NazwaBazy on NazwaDevice2 = 'RozmiarMB'
go
```

Weryfikacja

Sprawdzamy nowy rozmiar bazy i jej rozłożenie po device'ach:

```
sp_helpdb NazwaBazy
go
```

```
 name       db_size     owner  dbid  created      status
 ---------- ----------- ------ ----- ------------ -------------------

 produkcja  426966.0 MB sa     6     Mar 18 2021  trunc log on chkpt


 device_fragments      size        usage           free kbytes
 --------------------- ----------- --------------- -----------
 dataProdukcja1         326800.0 MB data only       814800
 dataProdukcja2         100166.0 MB data only       860


(2 rows affected)
```

Sprawdzamy też wolne miejsce zapytaniem z osobnej strony Rozpoznanie serwera i baz, żeby potwierdzić, że nowa przestrzeń jest realnie dostępna do zapisu, a nie tylko widoczna w statystykach.

Wycofanie zmian

Sybase nie udostępnia polecenia zmniejszającego bazę danych ani device w miejscu — raz przydzielona przestrzeń zostaje przypisana do bazy na stałe. Jedyna droga odzyskania miejsca to eksport danych, założenie nowej, mniejszej bazy i przeniesienie do niej danych (`bcp` albo `dump`/`load` na nowy zestaw device'ów), co jest operacją znacznie poważniejszą niż samo powiększenie i wymaga osobnego okna serwisowego.

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
`disk resize` kończy się błędem o braku miejsca	system plików pod device'em faktycznie nie ma wolnej przestrzeni	powiększenie wolumenu/dysku pod spodem, dopiero potem ponowna próba
`alter database` zwraca błąd, że device nie ma wystarczająco wolnego miejsca	krok 3 (`disk resize`) nie został wykonany albo powiększył za mało	ponowne `sp_helpdevice`, weryfikacja `free_kbytes` przed ponowieniem `alter database`
Operacja trwa bardzo długo na dużych przyrostach	silnik zeruje i inicjalizuje nowo przydzielone strony	operację wykonujemy w oknie serwisowym, nie w godzinach szczytu

Objaw	Przyczyna	Naprawa
`sp_helpdb` po operacji nie pokazuje nowego rozmiaru	zmiany wykonane w innej bazie niż zamierzona (brak `use master` przed `alter database`)	sprawdzenie bieżącego kontekstu (`select db_name()`), powtórzenie z właściwego kontekstu

Powiązane: Architektura Sybase ASE: serwer, device, baza, segment · Rozpoznanie serwera i baz · Rozmiar bazy: pliki, tablespace'y, autoextend

Źródła: notatki własne administratora Sybase ASE · dokumentacja SAP ASE, rozdział „System Administration Guide" (`disk resize`, `alter database`)

Revision #43
Created 2025-03-09 16:28:52 UTC by Tomasz Konieczny
Updated 2026-10-26 14:08:09 UTC by Tomasz Konieczny