

WME – Instalare PostgreSQL 18 și WinMENTOR Enterprise pe PostgreSQL

Documentul descrie instalarea serverului PostgreSQL 18 pe Windows, a componentelor suplimentare de care are nevoie WinMENTOR Enterprise (pgAgent, extensia HTTP), configurarea serverului (postgresql.conf, pg_hba.conf) și instalarea WinMENTOR Enterprise pe PostgreSQL, pe server și pe stațiile client. Documentul înlocuiește instrucțiunile de instalare pentru PostgreSQL 17. (Rev. 2.0)

I. KITUL DE INSTALARE

Kitul de instalare se descarcă de la adresa de mai jos și este organizat pe directoare, în ordinea pașilor de instalare: (Rev. 2.0)

http://download.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre_Instalare/PostGreSQL%20v18/

Director	Conținut
01.Kit PostgreSQL	Kitul oficial PostgreSQL 18 pentru Windows x64 (EnterpriseDB). Se folosește ultima versiune 18.x de pe www.enterprisedb.com/downloads/postgres-postgresql-downloads .
02.DLL-ri pentru clientul de postgres	DLL-urile de client PostgreSQL pentru WinMENTOR Enterprise, pe 32 și pe 64 de biți (capitolul IX).
03.Fisiere config postgres	postgresql.conf, pg_hba.conf și un exemplu de dbparams.ini (capitolele V, VI, IX).
04.Installer WME -PG	WMInstaller_PG, arhivele PGMARTOR / PGSERVICE, WinMSservice, executabilele WinMENTOR Enterprise, Nethasp.ini, PROTECT.DAT, dbparams.ini, AnafConnector, chkVIES.
05.DBeaver - pentru interogare BD	DBeaver Community, pentru rularea comenzilor SQL din acest document.
06.Extensii	Extensia HTTP compilată pentru PostgreSQL 18 (pg18http_w64.zip) și WM_PGTT

Ce se schimbă față de PostgreSQL 17 (Rev. 2.0)

- **Extensia HTTP** are o arhivă nouă, compilată pentru PostgreSQL 18 (pg18http_w64.zip). Arhiva pentru versiunea 17 nu funcționează pe versiunea 18 (capitolul VII). (Rev. 2.0)
- **maintenance_work_mem** nu mai este limitat la 2047MB pe Windows; valoarea propusă de PG Tune se păstrează (capitolul V). (Rev. 2.0)
- **Autentificarea MD5 este declarată învechită** de PostgreSQL 18; se recomandă metoda scram-sha-256 în pg_hba.conf (capitolul VI). (Rev. 2.0)

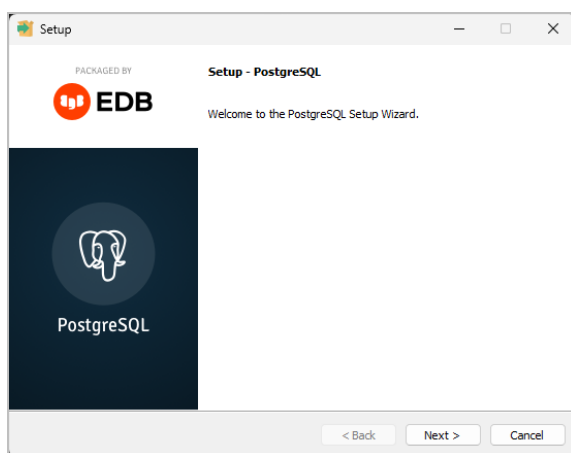
- **Sumele de control ale paginilor (data checksums)** sunt active implicit la crearea clusterului de date. (Rev. 2.0)
- **Extensia WM_PGTT** (opțională), care accelerează conectarea în WinMENTOR Enterprise, se livrează doar pentru PostgreSQL 18 (capitolul VIII). (Rev. 2.0)
- Fișierul postgresql.conf din kit este compatibil cu PostgreSQL 18 (a fost verificat cu serverul 18). (Rev. 2.0)

Obs! Capturile de ecran din capitolele II și III sunt din instalarea versiunii 17. Ferestrele kitului de instalare sunt aceleași la versiunea 18; în căi și în denumiri apare 18 în loc de 17 (de exemplu D:\PostgreSQL\18, postgresql-x64-18). (Rev. 2.0)

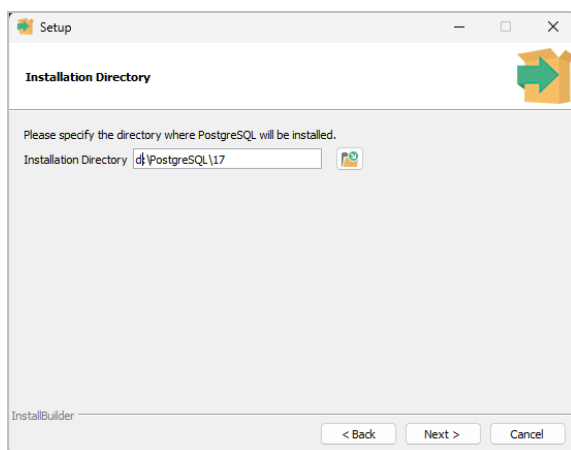
Obs! La trecerea unui server existent de pe PostgreSQL 17 pe 18 cu utilitarul pg_upgrade, ambele clusteruri trebuie să aibă aceeași setare a sumelor de control. Clusterul 17 creat fără sume de control (cazul implicit la versiunea 17) se actualizează doar într-un cluster 18 creat cu initdb --no-data-checksums. Extensiile HTTP și WM_PGTT se instalează din nou, din kitul pentru versiunea 18. (Rev. 2.0)

II. INSTALAREA SERVERULUI POSTGRESQL 18

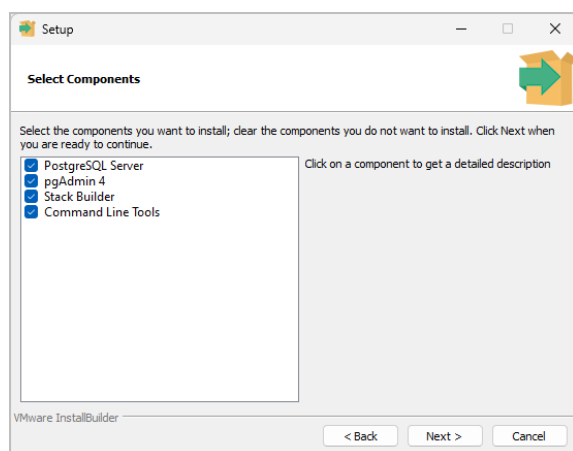
1. Se rulează kitul PostgreSQL 18 din directorul 01.Kit PostgreSQL, ca Administrator.



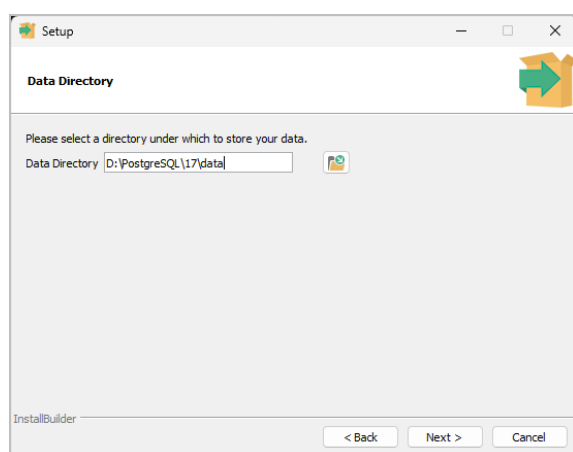
2. Se alege calea în care se instalează serverul PostgreSQL. Implicit este C:\Program Files, dar este de preferat o altă partiție decât C:\, din motive legate de drepturile de acces și de reinstalările de Windows (de exemplu D:\PostgreSQL\18).



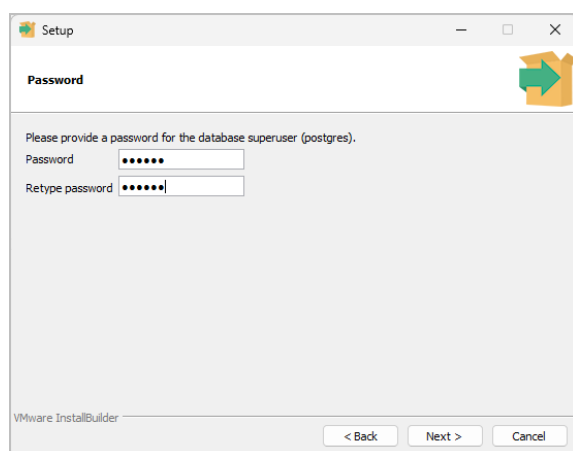
3. Se bifează toate componentele: PostgreSQL Server, pgAdmin 4, Stack Builder, Command Line Tools.



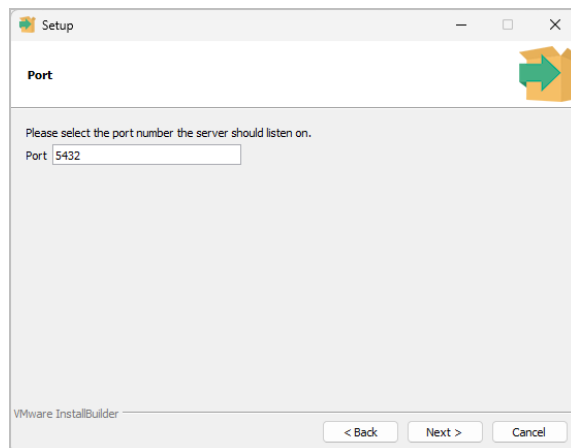
4. Se alege directorul bazei de date. Implicit este tot în C:\Program Files; este de preferat tot o altă partiție decât C:\ (de exemplu D:\PostgreSQL\18\data).



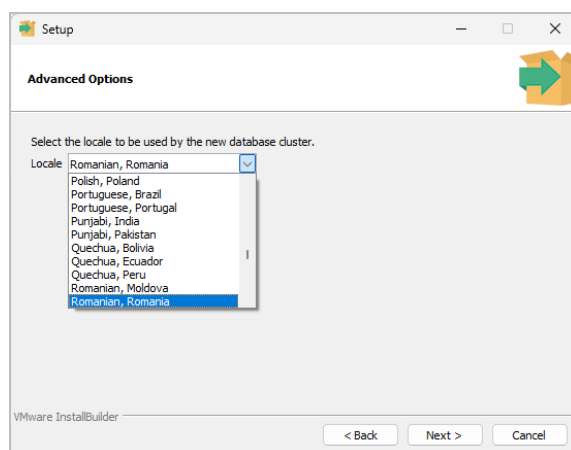
5. Se alege parola administratorului bazei de date (utilizatorul implicit: **postgres**). Parola se păstrează: este necesară la instalarea pgAgent, a extensiilor și a WinMENTOR Enterprise.



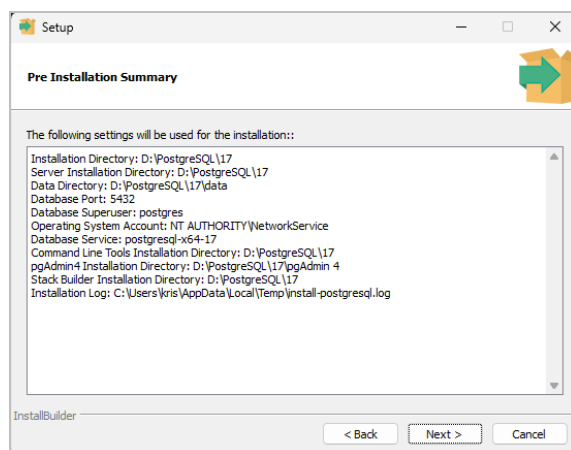
6. Portul serverului: implicit 5432.



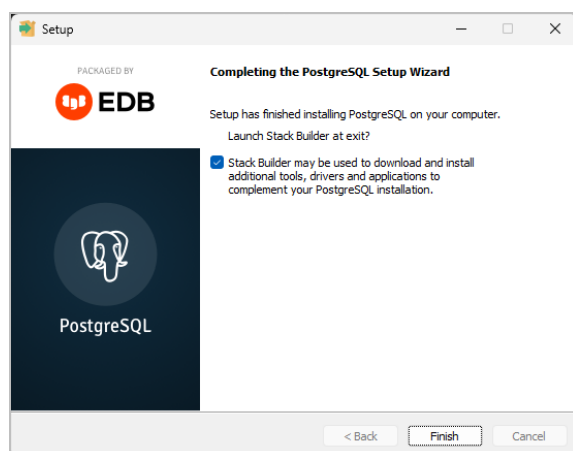
7. Se aleg setările regionale ale clusterului de date: Romanian, Romania.



8. Se verifică sumarul instalării și se continuă cu Next.



9. La final se lasă bifată opțiunea de pornire a Stack Builder, prin care se instalează pgAgent (capitolul III).

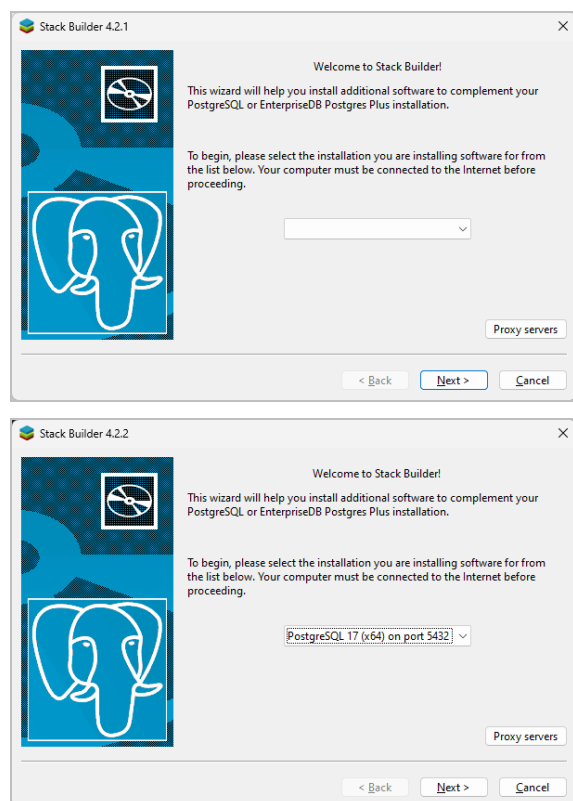


III. INSTALAREA PGAGENT (STACK BUILDER)

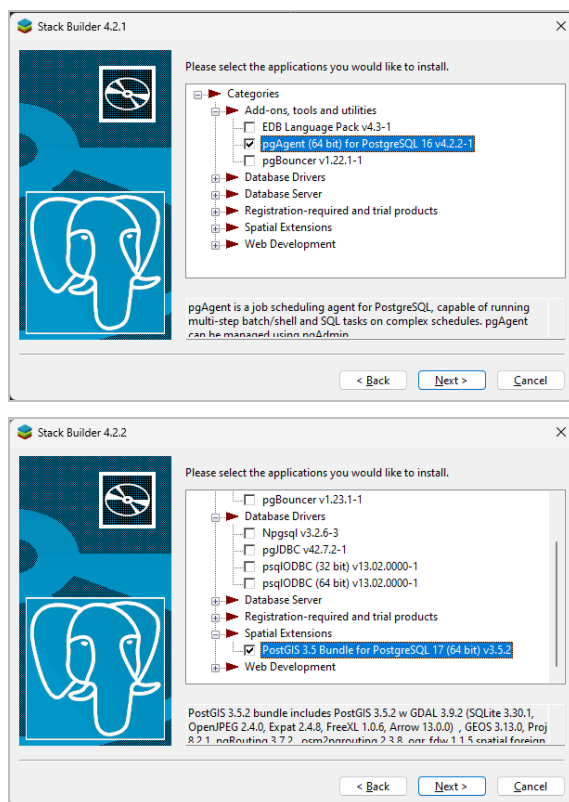
pgAgent este planificatorul de job-uri al PostgreSQL. Se instalează prin Stack Builder, pornit la sfârșitul instalării serverului sau ulterior, din directorul bin al PostgreSQL (stackbuilder.exe). Împreună cu el se instalează și PostGIS Bundle, necesar în unele cazuri pentru extensia HTTP (capitolul VII).

!!! **Atentie!** Pentru Stack Builder este necesară conexiune la internet pe serverul pe care se face instalarea.

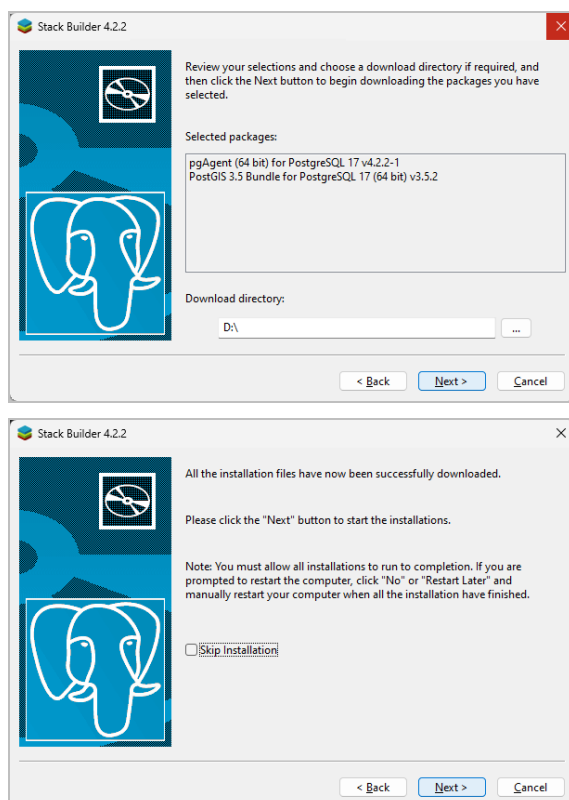
1. Se alege serverul PostgreSQL instalat: **PostgreSQL 18 (x64) on port 5432**.



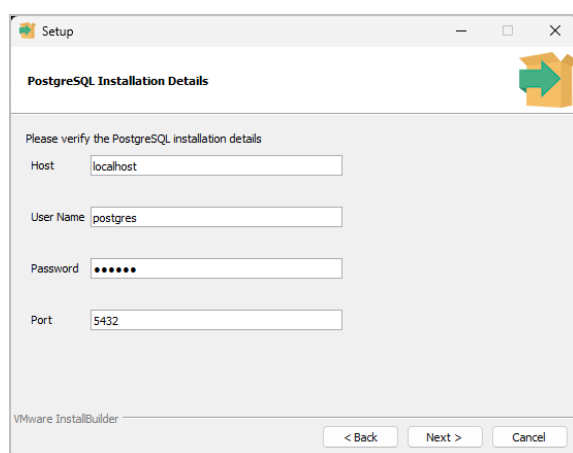
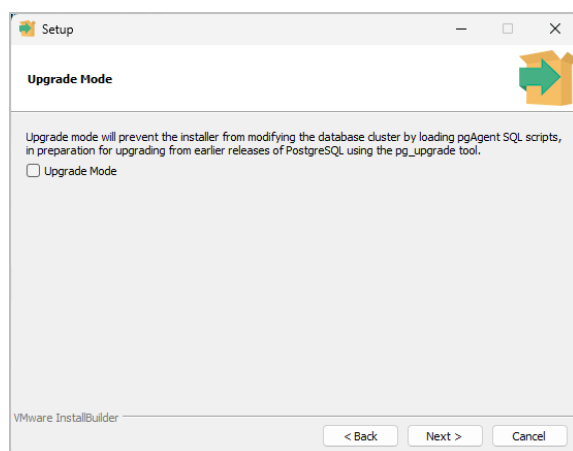
2. Se bifează pgAgent (64 bit) for PostgreSQL 18 sub Add-ons, tools and utilities și PostGIS Bundle for PostgreSQL 18 (64 bit) sub Spatial Extensions.



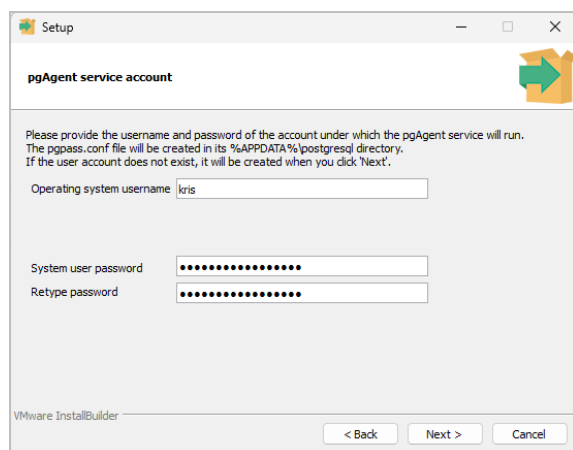
3. Se alege directorul în care se descarcă pachetele, apoi Next până la pornirea instalărilor.



4. Instalarea pgAgent: la *Upgrade Mode* opțiunea rămâne nebifată; se completează datele serverului și utilizatorul administrator al bazei de date (postgres).

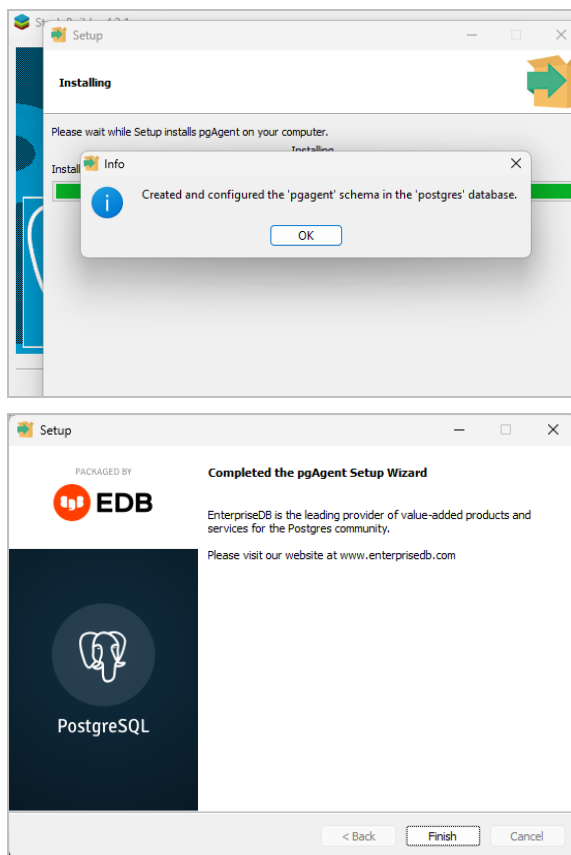


5. Se alege utilizatorul de sistem sub care pgAgent își creează și rulează serviciul.

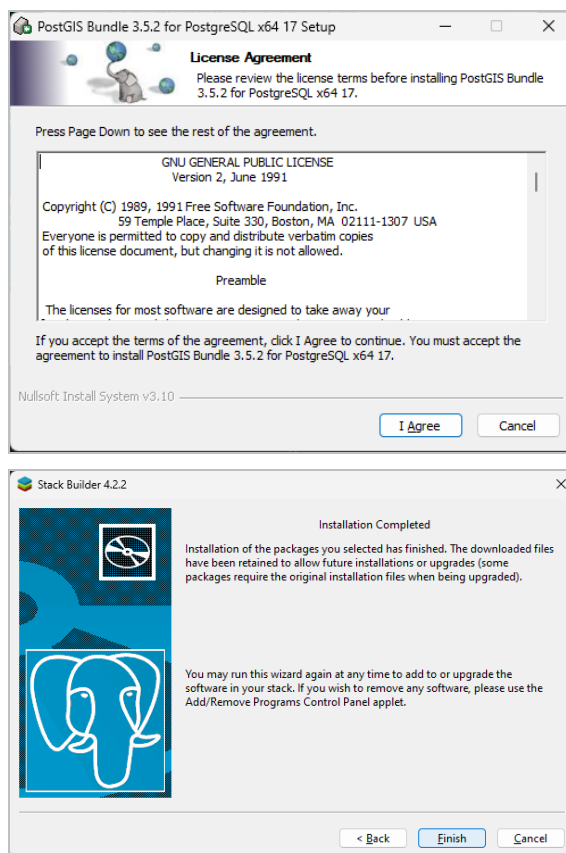


Obs! Pe unele sisteme de operare, când din această fereastră s-a creat un utilizator nou pentru serviciu, nu s-a creat structura %APPDATA% de care are nevoie serviciul la pornire. Recomandarea noastră este să folosiți un utilizator existent, cu drepturi de administrator, care s-a logat cel puțin o dată pe sistem și care are drepturile de administrator direct, nu moștenite prin apartenența la un alt grup inclus în grupul Administrators. Dacă utilizatorul ales nu are parolă, trebuie setată una: parola este obligatorie pentru pgAgent.

6. pgAgent își creează schema pgagent în baza postgres; se continuă până la Finish.



7. Instalarea PostGIS Bundle: se acceptă licența și se continuă cu opțiunile implicite; la final, Stack Builder confirmă instalarea pachetelor.



Obs! pgAgent este declarat învechit de producător și nu mai primește actualizări, dar este în continuare disponibil în Stack Builder pentru PostgreSQL 18. (Rev. 2.0)

IV. VERIFICAREA SERVICIILOR

După instalare se recomandă repornirea serverului. Se verifică în **Services** (services.msc) că ambele servicii sunt pornite (Running) și au pornire automată:

- **postgresql-x64-18** – serverul PostgreSQL 18; (Rev. 2.0)
- **pgagent-pg18** – PostgreSQL Scheduling Agent. (Rev. 2.0)

PostgreSQL Scheduling Agent - pgagent-pg17	Provides the ability to schedule tasks within a Postgres database server.	Running	Automatic	.\kris
postgresql-x64-17	Provides relational database storage.	Running	Automatic	Network S...

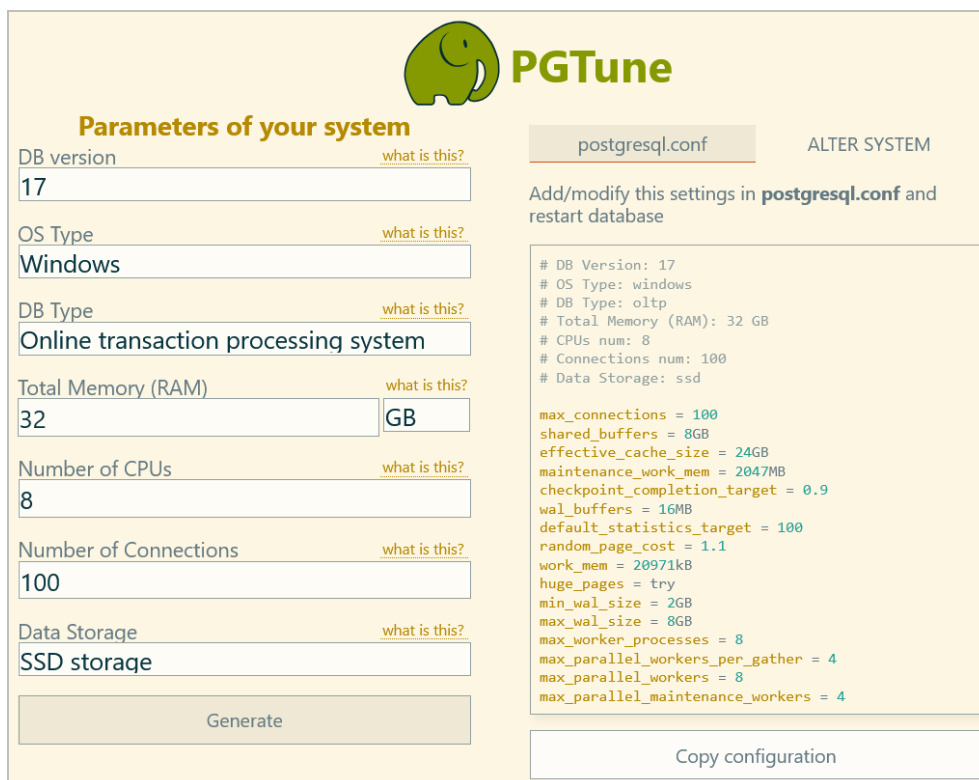
Verificarea se poate face și dintr-un PowerShell: (Rev. 2.0)

```
Get-Service postgresql-x64-18, pgagent-pg18 | ft Name, Status, StartType
```

V. CONFIGURAREA POSTGRESQL.CONF

Fișierul postgresql.conf conține parametrii de pornire ai serverului PostgreSQL și se află în directorul de date (de exemplu D:\PostgreSQL\18\data). Fișierul postgresql.conf din kit (directorul 03.Fisiere config postgres) conține setările pentru buna funcționare a WinMENTOR Enterprise; parametrii care depind de configurația hardware a serverului se completează după instalare și la fiecare modificare a configurației hardware.

1. Se face o copie a fișierului postgresql.conf creat la instalare, apoi se copiază peste el fișierul postgresql.conf din kit.
2. Se accesează pgtune.leopard.in.ua și se completează datele serverului: **DB version = 18**, OS Type = Windows, memoria RAM, numărul de procesoare, numărul de conexiuni și tipul de stocare, apoi se apasă **Generate**. (Rev. 2.0)



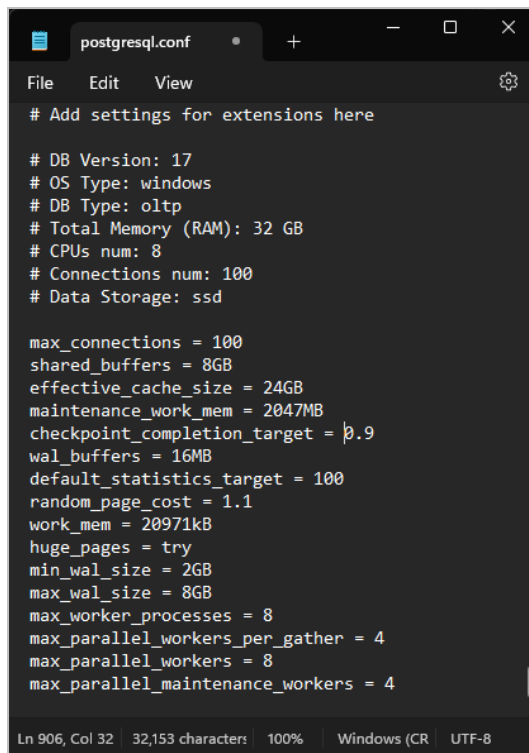
The screenshot shows the PGtune website interface. On the left, under 'Parameters of your system', there are input fields for: DB version (17), OS Type (Windows), DB Type (Online transaction processing system), Total Memory (RAM) (32 GB), Number of CPUs (8), Number of Connections (100), and Data Storage (SSD storage). Each field has a 'what is this?' link. At the bottom of this section is a 'Generate' button. On the right, there's a section for 'postgresql.conf' with an 'ALTER SYSTEM' button. Below this, it says 'Add/modify this settings in postgresql.conf and restart database'. A code block shows the generated configuration parameters, including max_connections = 100, shared_buffers = 8GB, effective_cache_size = 24GB, maintenance_work_mem = 2047MB, checkpoint_completion_target = 0.9, wal_buffers = 16MB, default_statistics_target = 100, random_page_cost = 1.1, work_mem = 20971kB, huge_pages = try, min_wal_size = 2GB, max_wal_size = 8GB, max_worker_processes = 8, max_parallel_workers_per_gather = 4, max_parallel_workers = 8, and max_parallel_maintenance_workers = 4. At the bottom right is a 'Copy configuration' button.

- Recomandarea inițială pentru DB Type este **Online transaction processing system (oltp)**.
- În funcție de modul de folosire a aplicației pot fi luate în calcul și **Mixed type of application** (server comun pentru baza de date și aplicație) sau **Data warehouse (dw)** (predominant rapoarte).

Number of Connections (max_connections) se alege după numărul de utilizatori WinMENTOR Enterprise:

Utilizatori WinMENTOR Enterprise	max_connections
între 1 și 5	40
între 5 și 10	60
între 10 și 20	80
între 20 și 50	100
peste 50	100 și se stabilește de la caz la caz

3. Se apasă *Copy configuration* și se lipesc parametrii la sfârșitul fișierului `postgresql.conf`, după linia **# Add settings for extensions here**. Parametrii adăugați la sfârșit au prioritate față de cei cu același nume de mai sus.



```

# Add settings for extensions here

# DB Version: 17
# OS Type: windows
# DB Type: oltp
# Total Memory (RAM): 32 GB
# CPUs num: 8
# Connections num: 100
# Data Storage: ssd

max_connections = 100
shared_buffers = 8GB
effective_cache_size = 24GB
maintenance_work_mem = 2047MB
checkpoint_completion_target = 0.9
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 1.1
work_mem = 20971kB
huge_pages = try
min_wal_size = 2GB
max_wal_size = 8GB
max_worker_processes = 8
max_parallel_workers_per_gather = 4
max_parallel_workers = 8
max_parallel_maintenance_workers = 4

```

Pe PostgreSQL 18, valoarea **maintenance_work_mem** propusă de PG Tune se păstrează așa cum este. Limita de 2047MB pe Windows, valabilă până la versiunea 17, a fost eliminată în versiunea 18. (Rev. 2.0)

!!! **Atenție!** Pe PostgreSQL 17 și pe versiunile anterioare, `maintenance_work_mem` nu poate depăși 2047MB pe Windows: o valoare mai mare (de exemplu 2GB, propusă la 32GB RAM) oprește pornirea serverului. Pe aceste versiuni se scrie `maintenance_work_mem = 2047MB`.

4. Se repornește serviciul `postgresql-x64-18`.

Obs! PostgreSQL 18 are câțiva parametri noi, care nu sunt în `postgresql.conf` din kit și rămân pe valorile implicite: `io_method` (citiri asincrone, implicit `worker`), `autovacuum_worker_slots` și `autovacuum_vacuum_max_threshold`. Nu este necesară modificarea lor. (Rev. 2.0)

Mentenanța tabelelor: VACUUM și AUTOVACUUM

În PostgreSQL, tabelele temporare sunt create pentru fiecare sesiune nouă, spre deosebire de Oracle. Din acest motiv tabela de sistem `pg_attribute` și indecșii ei pot crește, în anumite cazuri, foarte mult, iar creșterea afectează în timp performanța bazei de date. Pentru menținerea performanței și a spațiului, rândurile care nu mai sunt necesare se elimină periodic prin operația **VACUUM**.

În PostgreSQL, ștergerea unui rând (DELETE) nu îl elimină fizic, ci îl marchează ca „mort”; o actualizare (UPDATE) marchează rândul vechi ca mort și inserează unul nou (UPDATE = DELETE + INSERT). Rândurile moarte ocupă spațiu, „umflă” tabelele și

indecșii și încetinesc interogările, deoarece paginile încărcate în memorie conțin și aceste rânduri.

- **VACUUM** recuperează spațiul și îl face disponibil pentru reutilizare în aceeași tabelă; spațiul nu este returnat sistemului de operare.
- **VACUUM FULL** rescrie întreaga tabelă într-un fișier nou, fără spațiu liber, și returnează spațiul sistemului de operare. Este mult mai lent și blochează exclusiv fiecare tabelă pe durata prelucrării.

AUTOVACUUM este activ implicit în PostgreSQL și se configurează prin parametrii din postgresql.conf:

Parametru	Rol
autovacuum	activează sau dezactivează procesul de fundal autovacuum
autovacuum_vacuum_threshold	numărul minim de rânduri moarte dintr-o tabelă înainte de curățare (implicit 50)
autovacuum_analyze_threshold	numărul minim de rânduri modificate înainte de analiză (implicit 50)
autovacuum_vacuum_scale_factor	fracțiunea din dimensiunea tabelii care se adaugă la prag pentru declanșarea curățării (implicit 0,2)
autovacuum_analyze_scale_factor	fracțiunea din dimensiunea tabelii care se adaugă la prag pentru declanșarea analizei (implicit 0,1)
autovacuum_vacuum_cost_delay	pauza (în milisecunde) a procesului de curățare când atinge limita de cost (implicit 2 ms)
autovacuum_vacuum_cost_limit	limita de cost după care procesul de curățare face pauză (implicit -1, adică vacuum_cost_limit = 200)

Fișierul postgresql.conf din kit are deja aceste setări adaptate pentru WinMENTOR Enterprise. Este important ca vacuum și analiza să ruleze eficient, fără să încarce prea mult serverul. Se recomandă monitorizarea activității autovacuum și rularea manuală a comenzii VACUUM pentru tabelele care nu sunt întreținute suficient de autovacuum.

Fișierele de log

La importul de date în PostgreSQL, fișierele de log pot ajunge la dimensiuni foarte mari. Fișierele .log pot fi șterse manual sau de un job programat. Pe durata importului de date, sau a altor operații care generează log-uri mari, se poate opri generarea lor (necesită repornirea serviciului):

```
logging_collector = off
```

După terminarea importului se recomandă reactivarea log-urilor, de exemplu câte un fișier pentru fiecare zi a săptămânii, suprascris după 7 zile:

```
log_destination = 'stderr'
logging_collector = on
log_directory = 'log'
log_filename = 'postgresql-%a.log'
log_truncate_on_rotation = on
log_rotation_age = 1d
log_rotation_size = 0
```

Server PostgreSQL pe mașină virtuală

Pe mașini virtuale, serverul PostgreSQL nu are performanța unui server fizic, mai ales când utilizatorii lucrează remote direct pe serverul virtual. Diferența se reduce dacă la crearea mașinii virtuale se alocă procesoare și discuri dedicate (separat pentru instalarea PostgreSQL și pentru directorul de date) și suficientă memorie RAM, alocată fix, nu dinamic (minimum 16GB).

Dacă discurile mașinii virtuale nu sunt prealocate, se dezactivează în postgresql.conf următorii doi parametri, apoi se repornește serviciul:

```
wal_recycle = off
wal_init_zero = off
```

VI. CONFIGURAREA PG_HBA.CONF

Fișierul pg_hba.conf conține regulile de autentificare a clienților: adresele IP care au acces la serverul PostgreSQL și metoda de autentificare. Se copiază fișierul din kit în directorul de date (de exemplu D:\PostgreSQL\18\data) și se editează după rețeaua clientului. De exemplu, regula care permite conectarea de la orice adresă IPv4:

```
# IPv4: 0.0.0.0/0 permite conectarea de la orice IP
host    all             all             0.0.0.0/0               scram-sha-256
```

PostgreSQL 18 declară învechită autentificarea MD5, care va fi eliminată într-o versiune viitoare. Parolele noi sunt salvate implicit în format SCRAM (password_encryption = scram-sha-256), deci pe un server nou instalat se recomandă înlocuirea metodei **md5** cu **scram-sha-256** în toate liniile din pg_hba.conf din kit. (Rev. 2.0)

Obs! Metoda md5 funcționează în continuare pe PostgreSQL 18. Dacă baza de date provine de pe o versiune mai veche și are utilizatori cu parole salvate în format MD5, se lasă metoda md5 până când parolele acestor utilizatori sunt setate din nou (ALTER ROLE ... PASSWORD ...). Mesajele de avertizare privind MD5 se pot opri cu parametrul md5_password_warnings = off. (Rev. 2.0)

Se repornește serviciul postgresql-x64-18 după orice modificare a fișierelor postgresql.conf și pg_hba.conf.

VII. EXTENSIA HTTP

Extensia HTTP permite bazei de date să facă cereri HTTP (folosită, de exemplu, pentru comunicarea cu WMEWebService). Se instalează din arhiva **pg18http_w64.zip** din directorul 06.Extensii al kitului. (Rev. 2.0)

!!! **Atentie!** Arhiva pg18http_w64.zip este compatibilă doar cu PostgreSQL 18 (64 de biți, kitul EnterpriseDB). Pentru alte versiuni se folosește arhiva versiunii respective. (Rev. 2.0)

1. Se dezarchivează pg18http_w64.zip și se copiază conținutul directorelor **bin**, **lib**, **share** și **ssl** din arhivă în directoratele cu același nume din directorul de instalare PostgreSQL 18 (de exemplu D:\PostgreSQL\18).

Obs! Directorul extra_depends din arhivă nu se copiază. Conține bibliotecile Curl și SSL, care sunt deja în kitul PostgreSQL; se copiază în bin doar dacă la pasul 3 comanda CREATE EXTENSION http dă eroare din cauza lor. (Rev. 2.0)

2. Se instalează DBeaver din directorul 05.DBeaver - pentru interogare BD și se face o conexiune la serverul PostgreSQL, ca utilizator postgres.

3. Din DBeaver se execută:

```
SET search_path TO public;  
CREATE EXTENSION http;
```

4. Se verifică funcționarea extensiei: (Rev. 2.0)

```
SELECT extname, extversion FROM pg_extension WHERE extname = 'http';  
-- asteptat: http | 1.7  
SELECT status FROM http_get('http://postgis.net');  
-- asteptat: 200
```

Obs! Pentru adresele https, dacă apare eroarea „SSL certificate problem: unable to get local issuer certificate”, se indică fișierul de certificate copiat din arhivă (se înlocuiește calea cu directorul de instalare), apoi se reîncarcă configurația: (Rev. 2.0)

```
ALTER SYSTEM SET http.CURLOPT_CAINFO = 'D:/PostgreSQL/18/ssl/certs/ca-bundle.crt';  
SELECT pg_reload_conf();
```

Pe unele sisteme comanda CREATE EXTENSION http nu reușește. În acest caz se instalează din Stack Builder, categoria *Spatial Extensions*, pachetul **PostGIS Bundle for PostgreSQL 18** (capitolul III), apoi se repetă pașii 1 și 3.

VIII. EXTENSIA WM_PGTT (OPȚIONAL)

Extensia **wm_pggtt** creează tabelele temporare ale WinMENTOR Enterprise doar în momentul în care sunt folosite, în loc să le creeze pe toate la fiecare conectare; rezultatul este o conectare și o schimbare de firmă mult mai rapide. Extensia se livrează doar pentru PostgreSQL 18. Instalarea și activarea ei sunt descrise în documentul separat „WME – Instalare și configurare extensie PostgreSQL WM_PGTT” și se fac după instalarea WinMENTOR Enterprise (capitolul IX). (Rev. 2.0)

IX. INSTALAREA WINMENTOR ENTERPRISE PE SERVER

1. Se creează directorul WinMENTOR Enterprise, ca la instalarea pe Oracle.

2. Se copiază în acest director, din directorul 04.Installer WME -PG al kitului: WMInstaller_PG.exe, PGMARTOR.ZIP, PGSERVICE.ZIP, WinMService, dbparams.ini și Nethasp.ini. Fișierele dbparams.ini și Nethasp.ini se configurează pentru serverul clientului.



Exemplu de dbparams.ini pentru PostgreSQL:

```
[DB]
POSTGRES=1
[POSTGRESQL]
NumeServerul1=192.168.0.11
NumeServerul2=192.168.0.11:5433
DBName=postgres
ServerName=NumeServerul1
```

- **DBName** este numele bazei de date și ține cont de litere mari / mici (de exemplu postgres).
- **NumeServerul1, NumeServerul2, ...** sunt serverele la care se pot loga utilizatorii, cu adresa IP a serverului PostgreSQL; **ServerName** este serverul curent.
- Pe același server se pot instala mai multe versiuni PostgreSQL, pe porturi diferite (de exemplu 5432 și 5433), cu câte un director WinMENTOR Enterprise separat, fiecare cu dbparams.ini configurat pentru serverul lui.

3. Se copiază DLL-urile de client PostgreSQL din directorul 02.DLL-ri pentru clientul de postgres, după tabelul de mai jos.

Sistem / executabile	Arhiva	Se dezarchivează în
Windows pe 32 de biți	PG_DLL_x32.zip	C:\Windows\System32
Windows pe 64 de biți, executabile pe 32 de biți	PG_DLL_x32.zip	C:\Windows\SysWOW64
Windows pe 64 de biți, executabile pe 64 de biți	PG_DLL_x64.zip	C:\Windows\System32

Obs! Dacă se folosesc executabile atât pe 32, cât și pe 64 de biți, se copiază ambele variante, fiecare în directorul ei. Cele două variante au fișiere cu aceleași nume, deci nu se pot pune amândouă lângă executabilul WinMENTOR Enterprise: lângă executabil se pot pune doar DLL-urile de 64 de biți cu executabile de 64 de biți, sau doar cele de 32 de biți cu executabile de 32 de biți.

4. Se rulează WMInstaller_PG.exe din directorul creat (alături de PGMARTOR.ZIP și PGSERVICE.ZIP), care creează structura bazei de date WinMENTOR Enterprise și configurarea ei, ca la instalarea pe Oracle.

X. STAȚIILE CLIENT

Pe stațiile client nu se instalează PostgreSQL. Pe fiecare stație:

1. Se copiază DLL-urile de client PostgreSQL, după tabelul din capitolul IX.
2. Se configurează Nethasp.ini, PROTECT.DAT și dbparams.ini în directorul WinMENTOR Enterprise.
3. Se instalează driverul cheii de protecție, de la adresa:

<http://download.winmentor.ro/WMEnterprise/Tools/hasp/>

4. Se dezarchivează DLL_pt_AnafConnector.zip (variante pe 32 sau pe 64 de biți) și chkVIES.zip în directorul WinMENTOR Enterprise.

5. Se copiază executabilele WinMENTOR Enterprise.

XI. ACTUALIZAREA VERSIUNII WINMENTOR ENTERPRISE

Executabilele noi WinMENTOR Enterprise se copiază, când este cazul, din directorul versiunii respective, de la adresa:

| <http://download.winmentor.ro/WMEnterprise/Versiuni%20WME/>

XII. REVIEW-URI DOCUMENT

Rev. 1.0	06.02.2025	Versiunea inițială, pentru PostgreSQL 17
Rev. 2.0	02.10.2026	Actualizare pentru PostgreSQL 18 și trecere pe formatul documentației WME: kitul pentru versiunea 18, extensia HTTP pentru PostgreSQL 18, maintenance_work_mem fără limita de 2047MB, scram-sha-256 în pg_hba.conf, sumele de control, extensia WM_PGTT, serverul pe mașină virtuală

