

Administracja i programowanie pod Microsoft SQL Server 2000

Paweł Rajba

pawel@ii.uni.wroc.pl
<http://www.kursy24.eu/>

Zawartość modułu 13

- Zastosowania SQL Server Agent
- SQL Mail, SQL Server Agent Mail
- Konfiguracja SQLMaila
- SQLServerAgent
 - Zarządzanie zadaniami
 - Operatorzy
 - Zarządzanie zadaniami

SQL Server Agent

- Zastosowania
 - przetwarzanie alertów
 - uruchamianie zadań w terminie
 - wysyłanie powiadomień do operatorów

SQL Mail

- SQL Mail
 - służy do wykonywania zapytań i zwracania zbiorów wyników przez email
 - działa za pośrednictwem usługi SQL Server
 - można za jego pomocą m.in.:
 - przetworzyć maile ze skrzynki pocztowej SQL Mail
 - wysłać maila

SQL Server Agent Mail

- SQL Server Agent Mail
 - służy do wysyłania zawiadomień poprzez email lub pager, np.
 - czy zadanie się powiodło, czy nie
 - komunikat informujący o uruchomieniu alertu
 - działa za pośrednictwem usługi SQL Server Agent

Konfiguracja SQLMail

- SQLServer jako klient poczty
 - Podłączamy serwera Microsoft Exchange
 - Tworzymy skrzynkę pocztową na koncie uruchamiającym SQL Server i SQL Server Agent.
 - Instalujemy klienckie oprogramowanie pocztowe (np. Microsoft Outlook)
 - Dla każdego konta tworzymy profile pocztowe za pomocą narzędzia Poczta w Panelu Sterowania. Nazwy powinny być SQLMailAccount i/lub SQLServerAgentAccount
Wskazówka: przy tworzeniu profilu należy zalogować się na konto, dla którego tworzony jest profil
 - Konfiguracja SQL Mail
 - EM | Support Services | SQL Mail

Przykład konfiguracji SQLMail

- Środowisko testowe:
 - Kontroler domeny nwtraders.msft (LONDON):
 - Windows 2000 Server
 - Exchange 2000
 - Outlook 2003
 - Komputer wpięty do domeny
 - Windows 2000 Server
 - SQLServer 2000 + SP3
 - Outlook 2003

Przykład konfiguracji SQLMail

- Logujemy się na konto Administrator
 - tworzymy konta + skrzynki pocztowe
 - SQLServerAccount
 - SQLAgentAccount
 - dodajemy konta do grupy Administratorzy domeny
- Logujemy się na konto SQLServerAccount
 - Włączamy Outlooka, tworzymy profil na Exchange
 - Włączamy Panel Sterowania | Mail
 - usuwamy istniejący profil Outlook
 - dodajemy nowy profil o nazwie SQLServerMail do Exchange'a na konto SQLServerAccount

Przykład konfiguracji SQLMail

- Logujemy się na konto Administrator
 - Uruchamiamy Enterprise Manager
 - Wybieramy właściwości serwera, zakładkę security
 - Ustawiamy konto usługi na NWTRADERS\SQLServerAccount
 - Wybieramy Support Services, właściwości SQL Mail
 - Profil ustawiamy na SQLServerMail, klikamy Test

Przykład 1 użycia SQL Maila

- Logujemy się na konto Administrator
 - Uruchamiamy: Query Analyzer, Outlook
 - W Query Analyzer wykonujemy zapytanie:
 - xp_sendmail
 - @recipients='Administrator@nwtraders.msft',
 - @message='Ilość rekordów w tabeli pubs..titles:',
 - @subject='Wiadomość od mailera SQL',
 - @query='SELECT COUNT(*) FROM pubs..titles'
 - Jeśli pojawia się błąd 'xp_sendmail: failed with mail error 0x80004005', wykonujemy master.dbo.xp_stopmail i wykonujemy zapytanie ponownie
 - W Outlooku powinien pojawić się mail

Przykład 2 użycia SQL Maila

- Mamy bazę sklep z listy drugiej. Tworzymy następujący wyzwalacz

```
CREATE TRIGGER Przeterminowane ON dbo.Dostawa
FOR INSERT, UPDATE
AS BEGIN
    IF EXISTS (SELECT * FROM Sklep..Dostawa WHERE
                DataWaznosci < GETDATE ())
    BEGIN
        EXEC master.dbo.xp_sendmail
            @recipients='Administrator@nwtraders.msft',
            @message='Lista towarów przeterminowanych:',
            @subject='SQLMail: Przeterminowane',
            @query='zapytanie' --na następnym slajdzie
    END
END
```

Przykład 2 użycia SQL Maila

- Zapytanie dajemy następujące:

```
SELECT d.*, p.* FROM Sklep.dbo.Dostawa d JOIN  
    Sklep.dbo.Produkt p ON d.IdProduktu=p.IdProduktu  
WHERE d.DataWaznosci<GETDATE ()
```

- Następnie w Query Analyzer dajemy zapytania:

```
insert into sklep.dbo.produkt  
values (1, 'Cukier', 'kg')
```

```
insert into sklep.dbo.dostawa  
values (1,1,10, '2005-10-20', '2005-10-30')
```

```
insert into sklep.dbo.dostawa  
values (1,1,10, '2005-10-2', '2005-10-10')
```

Przykład 3 użycia SQL Maila

- Przykład prostszy, robimy demonstrację
- Tworzymy tabelę zamówienie
 - Zamowienie(id int, towar varchar(80), data datetime, akceptacja varchar(3))
-

Przykład 3 użycia SQL Maila

- Tworzymy wyzwalacz
- ```
CREATE TRIGGER TR_DoAkceptacji on Zamowienie
AFTER INSERT
AS
BEGIN
 CREATE TABLE Tempdb.Aux (Id INT, Towar VARCHAR(80),
 Data DATETIME);
 INSERT INTO Tempdb.Aux
 SELECT Id, Towar, Data FROM Inserted;
 EXEC xp_sendmail
 @recipients='Administrator@nwtraders.msft',
 @subject='Zamowienie do akceptacji',
 @query='SELECT * FROM Tempdb..Aux';
 DROP TABLE Tempdb.Aux;
END;
```

# Przykład 3 użycia SQL Maila

---

- Na końcu dodajemy zamówienia
  - INSERT INTO Zamowienie  
VALUES (1, 'Banany', '2006-05-20', null);
  - INSERT INTO Zamowienie  
VALUES (2, 'Gwoździe', '2006-05-22', null);
- I patrzemy, jak spływają maile

# Przykład 4 użycia SQL Maila

---

- Wysyłamy trzy maile od Administrator do SQLServerAccount; kolejne maile zawierają
  - `SELECT * FROM pubs..titles`
  - `SELECT * FROM pubs..authors`
  - `SELECT * FROM Northwind..Products`
- Następnie w SQL Query Analyzer odpalamy procedurę
  - `sp_processmail`
- I patrzymy, jak spływają maile



# SQL Server Agent

---

- Konfiguracja usługi SQL Server Agent
  - Usługa musi być włączona cały czas (należy włączyć autostart przy uruchamianiu)
  - Konto usługi musi mieć w SQLServer rolę sysadmin
    - Konto systemowe – należy do grupy adminów i jest mapowane na sysadmin; daje dostęp tylko do zasobów lokalnych i nie ma dostępu do sieci
    - Konto domenowe – niezbędne, gdy SQLSA musi się komunikować z innymi serwerami poczty, i ogólnie, gdy musi mieć dostęp do zasobów sieciowych

# SQL Server Agent

---

- Automatyczne ponowne uruchamianie usług
  - EM | Management | Właściwości SQL Server A. | Zakładka Advanced | Restart Services
- Ustawianie profilu pocztowego
  - EM | Management | Właściwości SQL Server A. | Mail Session
- Przeglądanie dzienników SQL Server Agent
  - EM | Management | SQL Server A. | prawy przycisk myszy – Display Error Log

# Zarządzanie zadaniami

---

- Tworzenie zadania
  - Enterprise Manager
    - EM | Management | Jobs | (ppm) New Job...
  - TSQL
    - sp\_add\_job (składnia w dokumentacji)
- Dane zapisywane są w
  - msdb..sysjobs

# Zarządzanie zadaniami

---

- Uwagi przy tworzeniu zadania
  - Pole Name zawiera opisową nazwę (max. 128 znaków). Zmiana nazwy NIE powoduje zmiany w odwołaniach (np. plikach historii, czy dziennikach)
  - Kategorie pomagają uporządkować zadania
    - Zarządzanie kategoriami  
EM | Management | Jobs | (ppm) Wszystkie zadania |  
| Manage Job Categories...

# Zarządzanie zadaniami

---

- Tworzenie kroku zadania
  - Enterprise Manager
    - New Job... | Zakładka steps | New...
  - TSQL
    - sp\_add\_jobstep (składnia w dokumentacji)
- Dane zapisywane są w
  - msdb..sysjobsteps

# Zarządzanie zadaniami

---

- Rodzaje kroków w zadaniu
  - Wyrażenia T-SQL-a
    - Przy definiowaniu kroku opartego na T-SQL
      - należy podać bazę danych
      - należy podać wymagane parametry do procedur
      - można wyniki zapytania przekierować do pliku
      - nie można wyników zapytania przekierować jako wejście dla następnego kroku
    - Każdy może wykonywać takie wyrażenia
  - Kroki związane z replikacją

# Zarządzanie zadaniami

---

- Rodzaje kroków w zadaniu
  - Komendy systemu operacyjnego
    - Przy definiowaniu kroku
      - należy podać kod wyjścia w przypadku sukcesu
      - należy podawać pełne ścieżki do poleceń
    - Należy mieć uprawnienia roli sysadmin

# Zarządzanie zadaniami

---

- Rodzaje kroków w zadaniu
  - Skrypty ActiveX
    - Przy definiowaniu kroku
      - należy sprecyzować język skryptu; językiem skryptu może być VBScript, JScript i dowolny inny, dla którego są odpowiednie biblioteki
    - Należy mieć uprawnienia roli sysadmin
- Uwagi przy tworzeniu kroku zadania
  - W zakładce Advanced określamy
    - Akcję po sukcesie i akcję po porażce
    - Liczbę prób i odstęp między nimi



# Zarządzanie zadaniami

---

- Tworzenie harmonogramu zadań
  - Enterprise Manager
    - New Job... | Zakładka Schedules | New Schedule...
  - TSQL
    - `sp_add_jobschedule` (składnia w dokumentacji)
- Dane zapisywane są w
  - `msdb..sysjobschedules`

# Zarządzanie zadaniami

---

- Harmonogram może działać wg schematu
  - Przy uruchamianiu SQL Server Agent
  - Jednorazowo w określonym momencie
  - Okresowo
  - W momencie bezczynności procesora

# Zarządzanie zadaniami

---

- **Zawiadomienia**
  - Są po to, by informować o sukcesie lub porażce zadania
  - Zarządzane w zakładce Notifications
  - Powiadamiani są operatorzy (omówione dalej)

# Operatorzy

---

- Co to jest operator?
  - Specjalne konta, które mogą otrzymywać zawiadomienia o wyzwoleniach alertów i udanym lub nie zakończeniu zaplanowanego zadania
- Przy tworzeniu operatora zaleca się:
  - Używać aliasu mailowego dla grupy w celu powiadomienia więcej niż jednej osoby
  - Przetestować każdy sposób powiadamiania
  - Zaplanować harmonogram pracy operatora, żeby wysyłać powiadomienia na pager w godzinach pracy
  - Używać polecenia net send do powiadamiania w obrębie sieci Windows

# Operatorzy

---

- Uwaga:
  - Przy podawaniu adresów e-mail, należy używać pełnej formy np. [smithj@microsoft.com](mailto:smithj@microsoft.com)
- Zarządzanie operatorami
  - EM | Management | SQLServerAgent | Operators
  - Dodanie poprzez procedurę
    - sp\_add\_operator
- Dane o operatorach zapisywane są w
  - msdb..sysoperators

# Operatorzy

---

- Operator awaryjny (fail-safe) otrzymuje powiadomienie, gdy
  - SQLServerAgent nie może uzyskać dostęp do niezbędnych tabel systemowych bazy msdb
  - Nie udało się skontaktować z wyznaczonym operatorem przez pager lub żaden z operatorów nie ma dyżuru (wg harmonogramu pracy)
- Ustawienie operatora awaryjnego
  - EM | Management | SQLServerAgent | Właściwości | zakładka Alert System | Fail-safe operator

# Operatorzy i zadania

---

- Przegląd historii zadań
  - Zadanie | (ppw) View Job History...
  - Tabela msdb..sysjobhistory

# Przykład 5: operatorzy i zadania

---

- Tworzymy operatora:
  - Name: Paweł
  - E-mail:Administrator@NWTraders.msft
  - net send address: cassiopeia
- Testujemy adres email i adres net send



# Przykład 5: operatorzy i zadania

---

- Tworzymy zadanie: Tools | Wizards... | Management | Create Job Wizard
  - Typ: Transact-SQL command
  - Baza danych: master
  - Transact-SQL: `BACKUP DATABASE master TO DISK = 'C:\Program Files\Microsoft SQL Server\Mssql\Backup\MasterDB.bak' WITH INIT`
  - Harmonogram: każdy poniedziałek o 17.00
  - Powiadomienie: email i netsend do operatora Paweł
  - Job name: Master DB Backup

# Przykład 5: operatorzy i zadania

---

- Uruchamiamy zadanie ręcznie
  - (ppm) Start Job
- Sprawdzamy, czy dostaliśmy e-maila
- Przeglądamy historię zadania
  - View Job History...
- Edytujemy zadanie Master DB Backup
  - Psujemy ścieżkę, gdzie ma być zapisany backup
- Ponownie uruchamiamy zadanie ręcznie
- Patrzymy: Podgląd Zdarzeń, Dziennik aplikacji

# Zarządzanie alertami

---

- Zastosowanie
  - Umożliwiają wysyłanie wiadomości w przypadku pojawienia się
    - błędów SQL Servera
    - błędów tworzonych przez użytkownika
    - warunków wydajnościowych
  - Pozwala na uruchomienie zadania

# Zarządzanie alertami

---

- Alerty domyślne
  - Demo: Full msdb log
    - pełny log bazy msdb
  - Demo: Full tempdb
    - pełny log bazy tempdb
  - Demo: Sev. 19 Errors
    - wyzwalany przy pojawieniu się błędu o poziomie krytyczności 19
  - ...
  - Demo: Sev. 25 Errors

# Zarządzanie alertami

---

- Tworzenie alertu
  - Enterprise Manager
    - EM | Management | SQL Server Agent | Alerts | (ppm) New alert...
  - T-SQL
    - sp\_add\_alert (składnia w dokumentacji)

# Zarządzanie alertami

---

- Alert może być oparty na
  - Zdarzeniu SQL Server; określamy wtedy
    - numer błędu lub
    - poziom krytyczności
  - Zdarzeniu wydajnościowym; określamy wtedy
    - obiekt, licznik
    - poziom licznika, przy którym pojawi się alert

# Przykład 6: zarządzanie alertami

---

- Prezentacja: tworzenie alertu na błąd zdefiniowany przez użytkownika
  - Tworzymy komunikat błędu
    - Przy tworzeniu należy zwrócić uwagę na
      - numer błędu musi być większy niż 50000
      - komunikaty błędów są składowane w master..sysmessages
      - tekst komunikatu może zawierać parametry
    - Do tworzenia i zmiany służą procedury
      - sp\_addmessage, sp\_altermessage
    - Przykład
      - EXEC sp\_addmessage 50099, 16, 'Customer %d was deleted by %s', 'us\_english','true'

# Przykład 6: zarządzanie alertami

---

- Prezentacja: tworzenie alertu ... (cd.)
  - Do rzucania błędów w wyzwalaczach i procedurach służy polecenie RAISERROR (przełączamy na bazę Northwind)
    - CREATE PROCEDURE removecustomer  
@CustomerID varchar(5) = NULL  
AS  
DECLARE @username varchar(60)  
SET @username = suser\_sname()  
BEGIN TRANSACTION  
DELETE Customers  
WHERE CustomerID = @CustomerID  
RAISERROR (50099, 16, 1, @CustomerID, @username)  
COMMIT TRANSACTION



# Przykład 6: zarządzanie alertami

---

- Prezentacja: tworzenie alertu ... (cd.)
  - Tworzymy alert
    - Name: DeleteCustomer
    - Type: SQL Server event alert
    - Error Number: 50099
    - Ustawiamy, że operator Paweł ma być powiadamiany przez email i netsend

# Przykład 6: zarządzanie alertami

---

- Prezentacja: testujemy działanie alertu
  - Dodajemy rekord do tabeli Customers:
    - INSERT INTO Customers (CustomerID, CompanyName) VALUES ('12345', 'YogiBear')
  - Usuwamy poprzez procedurę
    - exec removecustomer '12345'
  - Sprawdzamy maile

# Przykład 6: zarządzanie alertami

---

- Prezentacja: tworzenie alertu wydajnościowego
  - Tworzymy zadanie, które robi backup logu bazy Northwind co 4 godziny:
    - uruchamiamy skrypt NWLOG.SQL
  - W właściwościach bazy Northwind ustawiamy
    - EM | Databases | Northwind | Właściwości | Transaction Log | Automatically grow file = 0
    - EM | Databases | Northwind | Właściwości | Options | Recovery | Model = Full
  - Robimy backup bazy Northwind
    - uruchamiamy skrypt NWBackup.sql

# Przykład 6: zarządzanie alertami

---

- Prezentacja: tworzenie...
  - Tworzymy alert
    - Nazwa: Northwind Log 60% Full
    - Typ: SQL Server performance condition alert
    - Obiekt: SQLServer:Databases
    - Licznik: Percent Log Used
    - Instancja: Northwind
    - Alert, jeśli licznik „rises above 60”
    - Execute job: Backup Northwind Log
    - Powiadomienie przez email i netsend operatora Paweł
    - Delay between responses: 0

# Przykład 6: zarządzanie alertami

---

- Prezentacja: tworzenie...
  - Przeglądamy zadanie Backup Northwind Log | Schedules
  - Będziemy śledzić procent zużycia logu w konsoli Windows
    - Start|Programy|Narzędzia administracyjne|Wydajność
    - Monitor systemu | (ppm) Dodaj liczniki
      - Obiekt wydajności: SQLServer:Databases
      - Licznik: Percent Log Used
      - Instancja: Northwind

# Przykład 6: zarządzanie alertami

---

- Prezentacja: tworzenie...
  - Zaczynamy testowanie
    - Uruchamiamy skrypt Fulltlog.sql
    - Patrzymy na Dzienniki wydajności
  - Kończymy
    - Wyłączamy zapytanie
    - Patrzymy w Outlooku na otrzymane maile
    - Patrzymy na Historię Backup Northwind Log
    - We właściwościach alertu patrzymy na historię
    - Przywracamy
      - Northwind|Właściwości|Transaction Log|Automatically...
      - Northwind|Właściwości|Options|Recovery|Model = Simple