Microsoft

Wee-Hyong Tok
Rakesh Parida
Matt Masson
Xiaoning Ding
Kaarthik Sivashanmugam

Microsoft® SQL Server® 2012 Integration Services

Przekład: Marek Włodarz

APN Promise, Warszawa 2012

Microsoft® SQL Server® 2012 Integration Services
© 2012 APN PROMISE SA

Authorized Polish translation of English edition of Microsoft® SQL Server® 2012 Integration Services, ISBN: 978-0-7356-6585-9

APN PROMISE SA, biuro: ul. Kryniczna 2, 03-934 Warszawa
tel. +48 22 35 51 600, fax +48 22 35 51 699
e-mail: mspress@promise.pl

ISBN: 978-83-7541-104-1

Przekład: Marek Włodarz
Korekta: Ewa Swędrowska
Skład i łamanie: MAWart Marek Włodarz

Spis treści

Przedmowa

W roku 1989, gdy wszyscy byliśmy dużo młodsi, miałem paskudną weekendową robotę: w tygodniu pracowałem jako programista w firmie Microrim Incorporated, twórcy R:Base – wówczas drugiej najpopularniejszej bazy danych dla komputerów osobistych na świecie. Zaś w sobotnie ranki siadałem zupełnie sam w naszym budynku w Redmond i odbudowywałym bazę danych obsługującą call center. Obejmowało to uzyskanie z księgowości ostatnich zarejestrowanych licencji, zaktualizowanej listy pracowników z działu personalnego, arkuszy kalkulacyjnych z działu marketingu, które zawierały informacje o dostawcach oprogramowania, a przede wszystkim całą historię połączeń telefonicznych z dziennika centralki – i zbieranie tego wszystkiego razem. Oczywiście, w żadnym z tych systemów nie istniał spójny schemat formatowania ani przechowywania danych. Zajmowało to zwykle około sześciu godzin – o ile nie pomyliłem się w którymś kroku. Cały proces był „oskryptowany" na kartce papieru. Wówczas nie istniała jeszcze taka nazwa, ale w istocie tworzyłem hurtownię danych.

Każdy, kto kiedykolwiek wykonywał taką pracę, słyszał to wielokrotnie: uzyskanie właściwych danych we właściwym kształcie i miejscu, a do tego na czas, to 80 procent wysiłku wkładanego w każdy projekt oparty na danych. Integrowanie danych to stacja pomp w podziemiach, dzięki której piękna fontanna płynie bez zakłóceń. Zwykle to fontanna przyciąga całą uwagę, ale my w zespole SSIS w firmie Microsoft jesteśmy dumni z tych pomp, które budujemy.

Autorzy tej książki stanowią rdzeń tego zespołu. Tak długo, jak go znam, Kaarthik zawsze był żarliwym wyznawcą prostej prawdy: możemy ocenić jakość produktu tylko wtedy, gdy najpierw zrozumiemy dokładnie klientów, którzy go używają. Jako pierwszy pracownik zespołu pochodzący z Chin, Xiaoning przetarł szlak. Jest jednym z tych cichych geniuszy, którzy sprawiają, że wszyscy inni milkną, gdy zabierają głos, bo zawsze mają coś bardzo ważnego do powiedzenia. Jedną z moich najlepszych decyzji było nie zgodzenie się z radą mojego menedżera co do zatrudnienia Matta. Widzicie, on niezbyt pasuje do naszego modelu działania. Tak, potrafił napisać doskonały kod, ale było coś, co nie pasowało do naszych oczekiwań. Ciągle wracał do potrzeby zbudowania pełnego rozwiązania dla problemów biznesowych; wręcz nie potrafił przestać mówić o tym! No i ostatecznie – doprowadziliśmy do tego, że to działa. Nie mówcie Wee Hyongowi, że to powiedziałem, ale najprawdopodobniej ma zbyt wysokie kwalifikacje do tej pracy. Jego przygotowanie jako wykładowcy na uczelni i historia dokonań jako SQL MVP czyniły go doskonałym kandydatem na jedną z publicznych

twarzy zespołu SSIS. I na koniec Rakesh. Pod koniec pierwszego tygodnia pracy u nas postanowił zorganizować spotkanie dla naszych klientów podczas targów, które miały odbyć się w niedługim czasie. Wymusił na kolegach udział w tym wydarzeniu, znalazł pokój w centrum kongresowym i wysłał zaproszenia do naszych klientów. Wszystko w ciągu kilku dni.

Przygotowywanie wydania 2012 SSIS rozpoczęliśmy od wysłuchania oczekiwań tych klientów. Ich priorytety były jasne: sprawić, aby produkt był łatwiejszy w użyciu i prostszy do zarządzania. Może wydawać się to prostym celem, ale w trakcie lektury tej książki zaskoczyło mnie, jak wiele z tych celów udało się osiągnąć i jak bardzo zdołaliśmy ulepszyć i tak dobry produkt. Dla nowicjuszy w dziedzinie SSIS książka ta będzie dobrym punktem wyjścia przy rozwiązywaniu prawdziwych problemów, zaś weterani znajdą tu nowe spojrzenie na dobrze znane fakty.

Gdy poproszono mnie o napisanie tej przedmowy, pakowałem się przed wyjazdem w swoim biurze w budynku 34 w Redmond, One Microsoft Way. Przez okno mogłem zobaczyć budynek 21 po drugiej stronie ulicy. Dwadzieścia pięć lat temu budynek ten mieścił światową centralę Microrim Incorporated. Przypomniałem sobie tego samotnego chłopaka w sobotnie poranki. Świat jest mały.

Jeff Bernhardt
Group Program Manager, SQL Server Data Movement
Szanghaj, Chiny

Wprowadzenie

Microsoft SQL Server Integration Services to korporacyjna platforma dla projektowania i realizacji rozwiązań integrowania danych. Zapewnia możliwości wydobywania i ładowania danych z heterogenicznych źródeł i kierowania ich do rozmaitych miejsc docelowych. Dodatkowo zapewnia możliwości ułatwiające wdrażanie, zarządzanie i konfigurowanie tych rozwiązań. Jeśli ktoś jest programistą rozwiązań integracyjnych lub administratorem bazy danych potrzebującym takiego rozwiązania, SQL Server Integration Services jest właściwym narzędziem, którego powinien użyć.

Książka *Microsoft SQL Server 2012 Integration Services* przedstawia ogólny przegląd funkcjonalności Microsoft SQL Server Integration Services, ze szczególnym uwzględnieniem nowych możliwości wprowadzonych w wydaniu SQL Server 2012. Liczne przykłady koncentrują się na konkretnych zagadnieniach, ale przedstawione są również zasady funkcjonowania wewnętrznego mechanizmu. Choć książka nie zawiera wyczerpującego omówienia wszystkich funkcji Integration Services, jest dobrym przewodnikiem po wykorzystywaniu kluczowych możliwości tego systemu.

Poza właściwym tekstem, do każdego rozdziału dołączone są przykładowe projekty i procedury, które Czytelnik powinien przeanalizować samodzielnie.

Kto powinien przeczytać tę książkę

Książka ta nie jest przeznaczona dla początkujących, ale jeśli Czytelnik wykroczył już poza podstawy, lektura pomoże mu w tworzeniu prawdziwie działających rozwiązań SQL Server Integration Services! Znajdzie tu setki oszczędzających czas rozwiązań, wskazówek i podpowiedzi, a także:

- Pogłębione informacje o nowych możliwościach Integration Services wprowadzonych w SQL Server 2012.
- Wskazówki i wzorce implementacji rozwiązań Integration Services.
- Techniki rozwiązywania problemów wydajnościowych.
- Omówienie metod diagnozowania problemów i stosowania zaawansowanych funkcji debugowania.

Założenia

Autorzy książki oczekują, że Czytelnik dysponuje przynajmniej podstawową wiedzą na temat działania Microsoft SQL Server Integration Services oraz co najmniej dobrą znajomością koncepcji relacyjnych baz danych i platformy SQL Server. Książka zawiera przykłady kodu napisane w językach Transact-SQL, C# oraz PowerShell. Osoby, które nie znają przynajmniej podstaw tych języków, powinny rozważyć rozpoczęcie od lektury takich książek, jak *Microsoft Visual C# 2010 Step by Step* autorstwa Johna Sharpa (Microsoft Press, 2010)* lub doskonały podręcznik Itzika Ben-Gana *Microsoft SQL Server 2012 T-SQL Fundamentals* (Microsoft Press, 2012)**.

Książki tej nie należy traktować jako podręcznika tworzenia rozwiązań integracyjnych. Autorzy koncentrują się na zaawansowanych, szczegółowych zagadnieniach i zakładają, że Czytelnik potrafi już zbudować i uruchomić pakiety SSIS.

Kto nie powinien czytać tej książki

Książka ta nie zawiera omówienia podstaw SQL Server ani innych powiązanych technologii, takich jak Analysis Services, Reporting Services, Master Data Services czy Data Quality Services.

Organizacja książka

Książka została podzielona na pięć części, z których każda koncentruje się na innym aspekcie Microsoft SQL Server Integration Services. Część I, „Przegląd", zawiera krótkie omówienia koncepcji Integration Services oraz uwarunkowań dotyczących aktualizacji do wersji Microsoft SQL Server 2012. Część II, „Projektowanie", prezentuje nowe narzędzia projektowania Integration Services i ich zastosowanie w tworzeniu rozwiązań integracji danych. Dodatkowo w tej części przedstawiona została nowa funkcjonalność przechwytywania zmian danych (Change Data Capture) oraz funkcje czyszczenia danych. Część III, „Konfiguracje, zarządzanie i monitorowanie", pokazuje techniki konfigurowania projektów i pakietów, a także sposoby wykorzystania Transact-SQL i PowerShell do programowego zarządzania Integration Services. Dodatkowo zostały w niej przedstawione wbudowane raporty. Zaawansowane koncepcje i wewnętrzne mechanizmy Integration Services stanowią tematykę części IV, „Pogłębione informacje". Końcowa część V, „Rozwiązywanie problemów" koncentruje się na zagadnieniach monitorowania i rozwiązywania problemów, takich jak niepowodzenia wykonania pakietów, wykrywanie wąskich gardeł lub problemy dotyczące danych.

---

* Wydanie polskie: *Microsoft Visual C# 2010 Krok po kroku*, APN Promise, Warszawa 2010, ISBN 978-83-7541-066-2.

** Wydanie polskie: *Microsoft SQL Server 2012: Podstawy języka T-SQL*, APN Promise, Warszawa 2012, ISBN: 978-83-7541-101-0.

Znalezienie najlepszego punktu startowego

Poszczególne części książki *Microsoft SQL Server 2012 Integration Services* obejmują szeroki zakres zagadnień. Zależnie od potrzeb i dotychczasowej wiedzy na temat różnych możliwości oferowanych przez SQL Server Integration Services Czytelnik może skupić się na wybranych partiach książki. Poniższa tabela może pomóc w wybraniu właściwego miejsca, od którego należy rozpocząć lekturę.

Jeśli jesteś	Zacznij tu
Nowicjuszem w dziedzinie SQL Server Integration Services	Zacznij od Części I i II, a następnie przejdź do rozdziałów 10 i 11 w Części III, albo czytaj całą książkę po kolei.
Znasz wcześniejsze wydania SQL Server Integration Services	Możesz pominąć Część I, chyba że potrzebujesz odświeżenia podstawowych informacji. Przeczytaj o nowych rozwiązaniach w Części II, III, i V, nie pomijając rozdziału 17 w Części IV.
Zainteresowany wykorzystaniem Transact-SQL lub PowerShell do programowej obsługi SQL Server Integration Services	Tematykę taką przedstawiają rozdziały 12 i 13 w Części III.
Zainteresowany funkcjami monitorowania i rozwiązywania problemów w SQL Server Integration Services	Przeczytaj rozdziały wchodzące w skład Części V.

Większość rozdziałów zawiera praktyczne przykłady, które Czytelnik powinien wykonać, aby wypróbować właśnie poznane koncepcji. Bez względu na to, na których rozdziałach zamierza się skupić, zawsze warto pobrać i zainstalować przykładowe aplikacje w swoim systemie.

Konwencje używane w tej książce

W książce zastosowano kilka konwencji wydawniczych, które powinny ułatwić przekaz informacji.

- Nazwy elementów strukturalnych rozwiązań SSIS (takich jak zadania lub kontenery) pisane są *kursywą*.
- Dane wpisywane przez użytkownika w przykładach są wyróżniane **czcionką pogrubioną**.
- Nazwy poleceń, okien dialogowych i innych elementów interfejsu użytkownika są wyróżnione czcionką jednoelementową.
- Fragmenty kodu (listingi) złożone zostały `czcionką stałopozycyjną`.
- Ramki oznaczone ikonami i tytułami, takimi jak „Uwaga”, zawierają dodatkowe informacje lub alternatywne techniki osiągania celów.

Wymagania systemowe

Wykonanie ćwiczeń praktycznych opisanych w książce wymaga następującego sprzętu i oprogramowania:

- SQL Server 2012 Standard Edition lub wyższe wydanie wraz z SQL Server Management Studio 2012. Wymaga to następującego środowiska:
  - Komputera wyposażonego w procesor o częstotliwości taktowania 1.4 GHz lub szybszy i pamięć 1 GB (zalecane 2 GHz i 4 GB RAM lub więcej);
  - System operacyjny Windows Server 2008 lub późniejszy albo Windows Vista Business lub późniejszy;
  - .NET 3.5 Service Pack 1;
  - Bardziej szczegółowe wymagania systemowe dla SQL Server 2012 zawiera dokument *http://msdn.microsoft.com/en-us/library/ms143506.aspx*.
- Połączenie z Internetem w celu pobrania przykładowych plików.

Instalacja i konfigurowanie produktów wchodzących w skład pakietu SQL Server 2012 wymaga lokalnych praw administracyjnych na używanym komputerze.

Przykłady kodu

Wszystkie przykładowe projekty w postaci przed i po wykonaniu ćwiczeń można pobrać ze strony internetowej stowarzyszonej z książką pod następującym adresem:

http://go.microsoft.com/FWLink/?Linkid=258311

Po pobraniu pliku .zip należy rozpakować go do wybranej lokalizacji na dysku twardym. W przykładach przedstawionych w książce zakłada się, że lokalizacją tą jest C:\Insideout\. W razie wybrania innej lokalizacji konieczne będzie zmodyfikowanie niektórych parametrów w przykładach.

Uwaga Oprócz przykładów kodu na komputerze musi być zainstalowane oprogramowanie SQL Server 2012 wraz z usługami Integration Services oraz narzędzia SQL Server Management Studio.

Większość przykładów opiera się na przykładowej bazie danych AdventureWorks2012. Bazę tę można pobrać z witryny Codeplex pod następującym adresem:

http://msftdbprodsamples.codeplex.com/releases/view/55330

Podziękowania

Autorzy chcieliby podziękować wszystkim profesjonalistom SQL Server, którzy współpracowali z zespołem Integration Services przez ostatnie lata, aby doprowadzić produkt do obecnej postaci, a także innym członkom zespołu SQL Server Integration Services za ich pomoc i współpracę podczas tworzenia książki. W szczególności chcemy podziękować Jeffowi Bernhardtowi za napisanie przedmowy oraz zespołowi redakcyjnemu w Microsoft Press i O'Reilly.

Errata i pomoc dotycząca książki

Dołożyliśmy wszelkich starań, aby zagwarantować poprawność tej książki i towarzyszącej treści. Błędy zauważone po publikacji zostaną skorygowane na stronie Microsoft Press w witrynie oreilly.com:

http://go.microsoft.com/FWlink/?Linkid=258310

CZĘŚĆ I

Przegląd

ROZDZIAŁ 1

Ogólna charakterystyka SSIS

Przedsiębiorstwa polegają na integracji danych, aby móc przekształcić je w wartościowe spostrzeżenia i właściwe decyzje. Integracja danych na poziomie przedsiębiorstwa jest złożonym problemem ze względu na heterogeniczność źródeł danych i ich formatów, nieustannie powiększające się zbiory danych, a często także z powodu ich kiepskiej jakości. Dane są zazwyczaj przechowywane w odrębnych systemach, przez co występują różnice w ich formatach lub schematach organizacyjnych, które trzeba rozwiązywać. Stale malejące koszty magazynowania prowadzą do wydłużenia czasu przechowywania danych i towarzyszącego temu wzrostowi rozmiarów zbiorów informacji, które trzeba przetwarzać. To z kolei prowadzi do ciągłego zwiększania się popytu na skalowalne rozwiązania integracji danych o wysokiej wydajności, dzięki którym organizacje mogą na czas uzyskiwać wnioski wynikające ze zgromadzonych danych. Zróżnicowanie danych i ich niespójne duplikowanie powoduje problemy dotyczące jakości, które mogą wpływać na precyzję i dokładność spostrzeżeń analitycznych, a tym samym na jakość i wartość wynikających z nich decyzji. Projekty integrowania danych muszą radzić sobie z tymi wyzwaniami i efektywnie wykorzystywać dane z najrozmaitszych źródeł (takich jak bazy danych, arkusze kalkulacyjne, pliki tekstowe itp.), co pociąga za sobą potrzebę czyszczenia, korelowania, transformowania i przenoszenia danych źródłowych do systemów docelowych. Proces ten jest jeszcze bardziej złożony ze względu na fakt, że funkcjonowanie wielu organizacji jest zależne od ciągłej dostępności ich magazynów danych; z tego powodu integrowanie danych musi odbywać się często, zaś operacje te muszą być kończone tak szybko, jak to możliwe.

Technologia Microsoft SQL Server Integration Services (SSIS) stanowi odpowiedź na te wyzwania, zapewniając platformę do budowania i zarządzania rozwiązaniami integracji danych. Elastyczna, rozszerzalna i skalowalna platforma wysokiej wydajności oraz zestaw narzędzi zawartych w SSIS zaspokaja potrzeby przedsiębiorstw w zakresie tradycyjnych procesów ekstrakcji, transformowania i ładowania danych (*Extract-Transform-Load* – ETL), a także inne potrzeby integrowania danych. SSIS jest funkcjonalnością oprogramowania Microsoft SQL Server, która zapewnia płynne współdziałanie z innymi funkcjami wbudowanymi zarówno w SQL Server, jak i inne produkty firmy Microsoft. Typowe scenariusze integracji danych, które można zrealizować przy użyciu SSIS, obejmują:

- Konsolidowanie danych z heterogenicznych źródeł
- Przenoszenie danych pomiędzy różnymi systemami

- Ładowanie danych do hurtowni danych
- Czyszczenie, formatowanie lub standaryzowanie danych
- Identyfikowanie, przechwytywanie i przetwarzanie zmian w danych
- Koordynowanie konserwacji, przetwarzania lub analizowania danych

Niektóre scenariusze przetwarzania danych wymagają wyspecjalizowanych technik. SSIS nie jest odpowiednią platformą dla następujących typów przetwarzania danych:

- Przetwarzanie złożonych reguł biznesowych względem danych
- Koordynowanie, uzyskiwanie i przetwarzanie danych w procesach wewnątrzbiznesowych
- Przetwarzanie komunikatów o zdarzeniach w czasie rzeczywistym
- Koordynowanie komunikacji pomiędzy systemami
- Budowanie sfederowanych widoków źródeł danych
- Przetwarzanie i integracja niestrukturalnych danych

Typowe scenariusze wykorzystania SSIS

W tym podrozdziale zaprezentujemy szczegółowo kilka typowych scenariuszy integracji danych i pokażemy, jak kluczowe funkcje SSIS mogą pomóc w każdym z tych scenariuszy.

Konsolidowanie danych z heterogenicznych źródeł

W typowej organizacji dane zazwyczaj nie są zawarte w jednym systemie, ale są rozproszone po całym przedsiębiorstwie. Różne aplikacje mogą używać swoich własnych magazynów danych o różnych schematach. Analogicznie, różne części organizacji mogą dysponować swoimi własnymi, lokalnie skonsolidowanymi widokami danych. Mogą też występować izolowane starsze systemy, zapewniające dostępność danych dla reszty organizacji w regularnych odstępach czasu. Aby móc podejmować wynikające z całości tych danych ważne decyzje dotyczące całej organizacji, konieczne jest ściągnięcie danych ze wszystkich części przedsiębiorstwa, skomasowanie ich i przetransformowanie do spójnego stanu i kształtu.

Potrzeba konsolidacji danych wyłania się też podczas przejmowania lub łączenia się organizacji. Wsparcie łączności pomiędzy heterogenicznymi magazynami i wydobywanie danych jest kluczową funkcją każdego oprogramowania integracji danych. SSIS wspiera łączność z szeroką gamą popularnych magazynów danych przy użyciu wbudowanych adapterów i funkcji rozszerzających. Adaptery źródłowe wczytują dane ze źródeł zewnętrznych do SSIS, podczas gdy adaptery docelowe umożliwiają zapisanie danych z SSIS do zewnętrznych magazynów.

Niektóre z najważniejszych wbudowanych adapterów źródłowych i docelowych zawartych w SSIS to:

- OLE DB – źródłowy i docelowy
- ADO.NET – źródłowy i docelowy
- ODBC – źródłowy i docelowy
- Flat File (plik tekstowy) – źródłowy i docelowy
- Excel – źródłowy i docelowy
- XML – tylko źródłowy

Uwaga Komponenty źródłowe i docelowe Open Database Connectivity (ODBC) są dostępne począwszy od edycji Integration Services 2012 i nie są dostępne dla wcześniejszych wersji. W pakietach SQL Server 2008 oraz SQL Server 2008 R2 można wykorzystać komponenty źródłowe i docelowe ADO.NET do łączenia się ze źródłami ODBC przy użyciu .NET ODBC Data Provider. Komponent docelowy ADO.NET nie jest dostępny w pakiecie SQL Server 2005.

Do innych typów adapterów SSIS należą:

- Skrypty źródłowe i docelowe: pozwalają one programistom SSIS na tworzenie kodu autorskiego, zapewniającego łączność z magazynami danych, które nie są obsługiwane przez adaptery wbudowane w SSIS.
- Adaptery specjalnego zastosowania: większość adapterów w SSIS to mechanizmy ogólnego stosowania, wspierające dowolne magazyny danych, do ktorych można uzyskać dostęp poprzez interfejsy standardowe; jednak niektóre z tych adapterów są właściwe dla określonego typu magazynu danych i zależy od specyficznego interfejsu programowania aplikacji (API). Przykładem takich adapterów szczególnego zastosowania są *SQL Server Destination* oraz *Dimension Processing Destination*, odpowiednio zapewniające łączność z instancjami SQL Server oraz Analysis Server.
- Adaptery niestandardowe: przy użyciu mechanizmów rozszerzających SSIS klienci lub niezależni dostawcy oprogramowania mogą tworzyć własne adaptery, które pozwolą na połączenie z magazynami danych, dla których nie ma wbudowanego wsparcia w SSIS.

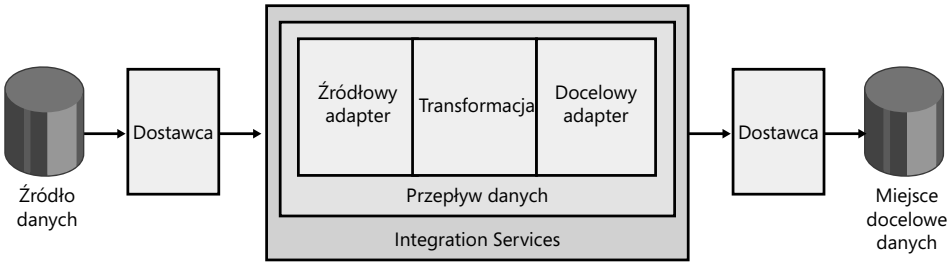
Adaptery źródłowe i docelowe, które nie są częścią instalacji SSIS, ale są dostępne do pobrania z witryny Microsoft.com, zawierają między innymi:

- Oracle – źródłowy i docelowy
- Teradata – źródłowy i docelowy
- SAP BW – źródłowy i docelowy

Uwaga Skryptowanie SSIS jest obsługiwane przez Visual Studio for Applications w wersji SQL Server 2005 oraz Visual Studio Tools for Applications w wersji SQL Server 2008 i późniejszych. Visual Studio for Applications oraz Visual Studio Tools for Applications to oparte na platformie .NET technologie skryptowe, pozwalające dołączyć niestandardowe funkcjonalności do aplikacji. Obydwa rozwiązania zapewniają środowisko wykonawcze (*runtime*), które wykonuje niestandardowy kod używając silnika skryptowego oraz zintegrowane środowisko programistyczne (IDE) do pisania i debugowania kodu. Visual Studio for Applications wspiera język VB.Net, zaś Visual Studio Tools for Applications obsługuje zarówno język VB.Net, jak i C#.

Uwaga Konektory Oracle, Teradata oraz SAP BW są dostępne tylko dla zaawansowanych wydań SQL Server. Szczegółowe informacje o poszczególnych wydaniach SQL Server zawarte są w dalszej części tego rozdziału. Konektory Oracle i Teradata są dostępne do pobrania ze strony *http://www.microsoft.com/download/en/details.aspx?id=29283*. Komponent Microsoft Connector 1.1 for SAP BW jest oferowany jako część pakietu SQL Server Feature Pack, dostępnego pod adresem *http://www.microsoft.com/download/en/details.aspx?id=29065*.

Adaptery SSIS obsługują informacje o połączeniach z zewnętrznymi magazynami danych przy użyciu *menedżerów połączeń* (*connection managers*). Menedżery połączeń SSIS opierają się na zależnych od rozwiązania dostawcach danych lub sterownikach. Na przykład adaptery OLE DB wykorzystują OLE DB API oraz dostawcę danych do uzyskiwania dostępu do magazynów wspierających standard OLE DB. Adaptery połączeniowe SSIS używane są wewnątrz zadania *Data Flow* (Przepływu danych), które jest wykonywane za pośrednictwem silnika przepływu danych, wykorzystującego wysoko wydajne przesyłanie i transformowanie danych pomiędzy źródłami a miejscami docelowymi. Rysunek 1-1 ilustruje przepływ danych od źródła do miejsca docelowego za pośrednictwem dostawców danych lub sterowników.



Rysunek 1-1 Schemat przepływu danych od źródła do miejsca docelowego

Integration Services oferują wiele opcji łączenia się z relacyjnymi bazami danych. Adaptery OLE DB, ADO.NET i ODBC zapewniają podstawowe API magazynów danych, pozwalających na łączenie się z szeroką gamą baz danych. Jedyną popularną opcją łączności z bazą danych, która nie jest wspierana w SSIS, jest Java Database Connectivity (JDBC). Projektanci SSIS często stają w obliczu wyzwania, jakim jest wybranie odpowiedniego adaptera spośród dostępnych przy łączeniu się z określonym magazynem danych. Podczas dokonywania wyboru należy uwzględnić następujące czynniki:

- Wspierane typy danych
- Metadane udostępniane przez sterownik lub dostawcę
- Wsparcie dla sterownika lub dostawcy w 32- i 64-bitowych środowiskach
- Wydajność

Wspierane typy danych

Wsparcie dla typów danych w różnych relacyjnych bazach danych (wykraczających poza standard ANSI SQL) różni się znacznie; każda baza ma własny system typów. Typy danych wspierane przez dostawców danych i sterowniki zapewniają warstwę abstrakcji dla tych systemów typów w magazynach danych. Narzędzia integracji danych muszą zapewnić, że podczas odczytywania, przetwarzania lub zapisywania danych nie dojdzie do utraty informacji o typie danych. SSIS wykorzystuje swój własny system typów danych. Adaptery zawarte w SSIS mapują zewnętrzne typy danych ujawniane przez dostawców na typy danych SSIS i zapewniają wierność typów podczas interakcji z zewnętrznymi magazynami. System typów danych SSIS redukuje problemy występujące przy radzeniu sobie z różnicami w typach danych pomiędzy różnymi mechanizmami i dostawcami magazynującymi, zapewniając spójną podstawę dla przetwarzania danych. SSIS niejawnie konwertuje dane na równoważne typy w swoim własnym systemie podczas czytania lub zapisywania danych. Jeśli takie działanie nie jest możliwe, konieczne może być jawne przekonwertowanie danych na typy binarne lub łańcuchowe, aby uniknąć utraty informacji.

Uwaga Dokument *http://msdn.microsoft.com/en-us/library/ms141036.aspx* zawiera wyczerpującą listę typów danych SSIS.

Metadane ujawniane przez dostawcę

SQL Server Data Tools zapewnia środowisko projektowe, w którym można budować pakiety SSIS, będące jednostkami wykonywalnymi w środowisku SSIS. Możliwości projektowe dostępne w SQL Server Data Tools zależą od tego, jakie metadane są ujawniane przez magazyny danych za pośrednictwem sterowników lub dostawców, dzięki

czemu projektanci mogą właściwie definiować właściwości pakietów. Metadane takie umożliwiają pobranie listy baz danych, tabel, widoków oraz właściwości kolumn w tabelach i widokach podczas konstruowania pakietu. Jeśli magazyn danych nie ujawnia określonych metadanych lub jeśli sterownik nie udostępnia interfejsu pozwalającego pobrać pewne metadane z magazynu, będzie to miało wpływ na sprawność pracy projektanta pakietu SSIS. W takich przypadkach konieczne może być ręczne ustawienie odpowiednich właściwości w pakietach SSIS.

Uwaga Pakiet projektowy Integration Services w wersjach SQL Server 2005, 2008 oraz 2008 R2 nosi nazwę Business Intelligence Development Studio. W wersji SQL Server 2012, środowisko projektowe SSIS stało się częścią zintegrowanego zestawu narzędzi o nazwie SQL Server Data Tools, który udostępnia narzędzia projektowe baz danych i BI w jednolitym środowisku.

Wsparcie w środowiskach 32- i 64-bitowych

Pakiety SSIS można wykonywać zarówno w trybie 32-, jak i 64-bitowym. Jeśli konkretna aplikacja jest 32-bitowa, SSIS użyje 32-bitowego dostawcy danych.

Wersje 32- i 64-bitowe dostawców danych zazwyczaj mają ten sam identyfikator. Gdy biblioteka zostanie wskazana poprzez identyfikator, jej wersja ładowana w czasie wykonywania zależna będzie od aplikacji, która ją wywołuje. Dostawca danych dostępny dla pakietów SSIS zależeć będzie od trybu, w którym pakiet jest wykonywany. Na przykład wykonywanie pakietów wewnątrz narzędzi SQL Server Data Tools domyślnie odbywa się w trybie 32-bitowym; tym samym użyty będzie 32-bitowy dostawca danych. Pakiety, które są wykonywane z powodzeniem w trybie 32-bitowym, niekoniecznie muszą dać wykonywać się w trybie 64-bitowym (i *vice versa*). Wynika to z faktu, że dostawcy danych lub sterowniki mogą nie być dostępne dla obydwu trybów. Jeśli 64-bitowy sterownik nie jest dostępny na komputerze wykonującym pakiet, wykonanie się nie powiedzie, gdy podjęta zostanie próba wykonania w trybie 64-bitowym. Projektanci pakietów SSIS i administratorzy muszą mieć to na uwadze podczas tworzenia i wykonywania pakietów.

Uwaga Można pominąć domyślny tryb 32-bitowy w SQL Server Data Tools, ustawiając właściwość pakietu *Run64BitRuntime* na *True*. Właściwość ta jest skuteczna tylko wewnątrz SQL Server Data Tools; nie ma żadnego wpływu na wykonywanie pakietu za pośrednictwem SQL Server Management Studio lub narzędzia DTExec. Jeśli pakiet wykonywany jest w innym kontekście, właściwość ta jest ignorowana; tym niemniej, istnieją inne sposoby kontrolowania trybu wykonywania pakietu w tych kontekstach.

Wydajność

Istnieje wiele czynników wpływających na wydajność operacji integrowania danych. Jednym z najważniejszych czynników jest sprawność adapterów, która jest bezpośrednio zależna od wydajności dostawców danych lub sterowników niskiego poziomu używanych przez adaptery. Choć istnieją pewne ogólne zalecenia (patrz tabela 1-1) dotyczące tego, jakiego adaptera należy użyć dla każdej z popularnych baz danych, nie istnieje żadna gwarancja, że użycie zalecanych adapterów zapewni najlepszą wydajność. Sprawność adaptera zależy od wielu czynników, takich jak zaangażowane sterowniki lub dostawcy danych, a także od trybu bitowego sterowników. Zaleca się, aby projektanci SSIS dokonywali porównań różnych opcji łączności przed ustaleniem, której należy użyć w środowisku produkcyjnym.

Tabela 1-1 Zalecane adaptery dla niektórych popularnych magazynów danych

Baza danych	Zalecany adapter
SQL Server	OLE DB – źródłowy i docelowy
Oracle	Oracle – źródłowy i docelowy
Teradata	Teradata – źródłowy i docelowy
DB2	OLE DB – źródłowy i docelowy
MySQL	ODBC – źródłowy i docelowy
SAP BW	SAP BI – źródłowy i docelowy
SAP R/3	ADO.Net – źródłowy i docelowy

Uwaga Konektory Oracle i Teradata są dostępne do pobrania z adresu *http://www.microsoft.com/download/en-us/details.aspx?id=29283*. Połączenie z bazą danych SAP R/3 wymaga Microsoft .NET Data Provider for mySAP Business Suite, który jest dostępny jako część BizTalk Adapter Pack 2.0 pod adresem *http://www.microsoft.com/download/en-us/details.aspx?id=2755*. Oprogramowanie BizTalk nie jest wymagane do instalacji pakietu adapterów ani do korzystania z dostawcy danych SAP. Zalecamy wykorzystanie dostawcy Microsoft OLE DB Provider for DB2 dla łączności z bazami danych DB2 – jest on dostępny jako część Microsoft Host Integration Server albo w ramach pakietu SQL Server Feature Pack.

Przenoszenie danych pomiędzy systemami

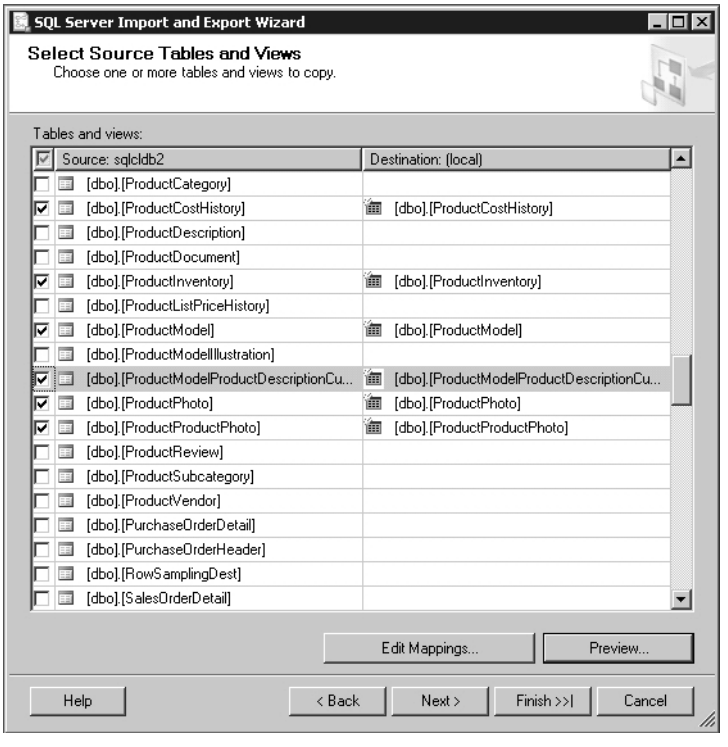
Scenariusze integracji danych przedstawione w tym podrozdziale objaśniają przenoszenie danych pomiędzy systemami magazynującymi. Przenoszenie danych może być jednorazową operacją wykonywaną podczas migracji do nowej wersji systemu lub aplikacji, ale może to być również powtarzalny proces, periodycznie przenoszący nowe dane z jednego magazynu danych do innego. Przykładem jednorazowego przeniesienia

może być migrowanie danych przed wyłączeniem starego systemu. Kopiowanie przyrostowych danych ze starszego systemu do nowego w regularnych odstępach czasu w celu zagwarantowania, że nowy system będzie w stanie zastąpić stary, może być przykładem powtarzalnego ruchu danych. Tego typu działania zazwyczaj obejmują konieczność transformacji danych, aby dostosować przenoszone informacje do schematu systemu docelowego. Źródłowe i docelowe adaptery w SSIS omówione wcześniej ułatwiają łączenie starszych i nowych systemów.

Komponenty transformacyjne zawarte w SSIS pozwalają wykonywać takie operacje, jak konwersja formatów, grupowanie, scalanie, próbkowanie, sortowanie, dystrybuowanie i wiele innych typowych działań względem danych kierowanych do potoku danych SSIS. W samym SSIS te komponenty przyjmują przepływ danych z potoku jako wejście, po czym zwracają przekształcone wyjście z powrotem do potoku, przy czym wynikowe dane mogą mieć taki sam kształt lub różnić się od wejścia. Komponenty transformacyjne mogą operować na danych wiersz po wierszu, na podzbiorach wierszy lub na raz na całych zbiorach danych. Wszystkie transformacje w SSIS są wykonywane w pamięci, co pomaga uzyskać wysoką wydajność przetwarzania i transformowania danych. Każda operacja transformacyjna jest definiowana dla jednej lub więcej kolumn danych w potoku przepływu danych. W celu wykonania operacji, które nie są wspierane wbudowanymi funkcjami, programiści SSIS mogą wykorzystać skrypty lub budować niestandardowe transformacje. Wbudowane transformacje SSIS, które zapewniają wsparcie dla najczęściej spotykanych operacji na danych, obejmują:

- **Aggregate (Agregowanie)** Stosuje funkcje agregujące, takie jak Average (Średnia), Count (Zliczanie) lub Group By (Grupowanie) do wartości w kolumnach i kopiuje wyniki do wyjścia transformacji.
- **Conditional Split (Podział warunkowy)** Kieruje wiersze danych do różnych wyjść w zależności od ich zawartości.
- **Multicast (Rozpraszanie)** Rozdziela kolejne wiersze z wejścia na kilka wyjść w celu przetwarzania rozproszonego.
- **Lookup (Wyszukiwanie)** Realizuje wyszukiwanie poprzez złączanie danych wejściowych kolumn z kolumnami w zbiorze referencyjnym.
- **Merge (Łączenie)** Scala dwa posortowane zbiory danych w pojedynczy zbiór danych.
- **Sort (Sortowanie)** Ustawia dane wejściowe w porządku rosnącym lub malejącym.
- **Union All (Złączanie)** Łączy wiele wejść w jedno wyjście.
- **Data Conversion (Konwersja danych)** Konwertuje dane w kolumnie wejściowej na inny typ danych.
- **Derived Column (Kolumna wynikowa)** Tworzy nowe wartości kolumn poprzez zastosowanie wyliczeń dla wartości zawartych w kolumnach wejściowych.

Jednorazowa migracja danych może obejmować szeroki zakres operacji – od prostego przeniesienia danych bez żadnych przekształceń, po skrajnie złożone działania wykorzystujące więcej niż jedno źródło i rozbudowaną logikę transformacji danych. Pakiety realizujące złożone przenoszenia danych mogą powstawać na bazie prostych, jednorazowych przeniesień, ale mogą być również budowane od podstaw przez programistów SSIS, korzystających z SQL Server Data Tools. Gdy pracownik pobiera dane z tabeli bazy danych i importuje je do arkusza Excel w celu dalszej analizy i przetwarzania, mamy do czynienia z prostym, jednorazowym ruchem danych. Użytkownik taki zazwyczaj nie dysponuje wiedzą na temat koncepcji ETL ani funkcji SSIS. Kreator *Import and Export Wizard* w SSIS pomaga takim użytkownikom budować proste rozwiązania przenoszenia danych. Kreator tworzy i wykonuje pakiet SSIS w tle, ukrywając przed użytkownikiem złożoność zaangażowaną w budowę pakietu. Tworzone przez kreatora pakiety wykorzystują źródłowe i docelowe adaptery dla magazynów danych uczestniczących w ruchu. Rysunek 1-2 ukazuje krok kreatora, w którym dokonywany jest wybór tabel będących źródłem danych do skopiowania.



Rysunek 1-2 Dokonywanie wyboru źródłowych danych w kreatorze Import and export Wizard

Po utworzeniu pakietu przez kreatora można go zapisać i edytować później w środowisku SQL Server Data Tools (omówionym bardziej szczegółowo w dalszej części

tego rozdziału). Ta funkcjonalność jest przydatna dla programistów SSIS, którzy mogą w ten sposób uaktualniać i rozbudowywać pakiety tworzone przez pracowników wiedzy, dodając bardziej złożone transformacje przed udostępnieniem tych pakietów pracownikom działu IT. Źródła i miejsca docelowe danych wspierane przez kreator importu i eksportu obejmują:

- Relacyjne bazy danych wspierające dostawców .NET Framework Provider lub OLE DB Provider
- Pliki programów Microsoft Office: Access oraz Excel
- Pliki tekstowe (*plain text*) rozdzielane separatorami

W tworzonych przez kreator pakietach można włączyć proste funkcje transformacyjne, aby obsłużyć mapowanie typów danych pomiędzy źródłem i miejscem docelowym. W celu uniknięcia nadmiernej złożoności przy dostosowywaniu typów danych kreator automatycznie mapuje typy z każdej kolumny wybranej do przeniesienia danych w źródle na typy kolumn docelowych, korzystając z plików mapujących będących częścią standardowej instalacji. SSIS udostępnia domyślne pliki mapujące w formacie XML dla najczęściej używanych kombinacji źródła i miejsca docelowego. Na przykład kreator wykorzysta plik mapujący o nazwie DB2ToMSSql10.xml, gdy dane są przenoszone z systemu DB2 do SQL Server 2008 lub nowszej wersji. Plik ten umożliwia zamapowanie każdego typu danych występującego w DB2 na odpowiadający mu typ w SQL Server 2008 lub wersji późniejszej. Listing 1-1 przedstawia fragment tego pliku, odpowiedzialny za przypisanie typu danych *Timestamp* w DB2 do typu *datetime2* w SQL Server.

Listing 1-1 Mapowanie typów danych w pliku *DB2ToMSSql10.xml*

```
<?xml version="1.0" encoding="utf-8" ?>
<dtm:DataTypeMappings
  xmlns:dtm="http://www.microsoft.com/SqlServer/Dts/DataTypeMapping.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  SourceType="DB2OLEDB;Microsoft.HostIntegration.MsDb2Client.MsDb2Connection"
  MinSourceVersion="*"
  MaxSourceVersion="*"
  DestinationType="SQLOLEDB;SQLNCLI*;System.Data.SqlClient.SqlConnection"
  MinDestinationVersion="10.*"
  MaxDestinationVersion="*">

...

  <!-- TIMESTAMP 10.* -->
  <dtm:DataTypeMapping>
     <dtm:SourceDataType>
        <dtm:DataTypeName>TIMESTAMP</dtm:DataTypeName>
     </dtm:SourceDataType>
     <dtm:DestinationDataType>
        <dtm:NumericType>
```

```
            <dtm:DataTypeName>datetime2</dtm:DataTypeName>
            <dtm:SkipPrecision/>
            <dtm:UseSourceScale/>
        </dtm:NumericType>
    </dtm:DestinationDataType>
  </dtm:DataTypeMapping>

...

</dtm:DataTypeMappings>
```

Uwaga W SQL Server Integration Services 2012 pliki mapujące instalowane są domyślnie pod adresem %Program Files%\Microsoft SQL Server\110\DTS\MappingFiles. Użytkownicy mogą modyfikować domyślne pliki mapujące, a także dodawać nowe pliki do tego folderu, aby zapewnić obsługę dla większej liczby źródeł lub miejsc docelowych. Nowe pliki mapujące muszą być zgodne z opublikowanym schematem XSD i dopasowywać typy dla unikatowych kombinacji źródła i miejsca docelowego.

Różne scenariusze wykorzystania kreatora *Import and Export Wizard* prowadzą do zróżnicowanych metod uruchamiania tego kreatora. Na przykład programista lub administrator bazy danych SQL Server, który chce zaimportować dane z pliku Microsoft Office Excel, może wywołać kreatora z narzędzia SQL Server Management Studio w kontekście docelowej bazy danych. Opcja ta pozwala mu zapisać pakiet skonstruowany przez kreatora i albo wywołać go od razu, albo odłożyć wykonanie na później.

Inną metodą wywołania kreatora jest wykorzystanie SQL Server Data Tools. Początkujący użytkownicy SSIS, którzy chcą rozpocząć od stosunkowo prostych konstrukcji pakietów przenoszenia danych, mogą wywołać kreatora w narzędziu Solution Explorer w SQL Server Data Tools, po czym dodać pakiet do aktualnego rozwiązania. Po dołączeniu pakietu można go dalej edytować, po czym zapisać jak każdy inny pakiet w rozwiązaniu SQL Server Data Tools.

Jednorazowe migracje często obejmują kopiowanie obiektów danych i samych danych z jednej instancji SQL Server do innej. SSIS wspiera taki scenariusz poprzez kilka zadań, których można użyć do przetransferowania baz danych, loginów, obiektów, procedur składowanych w bazie *master*, komunikatów błędów zdefiniowanych przez użytkownika albo zadań SQL Server Agent pomiędzy dwiema instancjami SQL Server. Wszystkie te zadania wykorzystują menedżera połączeń SQL Management Object (SMO) do utworzenia połączenia z instancjami SQL Server wykorzystywanymi w transferze.

Powtarzalne przenoszenie danych można realizować jako zadanie SQL Server Agent, nakazujące wykonywanie określonego pakietu SSIS w ustalonym harmonogramie.

Ładowanie danych do hurtowni

SSIS stanowi podstawowe narzędzie ETL oraz platformę dla tradycyjnej hurtowni danych (*Data Warehouse* – DW). Wypełnianie DW jest jednym z najbardziej popularnych zastosowań SSIS w przedsiębiorstwach. W hurtowniach danych źródłowe informacje są przenoszone z operacyjnych magazynów do centralnej lokalizacji, która jest optymalizowana pod kątem analizy i raportowania. Tego typu ładowanie może być wykonywane przyrostowo albo jako masowe odświeżanie i zazwyczaj obejmuje następujące operacje:

- Wydobycie wszystkich lub tylko zmienionych (nowych) danych z wielu źródeł
- Transformacja wydobytych danych przed załadowaniem do miejsca docelowego
- Ładowanie tabel wymiarów i faktów w miejscu docelowym
- Wyszukiwanie danych referencyjnych
- Generowanie kluczy
- Zarządzaniem zamianami historycznymi

SSIS może zostać skutecznie wykorzystane do implementacji wszystkich tych operacji. Wydobywanie danych z wielu źródeł, transformowanie ich i ładowanie tabel zostało już skrótowo przedstawione wcześniej w tym rozdziale. Pobieranie zmienionych danych stanowi treść kolejnego podrozdziału. W bieżącym podrozdziale zajmiemy się pozostałymi operacjami ładowania DW.

Wyszukiwanie danych referencyjnych angażuje pobieranie danych z zewnętrznego zbioru odnośników. Na przykład mając identyfikator klienta proces ładowania DW może musieć odczytać dodatkowe dane o tym kliencie, takie jak kod pocztowy z serwera CRM (*customer relations management* – zarządzanie relacjami z klientami) przechowującego wszystkie dane na temat klientów. W takim przypadku zewnętrzny zbiór danych na serwerze CRM jest używany jako dane referencyjne, umożliwiając dołączenie dodatkowych danych do potoku ETL. SSIS wspiera ten krok przetwarzania DW poprzez wykorzystanie komponentu Lookup Transform. Zbiór referencyjny może być istniejącą tabelą lub widokiem w relacyjnej bazie danych, wynikiem kwerendy SQL albo plikiem Lookup Cache. Wykonanie operacji wyszukiwania wymaga umieszczenia danych referencyjnych w pamięci. W przypadku wielkich zbiorów referencyjnych dane te mogą zostać wstępnie załadowane do specjalnego typu pliku nazywanego *Lookup Cache* w celu zapewnienia wysokiej wydajności. Transformacje wyszukiwania wykorzystują albo menedżera połączeń OLE DB, albo menedżera *Lookup Cache* do połączenia z referencyjnym zbiorem danych. Transformacja wyszukiwania odbywa się poprzez złączenie danych w kolumnach wejściowych z kolumnami w referencyjnym zbiorze danych. Komponent *Lookup Transform* może wykorzystywać wiele źródeł (wejść) w operacji wyszukiwania. Wejścia wyszukiwania nie mogą zawierać pewnych szczególnych typów danych, takich jak obrazy lub tekst. Dla obsługiwanych typów łańcuchowych operacje wyszukiwania rozróżniają wielkość liter; jeśli konieczne jest

zignorowanie tej właściwości, konieczne jest jawne użycie transformacji typu *Character Map* w celu przekonwertowania wejścia na same wielkie lub małe litery, aby dopasować wielkość liter do używanej w danych referencyjnych. Dane wejściowe niedopasowane do danych referencyjnych mogą zostać przekierowane przy użyciu transformacji *Lookup*. Możliwość połączenia komponentu Lookup Transform z relacyjnymi bazami danych jest ograniczona do łączności OLE DB, przy czym wspieranymi źródłami danych są SQL Server, Oracle oraz DB2. Jeśli referencyjne źródło danych nie wspiera połączeń OLE DB, tworzone jest zadanie *Data Flow* (przepływu danych) wykorzystujące dowolny wspierany adapter źródłowy, zaś do utworzenia pliku buforowego wykorzystywana jest transformacja *Cache* w kolejnym zadaniu *Data Flow*.

Uwaga Transformacje wyszukiwania obsługują tylko dokładne dopasowania ze zbiorem referencyjnym. Istnieje również transformacja *Fuzzy Lookup* (wyszukiwania rozmytego), omówiona w kolejnym podrozdziale, która wspiera niedokładne (częściowe) dopasowania. W wersji SQL Server 2005 nie było obsługiwane przekierowywanie niedopasowanych danych wejściowych.

Generowanie unikatowych kluczy i wykorzystywanie ich do zastępowania naturalnego klucza podstawowego w tabelach wymiarów w hurtowniach danych jest typowym działaniem. Klucze takie często określane są mianem *surogatów* (kluczy zastępczych) lub *kluczy sztucznych* i mogą być bardzo przydatne przy aktualizowaniu wymiarów w sytuacjach, gdy naturalne klucze podstawowe mogą się zmieniać. Gdy klucz naturalny jest alfanumerycznym lub złożonym kluczem zastępczym, można wykorzystać klucz całkowitoliczbowy w celu poprawy wydajności. Generowanie kluczy zastępczych w procesach ETL jest uważane za lepsze rozwiązanie, niż stosowanie generowania kluczy w miejscu docelowym (na przykład poprzez wykorzystanie funkcji klucza tożsamości w SQL Server) podczas wstawiania wierszy, gdyż integralność referencyjna tworzona w miejscu docelowym przy użyciu kluczy generowanych po tej stronie może zostać zerwana podczas przenoszenia danych. Klucze zastępcze, przeciwnie, zapewniają przenośność danych. W SSIS nie istnieje wbudowana funkcjonalność generująca takie klucze, ale stosunkowo łatwo można ją zaimplementować. Skrypty lub niestandardowe rozszerzenia to typowe podejścia do tworzenia generatorów kluczy zastępczych. Generowanie kluczy zastępczych zazwyczaj obejmuje pobranie maksymalnej wartości klucza zastępczego aktualnie używanego w interesującej nas tabeli, po czym wykorzystanie jej jako inicjatora dla przypisania klucza każdemu wierszowi w potoku przepływu danych ze wstępnie ustalonymi przyrostami. Niektórzy projektanci SSIS preferują obsługiwanie wartości inicjatora wewnątrz ich systemów ETL bez potrzeby odpytywania bazy danych na początku każdego procesu ETL.

W hurtowniach danych dane wymiarowe mogą zmieniać się z upływem czasu. Tego typu wymiary są zwykle określane jako wymiary wolnozmienne (*slowly changing dimensions* – SCD). Przetwarzanie SCD i zarządzanie zmianami historycznymi należą

do trudniejszych kroków w operacjach ładowania DW. Istnieją trzy często spotykane typy SCD:

- **Typ 1** Starsze dane są nadpisywane i zmiany historyczne nie są zachowywane. Ten typ zmian jest często stosowany wobec danych wymiarowych, które nie są już poprawne i gdy wartości historyczne nie niosą żadnej wartości biznesowej.
- **Typ 2** Zmiany historyczne są zachowywane i dla każdej nowej wartości dodawany jest nowy wiersz. Jest to najczęściej spotykany typ zmian w wymiarach DW. Każdy wiersz (bieżący i historyczny) wartości wymiaru będzie miał swój własny klucz zastępczy, numer wersji lub sygnaturę czasową, których można użyć do uzyskania najbardziej aktualnej wartości.
- **Typ 3** Tworzone są kolumny służące do przechowania wartości bieżącej i historycznej. Metodę tę można zastosować, gdy zmiany występują jedynie wyjątkowo albo we wstępnie zdefiniowanych odstępach czasu, a także gdy zachodzi potrzeba utrzymywania jedynie kilku najbardziej niedawnych wartości historycznych.

SSIS wspiera takie operacje przy użyciu transformacji *Slowly Changing Dimension*, która koordynuje aktualizację i wstawianie rekordów do tabel wymiarów hurtowni danych. Transformacja ta wspiera cztery typy zmian:

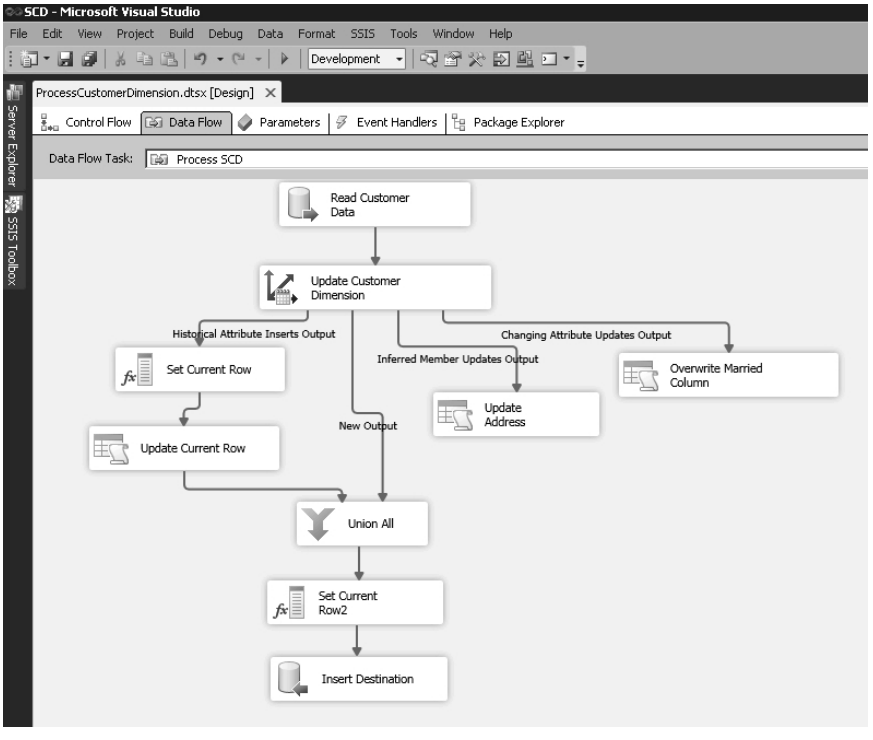
- **Zmienianie atrybutu** Realizuje typ 1 SCD opisany wcześniej.
- **Historyczne atrybuty** Realizuje typ 2 SCD opisany wcześniej.
- **Ustalony atrybut** Próba zmiany zwraca sygnał, że wartość kolumny nie może być zmieniona. Wiersze, których dotyczy próba zmiany wartości, są przekierowywane do dalszego przetwarzania.
- **Indukowany element** Tworzony jest rekord zastępczy, gdy wartości wymiarów nie są jeszcze dostępne.

Zmiany typu 3 nie są wspierane przez transformacje SCD, ale mogą zostać obsłużone przy użyciu kombinacji innych komponentów SSIS. Transformacja *Slowly Changing Dimension* wykorzystuje jedno wejście i do sześciu wyjść. Każde wyjście odpowiada aktualizacji, wstawieniu lub innemu wymaganiu przetwarzania rekordu w tabeli wymiarów w miejscu docelowym. Podczas wykonywania transformacja SCD identyfikuje przychodzące wiersze z dopasowaniami w tabeli wyszukiwania przy użyciu własnego menedżera połączeń. Po znalezieniu dopasowania SCD identyfikuje typ aktualizacji dla każdego wiersza i kolumny, po czym przekierowuje wiersz do właściwego wyjścia w celu odpowiedniego obsłużenia zmiany. Na przykład w trybie przetwarzania wymiarów typu 1 transformacja SCD przekaże wiersz do wyjścia *Changing Attributes Updates* (aktualizacja zmieniająca atrybuty), które jest połączone z transformacją *OLE DB Command*, aktualizującą rekord w tabeli wymiarów za pośrednictwem wyrażenia SQL UPDATE.

Uwaga Transformacja *OLE DB Command* w SSIS jest komponentem opartym na wierszach i może mieć znaczący (negatywny) wpływ na wydajność przetwarzania SCD. Liczba zmian wartości wymiarów do przetworzenia wymaga równej liczby wywołań do bazy danych, co oczywiście nie jest wydajnym podejściem. Istnieją alternatywne sposoby projektowania lepszego przetwarzania SCD, w tym:

- Wykorzystanie wyrażenia SQL MERGE dla prostych zmian.
- Niestandardowe komponenty w przypadku przetwarzania wielkich zbiorów wymiarów.
- Komponent SSIS Dimension Merge SCD, dostępny do pobrania ze strony *http://www.codeplex.com*.

Konstruowanie i konfigurowanie kolejnych kroków przetwarzania SCD może stać się bardzo złożone przy projektowaniu ETL. SSIS udostępnia kreatora, który ułatwia projektantom przejście przez standardowe kroki zarządzania SCD. Kreator tworzy transformacje dla przetwarzania SCD w zadaniu *Data Flow* i działa tylko z tabelami wymiarów zawartymi w bazach danych SQL Server. Rysunek 1-3 ukazuje typowe składniki SSIS zaangażowane w przetwarzanie SCD. Wszystkie widoczne komponenty (oprócz adaptera źródłowego) zostały dodane przez kreatora.



Rysunek 1-3 Przetwarzanie wolnozmiennych wymiarów w SSIS

Czyszczenie, formatowanie lub standaryzowanie danych

W większości organizacji dane występują w najrozmaitszych formach, kształtach, a także o różnej jakości. Różne części przedsiębiorstwa mogą stosować odmienne konwencje i formaty. Przy transakcjach pomiędzy firmami uzyskiwane są dane z innych organizacji, które mogą korzystać z różniących się magazynów danych i stosować odmienne standardy jakościowe. Globalne organizacje utrzymują dane w różnych językach w celu obsłużenia lokalnych potrzeb biznesowych w różnych regionach geograficznych. Często też dochodzi do uszkodzenia danych podczas transakcji i takie dane muszą zostać wyizolowane podczas przetwarzania.

Procesy integracji danych muszą radzić sobie z takimi problemami, gromadząc całość danych i gwarantując, że są one spójne, zanim zostaną skonsolidowane w środowisku integracyjnym lub załadowane do docelowych magazynów czy hurtowni danych. Większość narzędzi integrowania danych oferuje funkcje pomagające radzić sobie z brudnymi danymi, w tym błędami pisowni, niedokładnymi lub nieprawidłowymi datami, duplikatami czy nieoczekiwanymi skrótami.

SSIS udostępnia szereg opcji czyszczenia danych przydatne dla różnych potrzeb klientów, w tym komponenty transformacji *Fuzzy Lookup* (wyszukiwanie rozmyte) oraz *Fuzzy Grouping* (grupowanie rozmyte), które działają jako ogólne operacje przetwarzania danych bez konieczności dysponowania zbiorem reguł zależnych od kontekstu na poziomie eksperckim. *Fuzzy Lookup* ułatwia dopasowanie przychodzących, potencjalnie niskiej jakości danych do wyczyszczonych i standaryzowanych danych referencyjnych. Zwraca najbliższe dopasowanie w danych referencyjnych i ocenę jakości tego dopasowania. *Fuzzy Grouping* pomaga w identyfikowaniu grup wierszy w przychodzących danych, które prawdopodobnie odwołują się do tej samej jednostki w kolumnie znakowej, co pozwala na wykrycie duplikujących się danych. W wersji SQL Server 2012 SSIS udostępnia transformację *Data Quality Services* (DQS) *Cleansing*. Transformacja ta jest stosowana do wykonywania korekcji danych i likwidowania duplikatów przy użyciu baz wiedzy zbudowanych za pomocą DQS. Podczas wykonywania praca oczyszczająca odbywa się na serwerze DQS z wykorzystaniem baz wiedzy wskazanych w transformacji. DQS nie stanowi części SSIS. Jest to, podobnie jak SSIS, odrębny komponent linii produktów Microsoft SQL Server, zapewniający opartą na wiedzy funkcjonalność oczyszczania danych.

Uwaga Komponenty transformacyjne *Fuzzy Grouping* oraz *Fuzzy Lookup* nie są dostępne we wszystkich wydaniach SQL Server. Szczegóły na temat poszczególnych wydań SQL Server i dostępnych w nich funkcji SSIS zawarte są w dalszej części tego rozdziału.

W uzupełnieniu do opisanych transformacji czyszczących specjalnego stosowania w SSIS można wykonywać standaryzowanie i formatowanie danych przy użyciu następujących funkcji:

- **Transformacja Character Map** Umożliwia stosowanie funkcji łańcuchowych do danych znakowych.
- **Transformacja Data Conversion** Konwertuje dane z kolumny wejściowej na inny typ danych.
- **Transformacja Derived Column** Tworzy nowe wartości kolumny poprzez wykonanie działań (wyrażeń) wobec danych kolumn wejściowych.
- **Porównywanie i zastępowanie danych** Funkcje wykorzystywane w wyrażeniach, które są obliczane wobec kolumn wejściowych.

Oczyszczanie i manipulowanie formatami jest użyteczne, jednak w większości przypadków konieczne jest doskonałe zrozumienie natury danych, zanim jakikolwiek typ takiego przetwarzania zostanie zastosowany. SSIS udostępnia funkcjonalność o nazwie *Data Profiling* (profilowanie danych), które kompiluje różne statystyki na temat danych i może być pomocne w identyfikowaniu potrzeb czyszczenia i minimalizowania problemów związanych z jakością danych. Zadanie to jest konfigurowane do przetwarzania jednego lub kilku profili. Wynik, który jest zwracany w formacie XML, może zostać zapisany w pliku lub w zmiennej SSIS. Wyniki profilowania zapisane w pliku można przeglądać przy użyciu narzędzia Data Profiler Viewer. Możliwe jest sterowanie przepływem pracy w pakietach SSIS przy użyciu wyników zadania profilującego.

Uwaga Zadania *Data Profiling* działają tylko wobec danych przechowywanych w bazach SQL Server 2000 lub wersjach późniejszych.

Rozdział 9, „Czyszczenie i profilowanie danych przy użyciu SSIS", zawiera bardziej szczegółowe omówienie wszystkich funkcji zapewniania jakości danych i oczyszczania dostępnych w SSIS.

Identyfikowanie, przechwytywanie i przetwarzanie zmian danych

Nieustannie rosnące zbiory danych, zapotrzebowanie na raporty w czasie rzeczywistym i zmniejszające się rozmiary okien czasowych przeznaczonych na przetwarzanie danych – wszystkie te czynniki powodują, że użytkownicy domagają się funkcjonalności wykrywania i przetwarzania zmian w narzędziach integrowania danych. Przetwarzanie integrowania danych jest wydajne, jeśli jest wykonywane wobec przyrostowych zbiorów danych, a nie wobec całości danych dostępnych w zaangażowanych magazynach danych. Przyrostowe przetwarzanie danych redukuje czas wykonywania procesów integracyjnych, co z kolei pozwala zwiększyć częstotliwość uruchamiania tych procesów. Użycie sygnatur czasowych, sum kontrolnych lub rozwiązań opartych na funkcjach haszujących do wykrywania zmian i ich przechwytywania jest typową praktyką branżową. Stosunkowo niedawną i popularną alternatywą oferowaną przez

wielu dostawców baz danych jest wbudowana zdolność do identyfikowania zmienionych danych. Narzędzia integracji danych mogą posługiwać się tymi funkcjami do przechwytywania zmian w celu przyrostowego przetwarzania danych. Na przykład SQL Server udostępnia funkcje *Change Data Capture* (CDC) (przechwytywanie zmian danych) oraz *Change Tracking* (CT) (śledzenie zmian), zaś SSIS udostępnia wbudowane i niestandardowe opcje przetwarzania zmienionych danych, które mogą wykorzystać funkcjonalność CDC, jeśli jest używana.

Projektanci rozwiązań integrowania danych mogą posłużyć się infrastrukturą SQL Server CDC bezpośrednio w trakcie budowania pakietów. Funkcjonalność CDC w SQL Server jest bardzo efektywna, ale jest również złożona i pociąga za sobą konieczność stosowania znaczących ilości niestandardowego kodu. W celu usprawnienia przetwarzania CDC SSIS udostępniają następujące zadania i komponenty:

- Zadanie *CDC Control*
- *CDC Source* (Źródło CDC)
- Transformacja *CDC Splitter* (Rozdzielacz CDC)

Uwaga Funkcje CDC oraz CT są dostępne począwszy od wersji SQL Server 2008. Zadania i komponenty przetwarzania CDC w SSIS są dostępne dopiero w wersji SQL Server 2012.

Zadanie *CDC Control* jest przydatne do sterowania różnymi fazami przetwarzania zmienionych danych w pakietach SSIS. Wymaga ono jednego menedżera połączeń dla bazy danych, w której zmiany mają zostać wykryte i przechwycone, oraz drugiego (opcjonalnego) do zapewnienia trwałości stanu operacji przetwarzania CDC przechowywanego w zmiennej SSIS. Fazy przetwarzania CDC zarządzane przez SSIS za pośrednictwem tego zadania obejmują oznaczanie początku i końca wstępnego ładunku danych, rozpoczęcie operacji oraz ustalanie zakresu przetworzonych danych. Komponent *CDC Source* umożliwia ekstrakcję zmienionych wierszy z określonego zakresu do przetworzenia. Wiersze te mogą zostać przejęte przez zadanie kontrolne. Składnik źródłowy wykorzystuje artefakty bazodanowe generowane przez SQL Server podczas instalowania CDC w bazie danych, która ma być śledzona pod kątem zmian. Transformacja *CDC Splitter* kieruje zmienione dane wyselekcjonowane przez CDC Source do trzech kategorii wyjścia – Insert, Update oraz Delete – stosując odmienną logikę przetwarzania do każdej kategorii. Rozdział 8, „Korzystanie z Change Data Capture w SSIS 2012", zawiera bardziej szczegółowe omówienie tych komponentów. Jak wspomniano wcześniej, komponenty te wspierają tylko bazy danych SQL Server. SSIS zapewnia wsparcie CDC w bazach danych Oracle przy użyciu usługi Windows, która imituje zmiany Oracle w bazie danych SQL Server, tym samym umożliwiając przetwarzanie zmian poprzez wykorzystanie zadań i komponentów CDC.

Koordynowanie konserwowania, przetwarzania lub analizowania danych

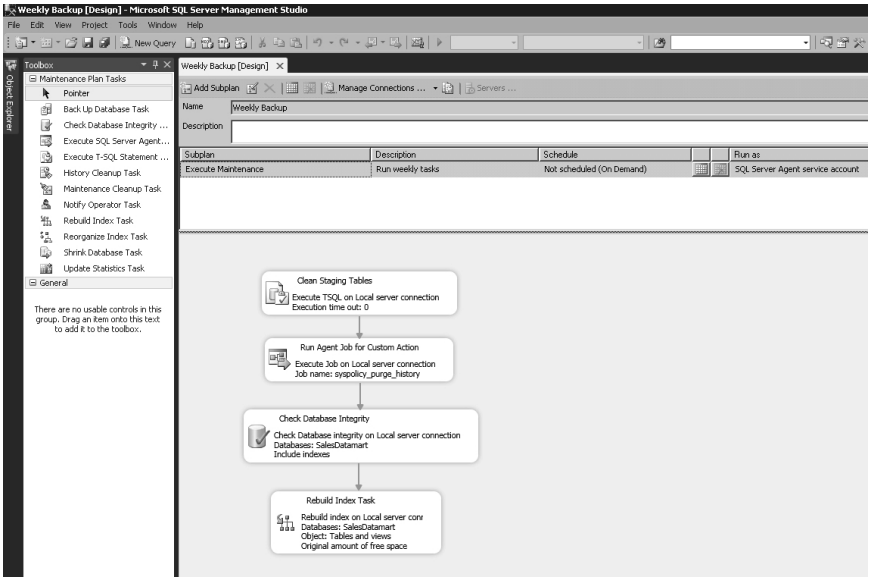
Zadanie *Data Flow* (przepływu danych), które wspiera wydobywanie, transformowanie i ładowanie danych opisane we wcześniejszych podrozdziałach, jest wykonywane w kontekście przepływu pracy zdefiniowanego w sekcji *Control Flow* (przepływu sterowania) pakietu SSIS. Sekcja *Control Flow* w SSIS jest zorientowana na zadania i umożliwia koordynowanie wykonywania zadań przetwarzania danych w procesie biznesowym. Oprócz zadań *Data Flow* specjalnego stosowania, obsługiwanych przez oparty na buforuch motor przepływu danych, SSIS zawiera wiele wbudowanych zadań, których można użyć przy budowaniu przepływu kontrolnego. Zadania te, wykonywane przez motor SSIS, są użyteczne przy projektowaniu takich operacji, jak typowe działania administracyjne w bazach danych, przygotowywanie wykonania przepływu danych, wykonywanie poleceń Analysis Services i wielu innych operacji typowych w scenariuszach integrowania danych. Możliwe jest również budowanie niestandardowych zadań przy użyciu modelu programistycznego SSIS i wykorzystanie tych elementów jako części *Control Flow*. Inna możliwość to wykonywanie niestandardowych skryptów.

Zadania mogą być umieszczane w kontenerach należących do jednego z trzech typów. Kontener *Sequence* (Sekwencja) służy do grupowania zadań i kontenerów w celu zarządzania nimi jako pojedynczą jednostką. Kontener *For Each Loop* (Dla każdego) obsługuje powtarzanie kroków w przepływie pracy poprzez wyliczanie plików lub obiektów, zaś kontener *For Loop* (pętla For) udostępnia inną opcję powtarzania kroków, wykorzystującą wyrażenia warunkowe. Kontenery te mogą mieścić w sobie obok zadań również inne kontenery (zagnieżdżanie). Zadania i kontenery w *Control Flow* są łączone ze sobą ograniczeniami pierwszeństwa, które determinują ścieżkę wykonania w procesie pracy. Ograniczenia te definiują kolejność, w jakiej zadania i kontenery zadań są wykonywane lub warunki określające, która część przepływu pracy zostanie wykonana w następnej kolejności. Prostota korzystania z tych ograniczeń sprawia, że wstępna instrumentacja kroków w przepływie SSIS jest łatwa do zbudowania, debugowania i zarządzania.

SSIS zawiera również zadania do realizowania operacji konserwowania baz danych. Zadania te są użyteczne przy budowaniu planów konserwacji bazy danych w SQL Server Management Studio, jak również w SQL Server Data Tools, wraz z innymi zadaniami, które można wykorzystać podczas konstruowania sterowania przepływem w SSIS. Rysunek 1-4 pokazuje moduł projektowania SQL Server Management Studio w trakcie budowania planu konserwacji bazy danych przy użyciu zadań konserwacyjnych SSIS. Wśród popularnych zadań konserwacji baz danych można wymienić:

- **Zadanie Backup Database** Wykonuje kopie zapasowe baz danych SQL Server.
- **Zadanie Rebuild Index** Odbudowuje indeksy w tabelach i widokach baz danych SQL Server.

- **Zadanie Update Statistics** Aktualizuje informacje o dystrybucji wartości kluczy dla jednego lub więcej zbiorów statystyk w określonej tabeli lub widoku.
- **Zadanie Shrink Database** Redukuje rozmiary plików danych i dzienników bazy danych SQL Server.



Rysunek 1-4 Zadania konserwacji baz danych w SQL Server Management Studio

Zadania dostępne do budowy *Control Flow* są zazwyczaj wykorzystywane do przygotowania wykonania zadania *Data Flow*. Na przykład zadanie *Execute SQL* (Wykonaj SQL) służy do wykonania wyrażeń SQL w magazynie danych. Zadanie to jest używane dla takich operacji, jak tworzenie lub usuwanie tabel, przygotowywanie pośredniczącej bazy danych, uzyskiwanie maksymalnej wartości kolumny tożsamości w tabeli, wykonywania procedur składowanych czy odczytywania liczby wierszy z arkusza. Zadanie *Execute SQL* może zostać użyte wobec szerokiego zakresu źródeł danych i wspiera w tym celu wiele menedżerów połączeń. Pobieranie plików danych z zewnętrznych systemów do serwerów integracji danych jest typowym krokiem przygotowawczym w operacjach ładowania danych. SSIS oferuje kilka opcji dla takich operacji, z których najbardziej popularne to:

- **Zadanie File System** Wykonuje operacje na plikach i katalogach systemu plików. Na przykład zadania tego można użyć do pobrania plików danych ze zdalnego udziału plikowego i skopiowania ich do nowo utworzonego katalogu w lokalnym systemie plików.
- **Zadanie FTP** Pobiera pliki danych z serwera FTP, ładuje pliki na serwer FTP lub zarządza katalogami na serwerze. Zadanie to może zostać wykorzystane

do pobrania danych z lokalizacji FTP do przetwarzania ETL i usunięcia oryginalnego pliku z tej lokalizacji po zakończeniu pobierania.

- **Zadanie Web Service** Realizuje metodę usługi sieci Web. Zadanie to może zostać użyte do pobrania danych z usługi Web, które następnie mogą zostać zapisane w zmiennej lub pliku.

Jeśli plik danych uzyskany przy użyciu jednego z tych zadań ma format XML, możemy wykorzystać zadanie XML dostępne w SSIS do jego przetworzenia. Zadanie to pozwala przeformatować dane przy użyciu mechanizmu XSLT, wybierać węzły XML za pomocą zapytań XPath lub scalać wiele dokumentów XML. Jeśli plik danych jest plikiem tekstowym lub XML i dane mają zostać załadowane do bazy SQL Server bez żadnych transformacji, możemy posłużyć się zadaniem *Bulk Insert*, które opakowuje wyrażenie BULK INSERT dostępne w SQL Server. Zapewnia to wydajną metodę kopiowania wielkich ilości danych do tabel lub widoków SQL Server.

SSIS zawiera dwa zadania Analysis Services służące do wykonywania operacji w bazach danych Microsoft SQL Server Analysis Services. Zadanie *Analysis Services Execute DDL* używane jest do wykonywania wyrażeń Data Definition Language (język definiowania danych), które pozwalają tworzyć, usuwać lub modyfikować modele drążenia, kostki lub wymiary Analysis Services. Zadanie *Analysis Services Processing* służy do przetwarzania takich artefaktów, gdy już zostaną one utworzone.

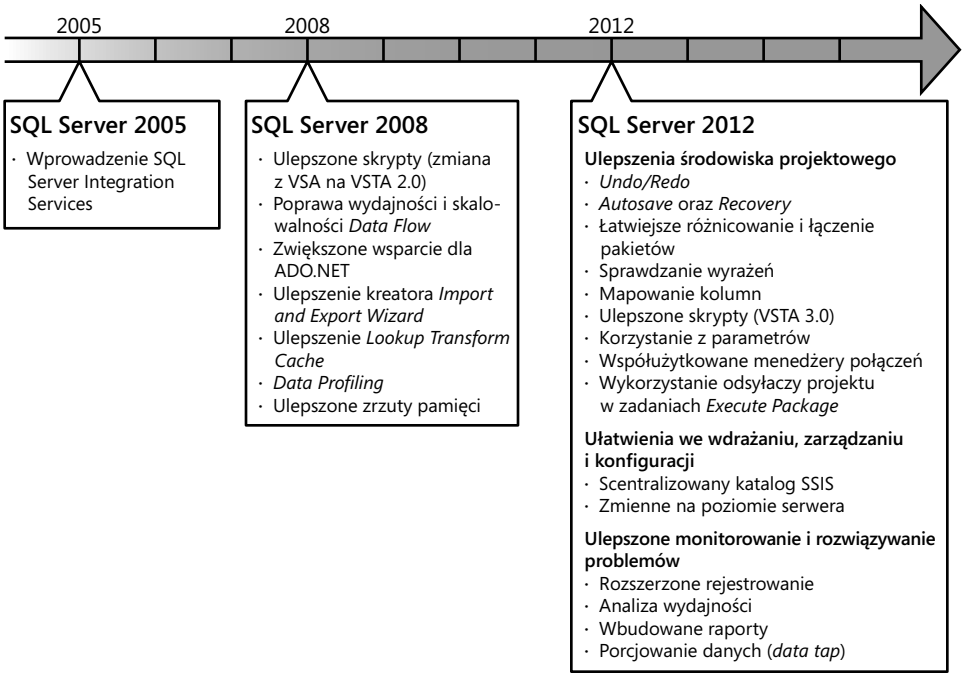
Dowolna funkcjonalność konserwacyjna lub procesowa, która nie jest dostępna wprost w SSIS, może zostać zaimplementowana przy użyciu zadania *Script*. Na przykład, jeśli przygotowania dla zadania *Data Flow* wymagają pobrania danych ze źródła, dla którego w SSIS nie istnieje wbudowany komponent źródłowy lub gdy określony krok przetwarzania danych, który musi być włączony do *Control Flow*, nie jest dostępny w SSIS, skryptowanie może rozwiązać ten problem.

Pakiety SSIS mogą zostać skonfigurowane do wznawiania wykonywania z miejsca, w którym wystąpił błąd, jeśli wykonywanie *Control Flow* się nie powiedzie. Wznawianie (restart) pakietów sterowane jest poprzez wykorzystanie plików punktów kontrolnych (*checkpoint*). Jeśli pakiet używa plików punktów kontrolnych, informacja o wykonywaniu pakietu jest zapisywana w pliku. Dzięki temu, jeśli pakiet jest wznawiany po błędzie, wykonywanie może rozpocząć się od tego zadania lub kontenera, w którym wystąpił błąd. Punkty kontrolne SSIS są szczególnie użyteczne, gdy chcemy uniknąć niepotrzebnego powtarzania intensywnych operacji, które już zakończyły się sukcesem, gdy ponownie uruchamiamy pakiet. Przykładem może być kopiowanie wielkich plików lub wykonywanie czasochłonnych przepływów danych.

Operacje administracyjne i działania konserwacyjne często powinny być wykonywane w regularnych odstępach czasu. SQL Server Agent jest narzędziem, które umożliwia zautomatyzowanie wykonywania pakietów SSIS.

Ewolucja SSIS

Firma Microsoft wprowadziła składnik Integration Services w wersji Microsoft SQL Server 2005. Dostępny w SQL Server 2000 składnik Data Transformation Services (DTS) można traktować jako poprzednika SSIS; jednak SSIS różni się zasadniczo od DTS zarówno pod względem koncepcji, jak i interfejsu użytkownika, zbioru dostępnych funkcjonalności i wewnętrznej architektury. Przejście od stosunkowo prostego narzędzia ETL, jakim było DTS, do platformy integracji danych, takiej jak SSIS, zostało bardzo dobrze przyjęte przez klientów korzystających z SQL Server. Niemal natychmiastowa czytelność i podobieństwo do znanych już środowisk projektowych spodobało się programistom i projektantom, zaś cechy skalowalności i wydajności zaspokoiły potrzeby przedsiębiorstw. Wykorzystanie SSIS w wielkoskalowych projektach integrowania danych stale wzrasta od momentu jego wprowadzenia na rynek. Rysunek 1-5 przedstawia kilka szczegółów dotyczących usprawnień i funkcji wprowadzanych w różnych wersjach SSIS.



Rysunek 1-5 Ewolucja SSIS

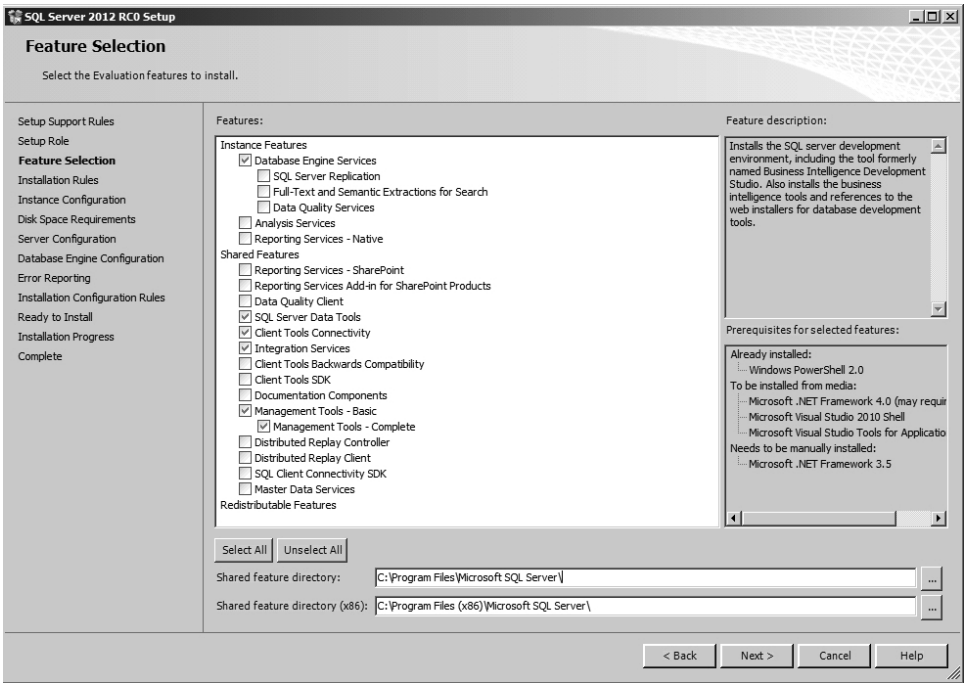
Uwaga Szczegóły na temat ulepszeń motoru przepływu danych dostępnych w różnych wersjach zawiera rozdział 15, „Motor SSIS".

Instalowanie SSIS

Wszystkie funkcje SQL Server – w tym SSIS – można zainstalować przy użyciu pojedynczego programu instalacyjnego. Funkcjonalności niezbędne do budowania, zarządzania lub wykonywania rozwiązań SSIS są rozproszone po wielu elementach SQL Server i muszą zostać zainstalowane odpowiednio. Rysunek 1-6 ukazuje ekran wyboru funkcji programu instalacyjnego SQL Server. Główne moduły w programie instalacyjnym SQL Server dotyczące SSIS to:

- Integration Services
- SQL Server Data Tools
- Database Engine Services
- Management Tools

Uwaga Najlepszą (zalecaną) praktyką jest skonfigurowanie dedykowanego serwera dla wszystkich potrzeb integrowania danych. Firma Microsoft zaleca na takich serwerach modułów instalację Integration Services oraz Database Engine Services.



Rysunek 1-6 Ekran wyboru funkcji w programie SQL Server Setup

Funkcje SQL Server wymagane dla integrowania danych

Ten podrozdział zawiera skrótowe omówienie funkcji, które trzeba wybrać podczas instalacji SQL Server, aby móc budować i uruchamiać rozwiązania integrowania danych.

Integration Services

Wybór ten powoduje zainstalowanie SSIS *runtime*, niektórych narzędzi (plików .exe), usługi Windows o nazwie SQL Server Integration Services (uzupełnionej o numer wersji) oraz rozmaite biblioteki (pliki .dll) umożliwiające wykonywanie pakietów SSIS poza środowiskiem projektowym, czyli SQL Server Data Tools. Usługa Windows zarządza lokalnie wykonywanymi pakietami oraz pakietami przechowywanymi w bazie danych *msdb* w instancji SQL Server. Funkcja Integration Services jest dostępna w węźle Shared Features (funkcje wspólne) w drzewie funkcji SQL Server pokazanym na rysunku 1-6, gdyż nie jest ona specyficzna dla instancji bazy danych SQL Server; mówiąc inaczej, funkcja ta jest współdzielona przez wszystkie instancje serwera. Tak więc, nawet jeśli komputer zawiera kilka instancji bazy danych, potrzebna jest na nim tylko jedna kopia Integration Services. Pliki instalowane dla tej funkcji w wersji SQL Server 2012 umieszczane są w folderze *%Program Files%\Microsoft SQL Server\110\DTS*. W starszych wersjach hierarchia folderów dla tej funkcji wygląda analogicznie, ale z innym numerem wskazującym wersję SQL Server (110 w pokazanej ścieżce). Na przykład Integration Services dla SQL Server 2008 będzie zawierać numer 100 w ścieżce dostępu.

SQL Server Data Tools

Funkcja SQL Server Data Tools udostępnia środowisko projektowe do budowania pakietów SSIS. To graficzne narzędzie projektowania stanowi intuicyjne i łatwe w użyciu środowisko komponowania wszystkich działań związanych z integracją danych. Moduł projektowy zawiera odrębne obszary dla tworzenia przepływów pracy i dla budowania potoków przepływu danych. Budowanie przepływów jest proste i obejmuje dodawanie wbudowanych zadań lub komponentów do obszaru projektowego z przybornika SSIS, a następnie ich konfigurowanie i łączenie ze sobą. Usługi Integration Services nie muszą być instalowane na tej samej maszynie, aby możliwe było tworzenie pakietów SSIS za pomocą SQL Server Data Tools. Tym niemniej, jeśli Integration Services nie są zainstalowane, nie będzie możliwe wykonanie zaprojektowanych pakietów poza narzędziem SQL Server Data Tools na tym samym komputerze, na przykład przy użyciu narzędzia DTExec.

Stosowanie SQL Server Data Tools nie jest ograniczone jedynie do projektowania rozwiązań SSIS; zamiast tego jest to zintegrowane środowisko do budowania różnych rozwiązań opartych na SQL Server, obejmujących również Analysis Services,

Reporting Services czy projektowanie baz danych. Narzędzie SQL Server Data Tools jest hostowane wewnątrz Microsoft Visual Studio. Wszystkie te rozwiązania obsługują wspólne mechanizmy projektowe i wymagają środowiska programistycznego. Ponieważ Visual Studio jest domyślnym środowiskiem projektowym dla wszystkich technologii Microsoft, jest też naturalnym wyborem dla projektowania SSIS (jak dla innych). Visual Studio zapewnia projektantom dobrze znane środowisko, zaś narzędzie SQL Server Data Tools jest dobrze zintegrowane z innymi funkcjami Visual Studio, takimi jak Solution Explorer, Toolbox, panel Properties czy okna Output lub Watch. Łącznie zapewnia to prawdziwe, spójne środowisko budowania rozwiązań „od początku do końca". Możliwe jest wykonywanie pakietów SSIS wewnątrz SQL Server Data Tools i wykorzystanie innych użytecznych możliwości, takich jak ustawianie punktów przerwania lub debugowanie w trakcie wykonywania. Business Intelligence Development Studio oraz SQL Server Data Tools współpracują z określonymi wersjami Visual Studio i pakietów SSIS. Szczegółowe zestawienie zawiera tabela 1-2.

Tabela 1-2 Kompatybilność pomiędzy Visual Studio a Business Intelligence Development Studio/SQL Server Data Tools w SQL Server

Wersja SQL Server	Nazwa środowiska projektowego SSIS	Wersja Visual Studio dla Business Intelligence Development Studio/SQL Server Data Tools
SQL Server 2005	Business Intelligence Development Studio	Visual Studio 2005 z pakietami serwisowymi
SQL Server 2008	Business Intelligence Development Studio	Visual Studio 2008 z pakietami serwisowymi
SQL Server 2008 R2	Business Intelligence Development Studio	Visual Studio 2008 z pakietami serwisowymi
SQL Server 2012	SQL Server Data Tools	Visual Studio 2010 Service Pack 1+

Uwaga SSDT jest aplikacją 32-bitową. Pracuje w trybie WoW64 w 64-bitowych systemach operacyjnych. Business Intelligence Development Studio w wersjach SQL Server 2005, SQL Server 2008 oraz SQL Server 2008 R2 nie może być uruchamiane na 64-bitowych komputerach z architekturą Itanium.

Database Engine Services

Jako część tej funkcji instalowana jest instancja motoru bazy danych. Pakiety Integration Services mogą być wdrażane na komputerach z zainstalowanym oprogramowaniem SQL Server. W wersji SQL Server 2012 pakiety są rozmieszczane w katalogu SSIS, który jest dedykowaną bazą danych w instancji SQL Server. We wcześniejszych wersjach SSIS pakiety mogły być wdrażane w bazie danych *msdb* w SQL Server. Katalog SSIS używany jest do zarządzania i administrowania pakietami SSIS. Katalog obejmuje systemowe obiekty bazodanowe niezbędne do wdrożenia, konfiguracji i wykonywania pakietów SSIS w kontekście instancji SQL Server. Zawiera także obiekty bazodanowe do monitorowania lub raportowania stanu wykonywania oraz rozwiązywania problemów z danymi i wydajnością. Usługi motoru baz danych obejmują składnik SQL Server Agent, który umożliwia zaplanowane wykonywanie pakietów SSIS według harmonogramu. Instalacja tej funkcji SQL Server powoduje rozmieszczenie komponentów wymaganych przez kreatora *Import and Export Wizard* (jeśli nie zostały one już wcześniej dołączone do instalacji poprzez wybór funkcji Integration Services). *Import and Export Wizard* jest dołączany w celu zapewnienia możliwości pobierania i eksportowania danych do motoru bazodanowego bez jawnego dołączania funkcji Integration Services podczas instalacji SQL Server.

Management Tools

W ramach tej funkcji instalowany jest komponent SQL Server Management Studio. Jest to popularne narzędzie dla administratorów baz danych, zapewniające przyjazne środowisko dla zarządzania katalogiem SSIS, folderami projektowymi, zmiennymi serwerowymi i odnośnikami środowiska projektowego w Integration Services 2012. Po rozmieszczeniu pakietów możemy je konfigurować, weryfikować, uruchamiać i monitorować przy użyciu SQL Server Management Studio. Katalog SSIS jest po prostu bazą danych o nazwie SSISDB w instancji SQL Server, zatem dowolna funkcja dostępna w SQL Server Management Studio do interakcji lub zarządzania bazą danych może zostać użyta również wobec samego katalogu. SQL Server Management Studio można również wykorzystać do obsługi starszych usług SSIS lub agentów wykonujących pakiety SSIS. Na przykład projektując plan konserwacji w SQL Server Management Studio można uwzględnić zadania specyficzne dla SSIS.

Wydania SQL Server a funkcje Integration Services

Istnieje wiele wydań SQL Server, przy czym każde z nich powiązane jest z określonym scenariuszem wykorzystania. Funkcje dostępne w różnych wydaniach odpowiadają oczekiwanym zastosowaniom. Funkcjonalność SSIS zmienia się zależnie od wyboru wersji. Szczegółowe zestawienie zawiera tabela 1-3.

Tabela 1-3 Funkcje Integration Services dostępne w różnych wydaniach SQL Server 2012

Wydanie	Funkcje
Express Express Tools Express Advanced Web	• Import and Export Wizard • Funkcje wykorzystywane przez kreatora (SSIS runtime, podstawowe adaptery)
Standard Business Intelligence	Wszystkie funkcjonalności SSIS z wyjątkiem zaawansowanych funkcji (wymienionych w kolejnym wierszu tabeli)
Enterprise Evaluation Developer	• Data Mining Model Destination (Miejsce docelowe modelu drążenia danych) • Dimension Processing Destination (Miejsce docelowe przetwarzania wymiarów) • Adapter przetwarzania partycjonowanego • Adaptery SAP BW • Adaptery Oracle wysokiej prędkości • Adaptery Teradata wysokiej prędkości • Wydobywanie terminów i transformacje wyszukujące • Rozmyte wyszukiwanie i grupowanie • Kwerendy drążenia danych

Wydanie Business Intelligence występuje tylko dla wersji SQL Server 2012; nie jest ono dostępne w wersjach wcześniejszych. Z drugiej strony, wydanie Workgroup nie jest dostępne dla SQL Server 2012, ale osiągalne w wersjach wcześniejszych. To samo dotyczy wydania SQL Server Datacenter, dostępnego w wersji SQL Server 2008 R2, ale nie w SQL Server 2012. Jeśli konkretne wydanie nie jest dostępne w SQL Server 2012, program instalacyjny SQL Server może dokonać zmiany wydania podczas wykonywania aktualizacji do SQL Server 2012. Wydania SQL Server Developer oraz Evaluation mają ograniczenia dotyczące stosowania: wydanie Developer dopuszcza tylko zastosowania projektowo-programistyczne, zaś licencja wydania Evaluation jest ważna tylko przez 180 dni od daty instalacji. W wydaniach niższych niż Standard kreator *Import and Export Wizard* nie umożliwia zapisywania pakietów lub wykorzystania innych narzędzi, takich jak *Upgrade Wizard* lub DTExec; funkcje te są zablokowane. Niektóre z wyższych wydań oferują elastyczne modele licencjonowania, jednak zagadnienia te wykraczają poza tematykę tej książki.

Uwaga Strona *http://www.microsoft.com/sqlserver/en/us/editions.aspx* zawiera szczegółowe omówienie scenariuszy zastosowań dla różnych wydań SQL Server.

Podsumowanie

W tym rozdziale przedstawiliśmy wiele typowych scenariuszy integracji danych oraz ogólny przegląd funkcji SSIS, które odpowiadają potrzebom każdego z tych scenariuszy. SSIS spełnia większość wymagań związanych z budowaniem i realizowaniem złożonych rozwiązań integracji danych w przedsiębiorstwach. W rozdziale tym pokazaliśmy też funkcje SQL Server potrzebne do tworzenia pełnych rozwiązań SSIS oraz informacje, jakie funkcje są dostępne w poszczególnych wydaniach SQL Server.

ROZDZIAŁ 2

Koncepcja SSIS

Jak pokazaliśmy w rozdziale 1, „Ogólna charakterystyka SSIS", Microsoft SQL Server 2012 Integration Services stanowi wszechstronną platformę dla aplikacji integrowania danych. Jednak przy pierwszej publikacji tego produktu w wersji SQL Server 7.0 było to zaledwie proste narzędzie ekstrakcji-transformacji-ładowania (ETL) o nazwie Data Transformation Services (DTS). Z upływem lat firma Microsoft dodawała do produktu coraz to nowe i liczniejsze funkcje; dodatkowo architektura tego rozwiązania została zmieniona znacząco, aby mogło hostować nowoczesne aplikacje integrowania danych. W rezultacie Integration Services stawały się coraz skuteczniejsze, jednak kosztem stale rosnącej złożoności. Zanim zagłębimy się w rozważania na temat szczegółowych funkcjonalności i zastosowań, co nastąpi w dalszej części książki, ważne jest poznanie koncepcji produktu i różnych „klocków" składających się na niego.

Z punktu widzenia użytkownika Integration Services stanowią zbiór różnych programów i narzędzi. Klienci wykorzystują te narzędzia do projektowania i wykonywania aplikacji integrowania danych. Dla przykładu programiści zazwyczaj wykorzystują narzędzie SQL Server Data Tools do zaprojektowania pakietu, po czym jest on wykonywany przy użyciu narzędzia DTexec.exe lub harmonogramu serwera. Jeśli projektant ma jakieś doświadczenia w dziedzinie rozszerzania Integration Services, takie jak projektowanie niestandardowego zadania, zorientuje się, że Integration Services udostępniają również pewien zbiór bibliotek i asemblacji, które pozwalają budować najrozmaitsze rozszerzenia SSIS.

W rzeczywistości wszystkie te programy, narzędzia i rozszerzenia są zbudowane w oparciu o leżący w tle jednolity model obiektowy. Model ten zapewnia szkielet dla Integration Services i implementację większej części funkcjonalności. Wszystkie zewnętrzne programy lub narzędzia jedynie interpretują instrukcje użytkowników i następnie wywołują model obiektowy w celu wykonania pracy, tak więc model ten stanowi centrum Integration Services.

W tym rozdziale odbędziemy krótką wycieczkę po tym modelu, aby poznać następujące elementy:

- Trzy główne części modelu obiektowego, którymi są: przepływ sterowania, przepływ danych oraz katalog SSIS.
- Najważniejsze obiekty wchodzące w skład tych części oraz sposoby ich wzajemnych interakcji.

- Nowe koncepcje wprowadzone w wersji SQL Server 2012, w tym projekty, parametry, katalog SSIS, środowiskowe zmienne oraz odnośniki.

Przepływ sterowania

Przepływ sterowania (*control flow*) to największa część modelu obiektowego, a także część najbardziej fundamentalna. Będąc środowiskiem przepływu pracy, definiuje jednostki pracy, sposób wzajemnego współdziałania tych jednostek, kolejność i harmonogramy ich wykonywania i tak dalej. Wszystko, co potrzebujemy zrobić, można umieścić w tym środowisku jako jednostkę pracy, nawet niektóre bardzo złożone operacje, takie jak transformacje danych. Przepływ sterowania składa się głównie z motoru wykonawczego oraz pewnej infrastruktury pomocniczej. Motor odpowiada za porządek wywoływania poszczególnych jednostek pracy w trakcie wykonywania, podczas gdy infrastruktura zapewnia podstawowe wsparcie dla takich elementów, jak kontenery, zmienne i menedżery połączeń.

Omówienie przepływu sterowania rozpoczniemy od najbardziej podstawowej jednostki: zadania.

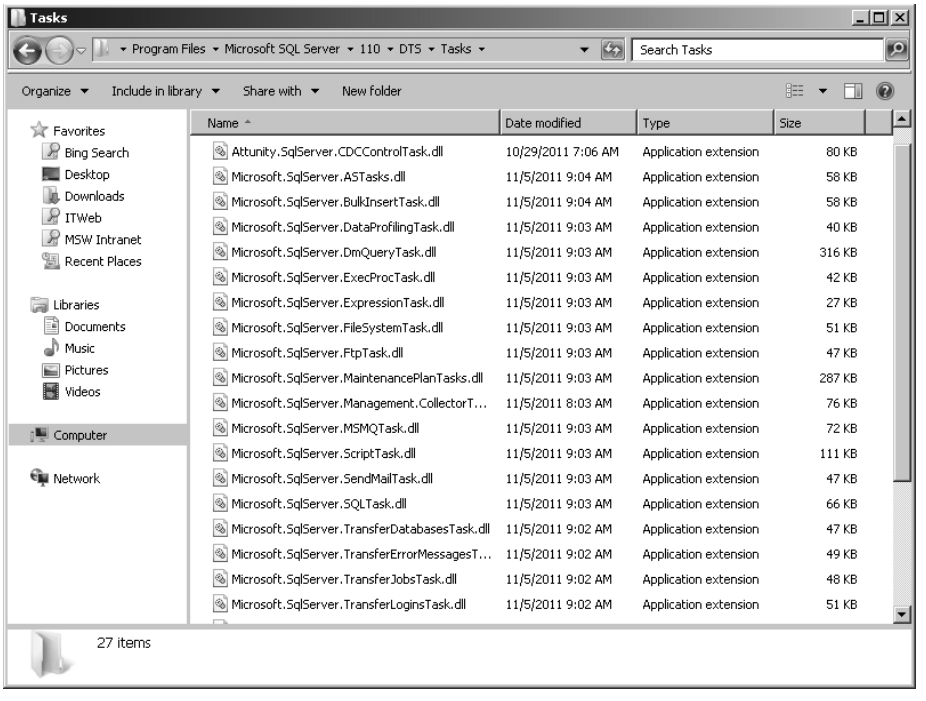
Zadania

Zadanie (*task*) to najmniejsza jednostka pracy w przepływie sterowania. Z punktu widzenia motoru przepływu sterowania zadanie jest atomową (niepodzielną) jednostką. Motor wywołuje zadanie we właściwym momencie i następnie czeka na jego zakończenie. Jako zadanie możemy zaimplementować dowolną czynność, która musi zostać wykonana, na przykład pobranie pliku z serwera FTP, wysłanie wiadomości email, utworzenie pliku dziennika lub wykonanie jakiejś transformacji danych.

Kiedy otworzymy okno Toolbox (Przybornik) w narzędziu Package Designer (Projektant pakietów), ujrzymy mnóstwo wbudowanych zadań. Każde z nich zaprojektowane zostało do spełnienia wymagań określonego scenariusza i udostępnia pewną liczbę właściwości, które należy skonfigurować w celu wykonania pracy. Na przykład zadanie *Execute Process* (Wykonaj proces) wymaga wskazania lokalizacji docelowej (pliku lub folderu), wybrania akcji do realizacji i tak dalej.

Zadanie nie jest powiązane („świadome istnienia") z innymi zadaniami zawartymi w tym samym pakiecie. Na przykład nie można utworzyć zadania skryptowego, które zmodyfikowałoby właściwości innych zadań w trakcie realizacji – co oznacza, że niemożliwe jest zbudowanie „samomodyfikującego się" pakietu. Istnieją dwie przyczyny, dla których takie zachowanie zostało zabronione: po pierwsze, takie zależności pomiędzy pakietami są bardzo trudne do zrozumienia i debugowania; po drugie, dla takich pakietów aktualizacja do przyszłych wersji SSIS byłaby bardzo trudna lub wręcz niemożliwa.

Większość wbudowanych zadań zostało napisane w językach należących do środowiska Microsoft .NET. Zadania takie instalowane są domyślnie w folderze *%Program Files%\Microsoft SQL Server\110\DTS\Tasks* (patrz rysunek 2-1).



Rysunek 2-1 Zarządzane zadania w folderze instalacyjnym

Istnieje także pewna liczba innych zadań, napisanych w językach natywnych. Przykładami mogą być zadania *Execute SQL* lub *Execute Package*. Te natywne zadania instalowane są w folderze *%Program Files%\Microsoft SQL Server\110\DTS\Binn*.

W najczęściej spotykanych scenariuszach integracji danych znajdziemy pasujące (odpowiednie do potrzeb) zadanie wbudowane. Możemy jednak zastanawiać się, co się stanie, jeśli zechcemy wykonać bardzo specjalną pracę – na przykład wysłać specjalny komunikat do systemu starszego typu. Zasadniczo, mamy wówczas do dyspozycji dwie możliwości:

1. **Wykorzystać zadanie Script** Standardowy zbiór zadań zawiera szczególne zadanie o nazwie *Script*. Jak wspomnieliśmy w rozdziale 1, zadania skryptowe budowane są w Visual Studio Tools for Application (VSTA). Narzędzie to zapewnia zintegrowane środowisko projektowe (*integrated development environment* – IDE), które pozwala tworzyć własny kod w języku C# lub Visual Basic. Zadanie *Script* daje programiście dostęp do całego modelu obiektowego, w tym takich możliwości jak czytanie lub zapisywanie zmiennych, odczytywanie danych

z menedżerów połączeń czy wyzwalanie zdarzeń. Co więcej, można odwołać się nawet do zewnętrznej asemblacji .NET, o ile została ona zainstalowana w Global Assembly Cache (GAC). W większości przypadków jednak samo zadanie *Script* zapewnia dostateczną elastyczność.

2. **Zaprojektować własne, niestandardowe zadanie** Jeśli nawet zadanie *Script* nie jest w stanie wypełnić naszych potrzeb, możemy napisać własne zadanie, korzystając ze standardowych interfejsów i klas bazowych udostępnianych przez SSIS. W celu utworzenia niestandardowego zadania musimy jedynie zbudować nową klasę spełniającą wymagania interfejsu. Wewnątrz nowej klasy możemy zasadniczo zrobić wszystko, co tylko zechcemy. Po stworzeniu klasy należy wbudować kod nowego zadania w plik biblioteki dynamicznej (DLL) i umieścić ten plik w folderze zadań wspomnianym wcześniej (*%Program Files%\Microsoft SQL Server\110\DTS\tasks*). Przy kolejnym uruchomieniu SQL Server Data Tools narzędzie to automatycznie odnajdzie nowe zadanie podczas skanowania tego folderu. Należy również dołączyć zbudowaną asemblację do GAC, aby motor SSIS mógł załadować ją podczas wykonywania.

Ograniczenia pierwszeństwa

Zadania muszą być połączone ze sobą, aby móc w sposób skoordynowany wykonać pracę. Na przykład w wielu pakietach SSIS można zauważyć typowy wzór, obejmujący zadanie *FTP*, po którym następuje zadanie *Data Flow*. Zadanie *FTP* pobiera pliki danych ze zdalnego serwera, zaś zadanie *Data Flow* przetwarza pobrane pliki i ładuje wynik do bazy danych. Oczywiste jest, że zadanie *Data Flow* nie powinno być uruchamiane, dopóki zadanie *FTP* nie zakończy się sukcesem. Opcjonalnie możemy nawet zechcieć dołączyć zadanie *Send Email*, które prześle powiadomienie w razie niepowodzenia zadania *FTP*.

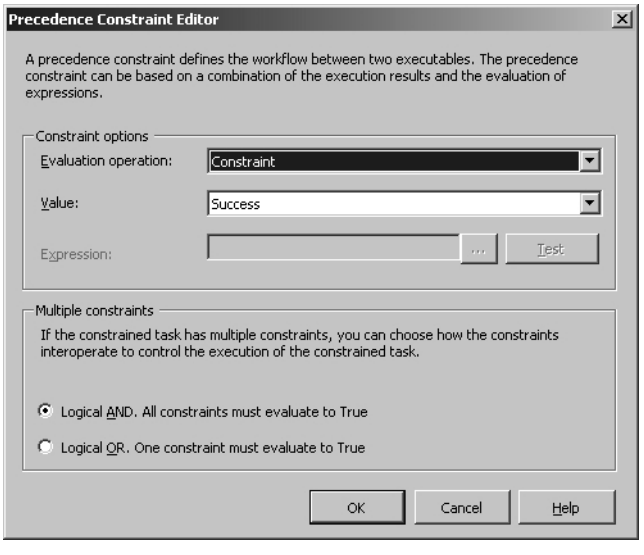
Powiązania i zależności pomiędzy takimi zadaniami odzwierciedlają *logikę integracji danych*. W terminologii SSIS takie uwarunkowanie nosi nazwę *ograniczenie pierwszeństwa*. Reprezentuje ono kryterium lub potwierdzenie, które musi zostać spełnione, aby zadanie mogło zostać uruchomione. Każde zadanie jest powiązane z przynajmniej jednym ograniczeniem pierwszeństwa. W przypadku zadań, dla których nie ma zadań poprzedzających, możemy wyobrazić sobie niewidzialne ograniczenie pierwszeństwa, którego kryterium jest zawsze spełnione.

Najprostszym ograniczeniem pierwszeństwa jest stan ukończenia dla zadania poprzedzającego. Każdemu zadaniu odpowiada pewien stan, który uzyskuje po zakończeniu działania. Stan ten to może być sukces, porażka lub po prostu zakończenie. W interfejsie graficznym projektanta ograniczenie pierwszeństwa prezentowane jest jako linia łącząca zadanie poprzedzające z bieżącym. Kolor tej linii odpowiada wymaganemu stanowi końcowemu. Linia zielona oznacza, że zadanie zostanie uruchomione jedynie wówczas, gdy poprzednie zadanie ukończy się sukcesem, podczas gdy linia

czerwona odpowiada wywołaniu zadania tylko wtedy, gdy poprzednie zadanie się nie powiodło. W przypadku zakończenia (bez względu na to, czy udanego, czy nie) kolor linii będzie niebieski. W celu zmiany rodzaju ograniczenia należy w projektancie kliknąć linię prawym przyciskiem myszy i wybrać polecenie Edit (Edytuj), po czym wybrać inną wartość stanu.

Bardziej złożone ograniczenia pierwszeństwa opierają się na kombinacji stanów ukończenia oraz wyrażeń. Poprzez połączenie tych elementów można sformułować bardziej skomplikowane wymagania. Na przykład można zdefiniować zmienną o nazwie *licznik* i zdefiniować ograniczenie pierwszeństwa jako „Uruchom to zadanie tylko wtedy, gdy *licznik* ma wartość większą od pięciu i poprzednie zadanie ukończyło się sukcesem".

Co więcej, zadanie może mieć więcej niż jedno ograniczenie pierwszeństwa. SSIS pozwala określić, czy ograniczenia takie mają być łączone poprzez logiczną koniunkcję (AND), czy też alternatywę (OR). Poprzez kliknięcie linii ograniczenia pierwszeństwa prawym przyciskiem myszy i wybranie polecenia Edit można określić szczegółową logikę uwarunkowań w narzędziu Precedence Constraint Editor (Edytor ograniczeń pierwszeństwa), pokazanym na rysunku 2-2.



Rysunek 2-2 Precedence Constraint Editor

Oto prosta demonstracja, jak działają ograniczenia pierwszeństwa. Na samym początku wykonywania pakietu motor przepływu sterowania ocenia każde zadanie w celu sprawdzenia, czy jego ograniczenia pierwszeństwa są spełnione. Na początku jedynie te zadania, dla których nie istnieją zadania poprzedzające, będą gotowe do wykonania, gdyż ich ograniczenia pierwszeństwa są zawsze spełnione. Po zakończeniu działania tych wstępnych zadań ich stany ukończenia ulegną zmianie, zatem motor ponownie

dokonuje oceny wszystkich pozostałych zadań i uruchamia te, dla których warunki ograniczenia zostały spełnione. Cykl ten powtarzany jest do chwili, gdy wszystkie zadania zostaną ukończone i nie ma już żadnych zadań gotowych do wykonania.

Zmienne i wyrażenia

W poprzednim podrozdziale zaznaczyliśmy, że zadanie nie jest świadome istnienia innych zadań, co oznacza, że nie może ono zmieniać właściwości ani przekazywać komunikatów do innych zadań. Pojawia się więc pytanie: jak zadania komunikują się ze sobą? Mimo wszystko bowiem zadania często potrzebują przekazania jakiejś informacji pomiędzy sobą, aby móc w ogóle wykonać swoją pracę. Na przykład w celu przetworzenia zbioru pobranych plików zadanie *Data Flow* musi znać nazwy plików pobranych przez poprzedzające zadanie *FTP*.

Inne pytanie brzmi: jak dostosować pakiet, aby nie był wykonywany zawsze w dokładnie taki sam sposób. Na przykład załóżmy, że mamy pakiet do przetwarzania plików danych o określonym formacie. Ponieważ ścieżki do plików danych mogą z czasem się zmieniać, zdecydowanie nie chcielibyśmy na sztywno wpisywać tych ścieżek w poszczególne zadania. Potrzebujemy więc mechanizmu, dzięki któremu moglibyśmy określić ścieżkę (nazwę pliku) przy uruchamianiu pakietu.

Te dwa cele można osiągnąć przy użyciu zmiennych SSIS. *Zmienna* to – zasadniczo – lokalizacja w pamięci. Może jej zostać przypisana jakoś wartość – przed lub w trakcie wykonywania pakietów. Aby przypisać wartości przed wykonaniem, możemy określić argumenty wiersza polecenia dla narzędzia Dtexec.exe. W celu wymiany informacji podczas wykonywania jedno zadanie może zapisać wartość w zmiennej, a inne będzie mogło ją odczytać.

Zmienne są ściśle typizowane, co oznacza (na przykład), że nie możemy zmiennej łańcuchowej (typu *string*) przypisać wartości całkowitoliczbowej. Zmienna ma również wartość początkową oraz zakres widoczności. Na przykład możemy zdefiniować zmienną w kontenerze *Sequence*. W rezultacie wszystkie zadania lub kontenery zawarte w tym kontenerze będą miały dostęp do tej zmiennej. Jednak z zewnątrz (spoza kontenera *Sequence*) zmienna ta będzie niewidoczna.

Niektóre zmienne są wbudowane; nazywamy je zmiennymi systemowymi. Reprezentują one zazwyczaj ogólne fakty, które mogą być potrzebne podczas wykonywania pakietu, takie jak czas uruchomienia pakietu. Informacje te można zapisać w pliku inspekcji podczas każdego wykonywania pakietu. Zmienne systemowe są zazwyczaj wartościami tylko do odczytu.

Możemy łączyć zmienne w wyrażenia zawierające inne zmienne lub stałe. *Wyrażenie* to seria zmiennych, wartości literalnych (stałych) oraz operatorów, które mogą zostać ostatecznie wyliczone do pojedynczej wartości. SSIS wykorzystuje swój własny język wyrażeń, udostępniający liczne operatory i funkcje.

Zależność pomiędzy zmiennymi i wyrażeniami może stać się bardzo złożona. Wyrażenie jest zwykle zbudowane na podstawie kilku zmiennych – ale zmienna

również może być utworzona jako wynik wyrażenia! Gdy zaznaczymy zmienną w interfejsie projektanta, zauważymy właściwość o nazwie *Evaluate As Expression* (wylicz jako wyrażenie). Właściwość ta określa, czy zawartość danej zmiennej ma być traktowana jako wyrażenie, czy jakoś wartość literalna.

Uwaga Należy zachować ostrożność przy definiowaniu zmiennych na podstawie innych zmiennych lub wyrażeń, gdyż może to niekiedy prowadzić do powstania odwołania cyklicznego, co oznacza, że zmienna ostatecznie odwołuje się do siebie samej poprzez jedną lub kilka zmiennych pośredniczących. Odwołania cykliczne nie tylko prowadzą do niejednoznaczności i błędów, ale również są bardzo trudne w debugowaniu.

Zadania odwołują się do zmiennych lub wyrażeń poprzez ich właściwości. Na przykład w zadaniu *FTP* potrzebujemy określić nazwę pliku do pobrania. Zamiast wpisywania nazwy na sztywno w kodzie zadania, możemy zdefiniować zmienną i odwoływać się do niej poprzez właściwość. Możemy również przypisać do właściwości wyrażenie zamiast pojedynczej zmiennej. Gdy wyrażenie zostanie wyliczone podczas wykonywania pakietu, uzyskana wartość zostanie przypisana tej właściwości.

Trzeba jednak pamiętać, że nie wszystkie właściwości mogą akceptować zmienne lub wyrażenia. Niektóre właściwości mogą mieć tylko wartości literalne. Właściwości, którym można przypisać zmienną lub wyrażenie, nazywane są właściwościami obliczalnymi. W większości edytorów zadań znajdziemy stronę o nazwie Expressions (Wyrażenia), zawierającą listę wszystkich właściwości obliczalnych dla danego zadania.

Kontenery

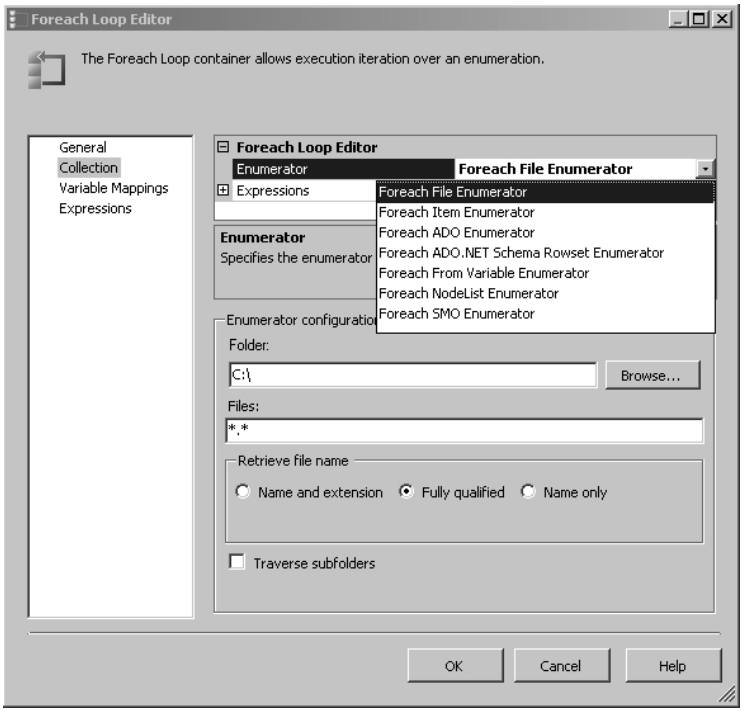
Jak wspomnieliśmy wcześniej, zadania stanowią najmniejsze bloki konstrukcyjne pakietu SSIS. Zadanie jest więc podobne do pojedynczego wyrażenia w języku programowania. Osoby mające jakieś doświadczenie w programowaniu wiedzą, że wyrażenia można porządkować w funkcje lub metody, dzięki czemu można programowi nadać przejrzystą strukturę. W SSIS zarządzamy zadaniami w podobny sposób, grupując je w różne kontenery, dzięki czemu można łatwiej kontrolować ich działanie i wzajemne zależności.

SSIS udostępnia trzy różne typy kontenerów: kontener *For Loop* (pętla), *Foreach Loop* (pętla dla każdego) oraz *Sequence* (sekwencja).

Dwa pierwsze kontenery umożliwiają wykonywanie powtarzalnych kroków pracy. Na przykład chcielibyśmy przetworzyć wszystkie pliki danych zawarte we wskazanym folderze, używając tej samej logiki. Kontener *Foreach Loop* pozwoli to zrealizować. Najpierw musimy zdefiniować zadanie *Data Flow*, które wykona faktyczne przetwarzanie. Następnie umieszczamy to zadanie w kontenerze *Foreach Loop* i konfigurujemy numerator plikowy jako kontekst pętli. Każda iteracja pętli zamapuje kolejną nazwę pliku do zmiennej, zaś zadanie *Data Flow* odczyta nazwę pliku do przetworzenia z tej zmiennej.

Kontener Foreach Loop

Na rysunku 2-3 widzimy, że kontener *Foreach Loop* umożliwia wykonywanie zadań w pętli dla różnych typów obiektów, takich jak pliki lub rekordy ADO.NET. Każdy obsługiwany rodzaj obiektu nazywany jest numeratorem pętli i dla każdego istnieją określone właściwości, które musimy określić. Na przykład dla numeratora plikowego musimy określić lokalizację folderu oraz filtr nazwy pliku. Dla każdego typu numeratora możemy zamapować różne wyliczone elementy do zmiennych poprzez stronę Variable Mappings, co pozwala odwoływać się do nich w zadaniach zawartych w kontenerze. Rozdział 7, „Połączenia SSIS", zawiera przykład wykorzystania numeratora plikowego Foreach, który może okazać się bardzo przydatny.

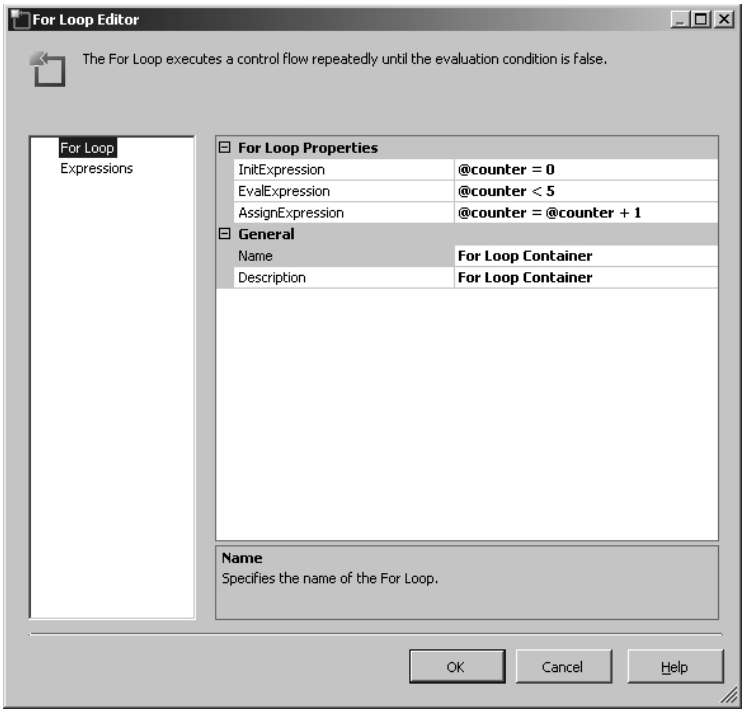


Rysunek 2-3 Numeratory Foreach Loop

Numeratory *Foreach Loop* są również rozszerzalne. Oznacza to, że możemy napisać swój własny numerator dla specjalnego stosowania. Na przykład, jeśli chcemy wykonać pętlę wobec pewnych szczególnych typów obiektów ze starszego systemu ERP (Enterprise Resource Planning – planowanie zasobów przedsiębiorstwa), konieczne będzie zbudowanie niestandardowego numeratora *Foreach Loop*. Rysunek 2-3 pokazuje dostępne (wbudowane) numeratory *Foreach Loop*.

Kontener For Loop

Kontener *For Loop* działa podobnie do kontenera *Foreach Loop*, ale jest prostszy. Przebieg pętli jest tu kontrolowany przez wyrażenie. Kontener kontynuuje wykonywanie zawartych w nim zadań, aż wyliczenie warunku pętli da logiczny fałsz. Zazwyczaj w wyrażeniu warunku odwołamy się do pewnej liczby zmiennych. Na przykład możemy zdefiniować zmienną przechowującą pozostałą liczbę wiadomości email, które mają zostać wysłane. Pętla wykonywana jest, dopóki liczba ta nie osiągnie zera (jest zmniejszana o jeden przy każdej iteracji). Rysunek 2-4 pokazuje narzędzie For Loop Editor.



Rysunek 2-4 For Loop Editor

Kontener Sequence

W porównaniu z dwoma wcześniejszymi, kontener *Sequence* wydaje się mniej atrakcyjny na pierwszy rzut oka. Powoduje on po prostu zgrupowanie istniejących zadań. Kontener ten nie zapewnia żadnej nowej funkcjonalności, co prowadzi do częstych pytań, dlaczego w ogóle taki kontener istnieje. Główną przyczyną jest to, że stosowanie tego kontenera upraszcza zarządzanie kodem. Na przykład trzy kolejne zadania mogą współdzielić takie same właściwości, takie jak ustawienia transakcji. Poprzez zgrupowanie tych zadań w kontener *Sequence* możemy ustawić wspólną właściwość dla kontenera, a nie dla poszczególnych zadań. Z punktu widzenia zarządzania oczywiste jest,

że lepiej ustawiać jedną właściwość wspólną, niż wiele właściwości indywidualnych zadań. Dodatkowo przy użyciu kontenera *Sequence* możemy włączać lub wyłączać zawarte w nim zadania jako całość. Wreszcie, o czym już wspomnieliśmy wcześniej, kontener ten pozwala zawęzić zakres widoczności dla zmiennych.

Warto wspomnieć tu o dwóch interesujących faktach na temat kontenerów SSIS. Pierwszym jest to, że cały pakiet jest w istocie kontenerem *Sequence*. Oczywiście, pakiet ma pewne dodatkowe właściwości jako obiekt najwyższego poziomu. Jednak pomijając to, pakiet jest zasadniczo po prostu kontenerem *Sequence*. Druga interesująca rzecz to to, że każde zadanie jest opakowane ukrytym kontenerem o nazwie *Task Host*. Kontener ten jest zaprojektowany do celów wewnętrznych programu. Każde zadanie musi bowiem obsłużyć typowe wymagania, takie jak dostęp do zmiennych, menedżerów połączeń, powiązane programy obsługi zdarzeń i tak dalej. Kontener *Task Host* obsługuje tego typu wspólną funkcjonalność, dzięki czemu zadania mogą koncentrować się tylko na ich własnej logice.

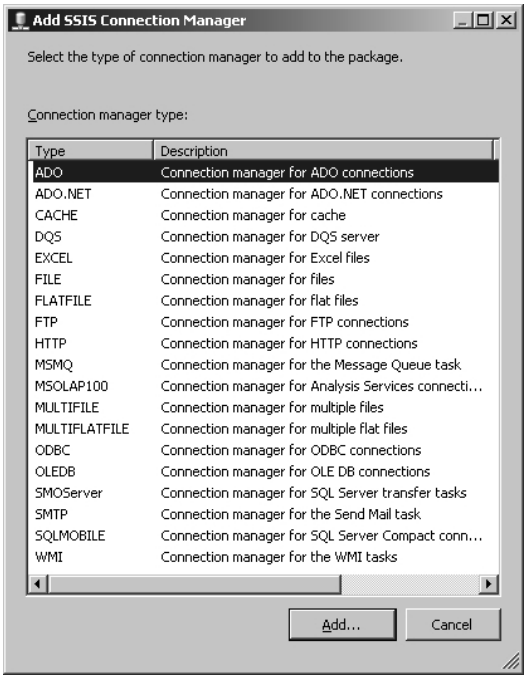
Kontenery mogą być zagnieżdżane. Gdy zadania zostaną umieszczone w kontenerze, on sam staje się nową jednostką pracy. Można następnie umieścić taki kontener w większym kontenerze, zawierającym inne zadania lub kontenery. Finalnie struktura pakietu staje się drzewem, w którym każdy liść jest albo zadaniem, albo pustym kontenerem.

Menedżery połączeń

Budowanie aplikacji integrowania danych niemal zawsze wymaga dostępu do najrozmaitszych zewnętrznych źródeł danych, takich jak standardowe relacyjne bazy danych (na przykład Microsoft SQL Server), płaskie pliki lub systemy starszego typu, takie jak ERP. Z wcześniejszych rozważań wiemy już, że całą pracę w SSIS realizują zadania. Aby wykonać tę konkretną pracę, zadanie musi połączyć się z odpowiednimi źródłami danych. Na przykład zadanie FTP wymaga połączenia ze wskazanym serwerem FTP.

W SSIS tego typu połączenia danych nie są implementowane przez same zadania. Zamiast tego są one kapsułkowane jako samodzielne obiekty zwane menedżerami połączeń (*connection managers*). W ten sposób różne zadania mogą dzielić wspólne połączenia danych bez duplikowania kodu. Wszystkie połączenia są udostępniane przez odpowiadające im menedżery połączeń. Każdy menedżer połączenia tworzy tylko swój własny typ połączenia. SSIS jest standardowo wyposażony w wielką liczbę menedżerów połączeń (rysunek 2-5).

W celu skonfigurowania nowego menedżera połączeń należy przypisać mu właściwości połączenia, takie jak adres, nazwa użytkownika lub hasło. Następnie menedżera połączenia przypisujemy do odpowiedniego zadania. W trakcie wykonywania zadanie zażąda utworzenia instancji połączenia od przypisanego mu menedżera połączeń, po czym wykona swoją pracę na uzyskanym połączeniu.



Rysunek 2-5 Wbudowane menedżery połączeń

Niektóre menedżery połączeń wspierają ważną właściwość o nazwie *Retain Same Connection* (Zachowaj to samo połączenie). Jeśli właściwość ta ma wartość *true*, menedżer połączeń zawsze zwróci tę samą instancję połączenia, nawet jeśli połączenia zażąda wiele zadań. Dla przykładu załóżmy, że mamy trzy zadania FTP w pakiecie i że wszystkie one używają tego samego menedżera połączenia FTP. Jeśli opcja jest wyłączona (*false*) i trzy zadania zażądają połączenia FTP, utworzone zostaną trzy instancje połączeń. Jeśli jednak włączymy tę opcję (*true*), utworzone zostanie tylko jedno połączenie i wszystkie trzy zadania będą je wykorzystywać wspólnie. W niektórych sytuacjach postępowanie takie może zwiększyć wydajność i zapewnić oszczędności na zasobach połączeń.

Innym interesującym faktem na temat menedżerów połączeń jest to, że nie muszą one koniecznie zwracać „prawdziwego połączenia"; zwrotem może być obiekt dowolnego typu, zależnie od implementacji menedżera. Na przykład menedżer połączenia Flat File w rzeczywistości zwraca ścieżkę dostępu do pliku, a nie uchwyt otwartego pliku.

Jak można oczekiwać, menedżery połączeń również są rozszerzalne w SSIS i zaprojektowanie niestandardowego menedżera jest bardzo proste. SSIS udostępnia standardowe interfejsy i klasy bazowe. Musimy jedynie napisać własną klasę implementacji opartą na tych interfejsach lub klasach bazowych, wbudować ją w DLL, po czym umieścić uzyskany plik DLL w folderze *%Program Files%\Microsoft SQL Server\110*

DTS\Connections. Przy kolejnym uruchomieniu SQL Server Data Tools niestandardowy menedżer połączeń pojawi się na liście.

Ważną zmianą dokonaną w wersji SQL Server 2012 jest to, że menedżery połączeń mogą być współdzielone przez wiele pakietów. Pozwala to oszczędzić wiele wysiłku projektantów i znacząco zwiększa produktywność. Większość rozwiązań ETL zawiera wiele pakietów, przy czym pakiety te zazwyczaj pracują względem tych samych baz danych lub hurtowni danych. We wcześniejszych wersjach SSIS konieczne było definiowanie menedżerów połączeń dla każdego pakietu, ale teraz musimy zdefiniować tylko jednego menedżera danego typu i wykorzystać go we wszystkich pakietach. Jeśli nastąpi jakaś zmiana, na przykład hasła, trzeba będzie wprowadzić ją w tylko jednym miejscu.

Pakiety i projekty

Pakiet jest obiektem SSIS, który zawiera wszystkie inne typy obiektów, takie jak zadania, kontenery lub zmienne. Po utworzeniu nowego rozwiązania Integration Services w narzędziu możemy zobaczyć, że został w nim utworzony nowy pakiet. Jest to obiekt najwyższego poziomu w wersjach SSIS wcześniejszych niż SQL Server 2012 – cała nasza praca definiowana jest wewnątrz pakietu. Jest to również podstawowa jednostka projektowa i wykonawcza.

Jednak w wersji SQL Server 2012 pojawiła się nowa koncepcja, zwana *projektem,* która zasadniczo jest kontenerem dla pakietów i innych współużytkowanych elementów. Tym samym pakiet nie jest już obiektem najwyższego poziomu. Termin *projekt* może być nieco mylący, gdyż już od dawna wykorzystywany jest w Visual Studio. Na przykład mówimy „Utwórz nowy projekt SSIS w Visual Studio". W tym wypadku „projekt", który mamy na myśli, nie będzie projektem w rozumieniu rozwiązań Visual Studio, ale nowym bytem wprowadzonym w SSIS 2012.

Koncepcja projektu zapewnia większy zakres działania, pozwalając na odwoływanie się do różnych pakietów SSIS wchodzących w skład projektu i współdzielenie informacji pomiędzy pakietami. Na przykład typowe rozwiązanie ETL hurtowni danych zawiera zwykle wiele pakietów. Jeden z nich jest głównym pakietem sterującym, zaś pozostałe pakietami podrzędnymi. Główny pakiet sterujący wywołuje pakiety podrzędne, realizujące całą pracę ETL, w tym aktualizowanie tabel wymiarów, tabel faktów, zapisywanie dzienników i tak dalej. Zazwyczaj główny pakiet będzie zawierał kilka zadań *Execute Package*, wywołujących te pakiety podrzędne. Aby móc wskazać pakiety podrzędne, konieczne jest określenie absolutnych ścieżek do plików pakietów. Przy takim podejściu, jeśli konieczne będzie przeniesienie pakietów do innego folderu, powiązania zostaną zerwane, gdyż zmienią się lokalizacje pakietów podrzędnych. Przy tak wielu wzajemnych zależnościach trudne może być wykrywanie i naprawianie błędów. Dla kontrastu, poprzez wprowadzenie pojęcia projektu zawierającego pakiety musimy tylko określić nazwę pakietu podrzędnego zawartego wewnątrz projektu, bez konieczności zajmowania się bezwzględną ścieżką do jego pliku.

Co więcej, koncepcja projektu ułatwia współdzielenie informacji pomiędzy wieloma pakietami. W SSIS 2012 można wspólnie wykorzystywać menedżery połączeń i parametry ogólne projektu we wszystkich pakietach, co jest bardzo użyteczne, gdyż ponownie, w razie konieczności dokonania zmian, trzeba je wykonać tylko w jednym miejscu, a nie we wszystkich zaangażowanych pakietach.

Ta nowa koncepcja upraszcza również pracę programisty. Po wykonaniu polecenia *build* w SSDT cała zawartość projektu SSIS jest umieszczana w pojedynczym pliku o rozszerzeniu .ispac. Zbudowane rozwiązanie obejmuje pakiety, wspólne menedżery połączeń i parametry projektu, co oznacza, że całą pracę wdrożeniową można wykonać posługując się tylko jednym plikiem.

Wskazówka Plik projektu jest w rzeczywistości standardowym plikiem zip. Jeśli zmienimy rozszerzenie nazwy pliku z .ispac na .zip, można go łatwo otworzyć i obejrzeć całą wewnętrzną zawartość.

Parametry

Kolejną nową koncepcją wprowadzoną w wersji SQL Server 2012 jest *parametr*. Istnieją dwa typy parametrów: *parametry pakietu* oraz *parametry projektu*. Wszystkie parametry pakietu można obejrzeć na zakładce Parameters w widoku projektu pakietu. Aby zobaczyć parametry projektu, należy podwójnie kliknąć węzeł Project.params w narzędziu Solution Explorer.

Parametr pakietu można porównać do kontraktu pomiędzy pakietem a elementem wywołującym, na przykład pakietem nadrzędnym. Pakiet podrzędny może wymagać pewnych informacji do wykonania swojej pracy, na przykład nazwy tabeli wymiarów, którą powinien zaktualizować. We wcześniejszych wersjach SSIS projektanci zwykle przekazywali takie informacje poprzez pewnego typu konfigurację zewnętrzną, jednak nie jest zbyt wydajne, gdyż jest to niejawna droga przesyłania informacji. Po pierwsze, czasem trudno jest dokładnie ustalić, jakie informacje konfiguracyjne są niezbędne do wykonania pakietu podrzędnego. Po drugie, może to łatwo prowadzić do błędów, jeśli coś zostanie wykonane nieprawidłowo w konfiguracji zewnętrznej. Parametry pakietów są, przeciwnie, jawną metodą przesyłania informacji: można łatwo sprawdzić, jakie informacje są potrzebne, gdy otworzymy pakiet. Łatwiejsze jest również weryfikowanie przez SSIS, czy przekazanym parametrom zostały przypisane właściwe wartości podczas wykonywania.

Parametr projektu jest podobny do parametru pakietu – po prostu operuje na większym zakresie: opisuje wymagania potrzebne do uruchomienia projektu. Kolejną korzyścią jest to, że parametry projektu mogą być współużytkowane przez wszystkie pakiety.

Zazwyczaj parametry są ustawiane poprzez argumenty wiersza polecenia lub procedury składowane T-SQL, gdy projekt jest wykonywany z wnętrza katalogu SSIS (bazy

SSISDB). Także zmienne mogą być ustawiane poprzez argumenty wiersza polecenia. Wiele osób myli ze sobą obydwa terminy: *parametry* oraz *zmienne*, a sytuacja może stać się jeszcze mniej jasna, jeśli zdefiniujemy zmienne lub wyrażenia oparte na wartościach parametrów. Oto różnice:

1. Wartości zmiennych mogą być zmieniane w trakcie wykonywania pakietu, natomiast wartości parametrów pozostają stałe (tylko do odczytu).
2. Z punktu widzenia wykorzystania parametry są zwykle używane do zdefiniowania kontraktu pomiędzy pakietem i elementem wywołującym, podczas gdy zmienne służą przede wszystkim do wymiany informacji wewnątrz pakietu.

Dostawcy dzienników

Koncepcja dostawcy dziennika (*log provider*) jest stosunkowo prosta. Jak sugeruje nazwa, dostawca dziennika to dostawca magazynu, w którym można zapisywać dzienniki.

Wykonywanie pakietu SSIS może tworzyć bogaty zbiór dzienników (zależnie od konfiguracji rejestrowania). Dzienniki te są ważne, gdy zachodzi potrzeba inspekcji lub rozwiązywania problemów. Dzienniki te możemy zapisywać w rozmaitych miejscach, takich jak pliki dyskowe, bazy danych lub dzienniki zdarzeń systemu Windows. Dostawcy dzienników reprezentują te miejsca docelowe. Dostawca dziennika jest więc podobny do menedżera połączeń, ale zawiera dodatkowo pewne szczególne informacje na temat formatu rejestrowania.

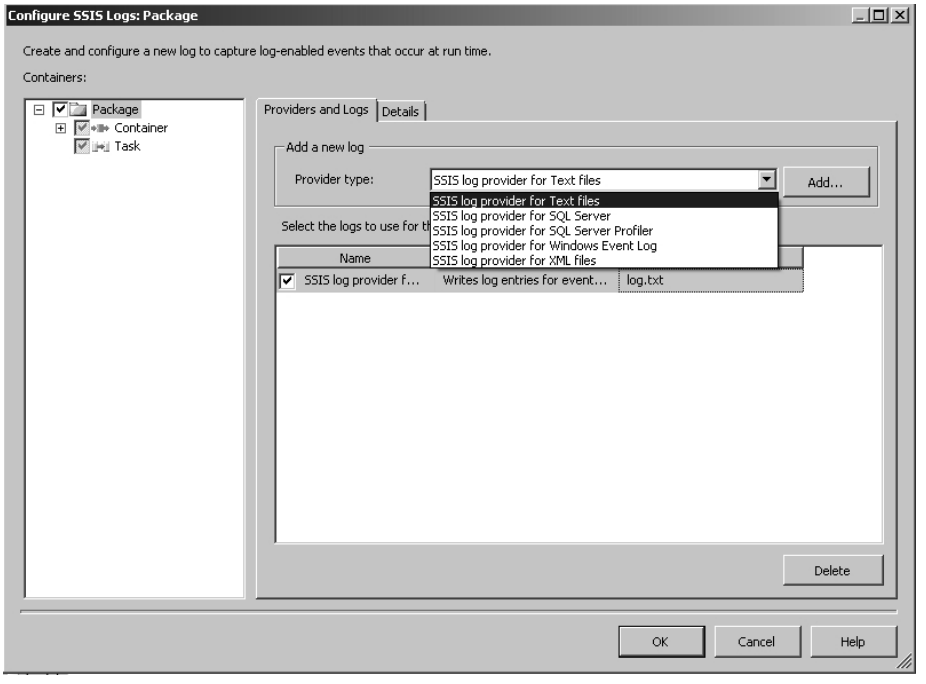
Wszyscy dostępni dostawcy dzienników w pakiecie są definiowani na poziomie pakietów, co oznacza, że nie możemy zdefiniować dostawcy dziennika obsługującego tylko wybrany kontener. Po zdefiniowaniu dostawcy dziennika wszystkie kontenery i zadania mają do niego dostęp. Jeśli dla kontenera wybranych zostanie wiele dostawców dzienników, dzienniki tworzone w tym kontenerze będą zapisywane we wszystkich miejscach docelowych niezależnie.

W celu wyświetlenia wszystkich dostawców dziennika zdefiniowanych w pakiecie SSIS należy wybrać opcję menu SSIS | Logging w projektancie pakietów. Rysunek 2-6 ukazuje wszystkie wbudowane typy dostawców dzienników.

Można postawić pytanie, jak dostawcy dzienników działają, jeśli całość procesu obejmuje pakiety nadrzędne i podrzędne. SSIS adaptuje w tym przypadku doskonałe rozwiązanie, w którym wszyscy dostawcy dzienników są opakowywani jako pojedynczy dostawca dla pakietów podrzędnych. Jest to całkowicie przezroczyste dla pakietu podrzędnego. Nawet jeśli nie zdefiniujemy w pakiecie podrzędnym żadnego dostawcy dziennika, dzienniki nadal będą mogły być zapisywane poprzez dostawców zdefiniowanych w pakiecie nadrzędnym.

Najpopularniejsze miejsca docelowe dzienników są obsługiwane przez wbudowane w SSIS typy dostawców, takie jak *Text File* lub *SQL Server*. Można również

zaprojektować własnych (niestandardowych) dostawców dzienników, jeśli chcemy zapisywać dzienniki na nietypowych nośnikach.



Rysunek 2-6 Wszystkie wbudowane typy dostawców dzienników

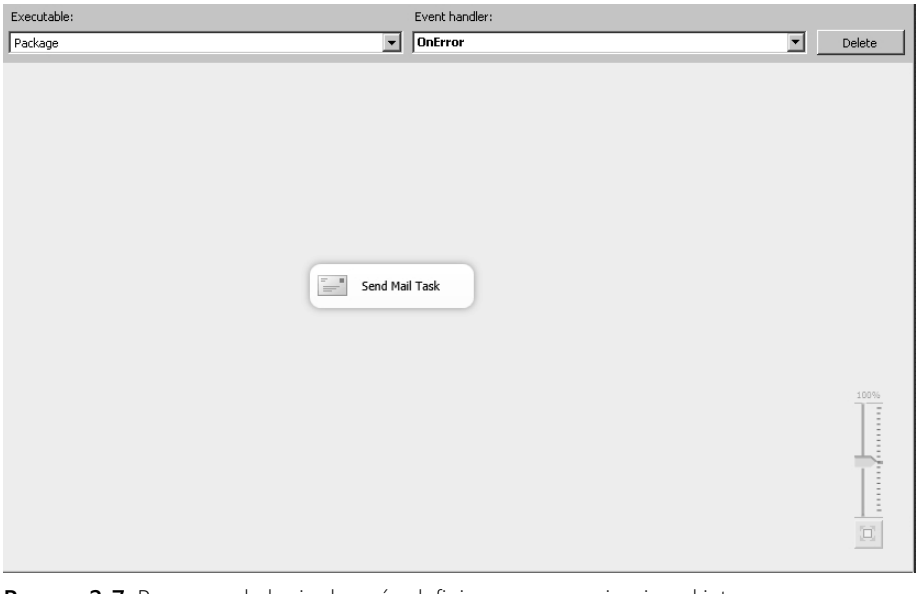
Obsługa zdarzeń

Programy obsługi zdarzeń (*event handler*) stanowią mechanizm pozwalający obsłużyć zdarzenia występujące podczas wykonywania pakietów. Zapewnia to sposoby na radzenie sobie ze szczególnymi sytuacjami, które wykraczają poza normalny przebieg wykonania. Liczne obiekty SSIS, takie jak zadania, kontenery czy menedżery połączeń, mogą wyzwalać zdarzenia podczas wykonywania. Dostępne typy zdarzeń zależne są od typu obiektu. Na przykład dla zmiennych istnieje typ zdarzenia o nazwie *OnVariableValueChanged* (Przy zmianie wartości). Najczęściej stosowane typy zdarzeń to *OnError* (Przy błędzie), *OnWarning* (Przy ostrzeżeniu) oraz *OnTaskFailed* (Przy niepowodzeniu zadania).

Program obsługi zdarzeń jest podobny do samodzielnego pakietu, który ma dostęp do niektórych specjalnych zmiennych systemowych, takich jak *ErrorCode* (Kod błędu), *ErrorDescription* (Opis błędu), *EventHandlerStartTime* (Czas rozpoczęcia obsługi) i tak dalej. Te zmienne systemowe pozwalają nam uzyskać więcej informacji kontekstowych, gdy program obsługi zostanie wywołany.

Obsługę zdarzeń można definiować na różnych poziomach, takich jak zadanie, kontener lub cały pakiet. Gdy wystąpi zdarzenie, motor przepływu sterowania najpierw sprawdza, czy na bieżącym poziomie obiektu został zdefiniowany odpowiedni program obsługi zdarzenia. Jeśli tak, wywoła program obsługi i prześle zdarzenia do zakresu nadrzędnego. Na tym poziomie motor przepływu sterowania powtarza swoją pracę i kontynuuje przekazywanie zdarzenia. Proces trwa do momentu osiągnięcia poziomu pakietu lub poziomu, na którym został zdefiniowany program obsługi zdarzenia zawierający deklarację, że zdarzenie nie powinno być dalej propagowane.

Narzędzie projektowania pakietów zawiera zakładkę Event Handlers, na której można dotrzeć do wszystkich poziomów kontenerów i powiązanych z nimi programów obsługi zdarzeń. Rysunek 2-7 pokazuje obsługę zdarzenia *OnError* zdefiniowaną na poziomie pakietu. Wewnętrznie program obsługi wysyła wiadomość email do administratora.



Rysunek 2-7 Program obsługi zdarzeń zdefiniowany na poziomie pakietu

Przepływ danych

Integration Services wyposażone są w bogactwo wbudowanych zadań, z których należy wyróżnić specjalne zadanie o nazwie *Data Flow* (Przepływ danych). Zadanie to zapewnia pełną funkcjonalność potrzebną dla wydobywania, transformowania i ładowania danych. Zadanie *Data Flow* jest kluczowym elementem każdej aplikacji integrowania danych. W większości przypadków inne zadania wykonują tylko pracę pomocniczą, taką jak pobieranie plików lub inicjowanie baz danych.

Z perspektywy przepływu sterowania zadanie *Data Flow* jest normalnym zadaniem z takimi samymi interfejsami jak każde inne, jednak złożoność tego zadania jest niemal taka sama, jak całego przepływu sterowania. Każde zadanie *Data Flow* hostuje instancję przepływu danych. Wewnątrz zadania znajduje się motor przepływu danych, który określa plan i harmonogram wykonania tej instancji przepływu danych.

Instancja przepływu danych składa się z adapterów źródłowych, komponentów transformacyjnych oraz adapterów docelowych. Źródłowe i docelowe adaptery wyznaczają granice przepływu danych SSIS, podczas gdy komponenty transformacyjne pomiędzy nimi wykonują całą pracę konwertowania danych. Wszystkie dane przychodzą z adaptera źródłowego, przechodzą przez komponenty transformacyjne (niekoniecznie przez wszystkie) i ostatecznie trafiają do adaptera docelowego.

Adaptery źródłowe

Adapter źródłowy wyciąga dane z zewnętrznych źródeł, konwertuje je na format przepływu SSIS, po czym wypycha je do kolejnych komponentów przepływu danych. Po osiągnięciu końca strumienia danych wysyła specjalny wiersz danych o nazwie *end of row set* (koniec zbioru wierszy), dzięki czemu motor przepływu danych wie, że wszystkie dane zostały pobrane. Logika adaptera źródłowego jest bardzo prosta. W większości przypadków kod głównego wątku to po prostu pętla, która jest wykonywana do natrafienia na ostatni wiersz. Adapter źródłowy ma zero wejść i *n* wyjść.

Gdy mowa o konwertowaniu danych do formatów przepływu danych SSIS, mamy na myśli dwie podstawowe konwersje, które adapter musi wykonać. Po pierwsze, musi zmienić postać danych na tabelaryczną. Następnie musi zamienić zewnętrzny system typów danych na wewnętrzny (specyficzny dla SSIS) system typów.

Dane zewnętrzne mogą mieć dowolną postać. Może być to już format tabelaryczny, jak w przypadku danych odczytywanych z systemu bazodanowego, jak Microsoft SQL Server. Może być to również postać hierarchiczna, jak w przypadku danych w pliku XML. W każdym przypadku adapter źródłowy musi zamienić to na format tabelaryczny, będący jedynym formatem akceptowanym przez motor przepływu danych SSIS.

Wszystkie dane wypychane do przepływu danych SSIS muszą być zgodne z systemem typów danych SSIS, który składa się z ustalonego zbioru typów, takich jak *DT_BOOL*, *DT_I4* czy *DT_WSTR*. Jakkolwiek SSIS natywnie zawiera większość zwykle stosowanych typów danych, w zewnętrznych źródłach mogą istnieć pewne specjalne typy danych. Na przykład starsze systemy ERP często wykorzystują specjalną reprezentację dla daty i czasu. W takich sytuacjach adapter źródłowy musi przekonwertować te specjalne zapisy na równoważne typy danych SSIS.

SSIS jest wyposażony w wiele standardowych adapterów źródłowych, takich jak adapter OLE DB lub ADO.NET. Możliwe jest również zaprojektowanie własnych adapterów, jeśli żaden z istniejących nie spełnia naszych potrzeb.

Adaptery docelowe

Adapter docelowy działa analogicznie do źródłowego, ale przepływ danych odbywa się w przeciwnym kierunku. Adapter docelowy kontynuuje odbieranie danych z wcześniejszych komponentów do momentu otrzymania *end of row set*. Dla każdego wiersza danych komponent docelowy konwertuje wartości na format wymagany przez miejsce docelowe danych, po czym ładuje dany wiersz do miejsca docelowego poprzez specyficzne dla połączenia interfejsy API. Adapter docelowy zawiera *n* wejść i 1 wyjście błędu.

SSIS jest wyposażony w większą liczbę adapterów docelowych niż źródłowych. Oprócz popularnych adapterów docelowych, takich jak OLE DB lub ADO.NET, udostępnia także kilka specjalnych adapterów docelowych powiązanych z Online Analysis Processing (OLAP). Do adapterów tych należą Data Mining Model Training, Dimension Processing oraz Partition Processing. Wszystkie te adaptery wymagają połączenia z SQL Server Analysis Services. Adaptery te pozwalają na wykonywanie większości typowych zadań hurtowni danych wewnątrz jednego pakietu, bez konieczności wywoływania dodatkowych aplikacji.

Uwaga Niektóre komponenty przepływu danych są dostępne tylko w wybranych wydaniach SQL Server. Na przykład te adaptery docelowe, które dotyczą SQL Server Analysis Services, nie są dostępne w wydaniu SQL Standard ani niższych wersjach.

Rysunek 2-8 ukazuje wszystkie wbudowane adaptery źródłowe i docelowe.

Rysunek 2-8 Źródłowe i docelowe adaptery