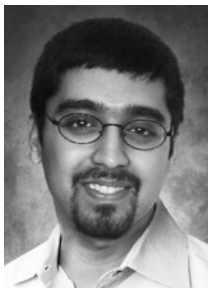
Minimize data warehouse disruption post mergers and acquisitions with SAP NetWeaver 7.0's Universal Data Connect and Master Data Management

by Prakash Darji and Scott Cairncross



Prakash Darji
Product Manager,
Corporate Performance Management,
SAP Labs, LLC

Scott Cairncross
Platinum Consultant,
Corporate Performance Management RIG,
SAP Labs, LLC

(full bios appear on page 92)

Mergers and acquisitions are common events in the business world. As they occur, they typically cause disruptions that ripple through both acquiring and target companies. One of the biggest disruptions takes place within the enterprise data warehouse. Once a company is acquired, two data warehouses (one from the acquirer and the other from the acquiree) with disparate data exist. Traditional business data-warehousing functions allow you to observe and analyze your company's information, but they don't provide a way to explore and evaluate data from the acquired company — data that has yet to be integrated with the acquirer's existing data.

You need a way to bring together heterogeneous data from dissimilar data warehouses and IT groups. The business must determine how to organize that heterogeneous data. Its primary concern is minimizing disruption while continuing to get accurate data into its reporting — including the data from the acquired company. IT typically is concerned with the long-term implications of both systems' architectures and with ensuring that they come together in the soundest way possible. These scenarios become easier to handle using the universal data integration (UDI) capabilities provided by SAP NetWeaver 7.0 (formerly 2004s) through its Universal Data (UD) Connect interface and SAP NetWeaver Master Data Management (MDM). Keeping these priorities in mind, let's look at a scenario that shows how to achieve both goals.

The following example is a common situation that many companies encounter. Let's say that a very large consumer product goods (CPG) company, named CompanyA, buys a mid-sized CPG company, named CompanyB. CompanyA uses SAP NetWeaver 7.0, and CompanyB uses a third-party system, in this case, Microsoft Analysis Services.[1] This scenario

[1] For details on Microsoft Analysis Services, go to www.microsoft.com/sql/technologies/analysis/default.mspx.

is common because many large companies currently use SAP solutions and smaller ones tend to use Microsoft solutions. Because these two companies have their own data-warehousing solutions with disparate data models and different master data, it may take a while to unify and integrate CompanyB's information with CompanyA's data. By "unify," we mean that CompanyB may have a product number "1000" that refers to facial tissue, while CompanyA's product number "1000" refers to water bottles. To ensure that you can consistently report this information and maintain the difference between water bottles and facial tissue, you need to unify your data (i.e., make the product numbers different for water bottles and facial tissue).

The business users in CompanyA, perhaps the Operations and Planning Department, will want to analyze the data in both systems immediately with a consistent user interface (UI) and set of tools. They also will want to retain the distinction between facial tissue and water bottles. Most likely, CompanyA's business users will apply the SAP NetWeaver BI reporting tools they are accustomed to, as opposed to the unfamiliar Microsoft Analysis Services.

We've been through this scenario several times and would like to share some of the lessons we've learned. The goal of this article is to explain how to handle mergers and acquisitions within your data warehouse. It targets users who have a basic understanding of the capabilities of SAP NetWeaver 7.0, from data-loading and data-reporting perspectives. Therefore, this article will cover some of the functionality of the BEx Web Analyzer as it relates to this scenario between CompanyA and CompanyB, but assumes the reader has some basic knowledge of the BI Reporting tools. It will help to have an understanding of the new data staging concepts within SAP NetWeaver 7.0 as well. If you are planning to work with data from numerous systems (SAP and non-SAP), then this article will show you how to implement the scenario effectively. Finally, we will cover how the SAP Corporate Performance Management (CPM) Suite can help you go forward to effectively monitor and add insight into this scenario between CompanyA and CompanyB.

We'll begin with an overview of where UD Connect fits into the SAP NetWeaver architecture, and then we'll discuss how to configure UD Connect. After that, we'll talk about the mergers and acquisitions scenario mentioned in the beginning of this article and show you how to implement UD Connect using a four-phased approach that we developed to help you to integrate data from multiple systems while causing the least amount of disruption to the business. Finally, we'll explain what you need to do to monitor your data after the merger or acquisition.

Keep in mind that although UD Connect has been available since the release of SAP NetWeaver 2004, this article focuses on the features offered in SAP NetWeaver 7.0. The functionality enhancements to UD Connect that come with SAP NetWeaver 7.0 also include the new DataSource concept for staging your data.[2]

The new DataSource concept allows you to use both transient and persistent staging (see the sidebar on the following page for information on staging). With the exception of the new DataSource concept, the capabilities of SAP NetWeaver 2004 and SAP NetWeaver 7.0 in this type of situation are identical.

What is UD Connect?

Very simply, UD Connect is the functionality within SAP NetWeaver 7.0 that allows you to work with third-party systems data within your SAP NetWeaver Business Intelligence (BI) data warehouse. There are typically two options for getting third-party data into BI: flat files or UD Connect. Because UD Connect offers very tight integration and metadata discovery from your sources, it is often easier to use than flat files. That's why we'll use UD Connect for the mergers and acquisitions scenario.

[2] For details about the new DataSource concept, go to the SAP NetWeaver 7.0 (2004s) documentation on the SAP Help Portal at www.help.sap.com.

Transient vs. persistent staging

It's important to know the difference between transient and persistent staging. For example, consider our scenario example in which CompanyA uses SAP NetWeaver 7.0 and CompanyB uses Microsoft Analysis Services:

- **Persistent staging:** Configure UD Connect, move the data from Microsoft Analysis Services to SAP NetWeaver, and report on it directly from your SAP NetWeaver system.
- **Transient staging:** Configure UD Connect, leave the data in the Microsoft Analysis Services system, and allow reports to access this data virtually.

UD Connect also permits flexibility by allowing both transient and persistent data storage scenarios. Transiently, you can use direct integration of external data into SAP NetWeaver BI queries through UD Connect.

Note!

Universal data integration (UDI) is a process for obtaining external data. UD Connect is the tool for achieving UDI. Throughout this article, we'll use the term UDI when discussing the process and UD Connect when discussing the toolset.

Prior to SAP NetWeaver 2004, SAP offered DB Connect functionality to access third-party information; however, DB Connect was restricted to databases supported by SAP. If a customer had data in a database that SAP didn't support or in a third-party system, such as a Teradata data warehouse or Microsoft Analysis Services, you had to move the data from the third-party system to flat files and then upload those flat files into the SAP NetWeaver BI data warehouse.

With the release of SAP NetWeaver 2004, SAP introduced UD Connect, which enables you to move data into an SAP data warehouse from a broader range of sources. UD Connect allows access to almost all multidimensional and relational database systems. **Figure 1** (on the next page) provides some examples of third-party systems from which UD Connect enables you to obtain data. The list isn't comprehensive, but does give you a wide range of options to consider. If you are interested in finding out whether a particular system has any of the available connectors, you will have to check the documentation from that vendor.

How does UD Connect work? It uses SAP's BI Java Connectors to integrate external data. The BI Java Connectors allow you to directly access any relational and multidimensional data sources in BI. BI Java Connectors are powerful because they are based on industry standards for interoperability, such as JDBC, XMLA, and ODBO. Some of these connectors require drivers from a third-party vendor. For example, to use BI Java Connectors to connect to an Oracle database via JDBC, you need to install JDBC drivers. The third-party system manufacturer (in this case, Oracle) provides the JDBC drivers. For connectivity using JDBC, you always need third-party drivers. XMLA, however, is an open interface that doesn't require any third-party drivers to connect. Therefore, BI Java Connectors allow companies to integrate any industry-compliant data into SAP NetWeaver BI.

In **Figure 2** (on the next page), you can see where UD Connect fits into the SAP NetWeaver architecture, as well as the roles of the four BI Java Connectors.

Connector type	Connector	Access to	Technology based on	System requirements
Relational	BI JDBC Connector	Relational data sources: more than 170 JDBC drivers Examples: Teradata, Oracle, Microsoft SQL Server, Microsoft Access, DB2, Microsoft Excel, text files such as comma-separated values (CSV)	Sun's Java Database Connectivity (JDBC): the standard Java API for Relational Database Management Systems (RDBMS)	JDBC driver
Relational	BI SAP Query Connector	SAP operational applications Examples: Data in transactional systems, such as SAP R/3, ad hoc, and operational reporting	SAP Query: a component of SAP NetWeaver Application Server (AS) that allows you to create custom reports without any ABAP programming knowledge	SAP Java Connector (SAP JCo)
Multi-dimensional	BI ODBO Connector	OLAP data sources: OLE DB for OLAP-compliant data sources Examples: Microsoft Analysis Services, Software as a Service (SaaS), Microsoft PivotTable Services	Microsoft's OLE DB for OLAP (ODBO: the established industry-standard OLAP API for the Windows platform	Microsoft Windows 2000/NT/XP/ Vista
Multi-dimensional	BI XMLA Connector	OLAP data sources Examples: Microsoft Analysis Services, Hyperion, MicroStrategy, and SAP BW 3.x	Microsoft's XMLA (XML for Analysis): Web services-based platform-independent access to OLAP providers; exchanges analytical data between a client application and a data provider working over the Web, using a SOAP-based XML communication API	None

Figure 1 Examples of different connectors and systems you can use with UD Connect

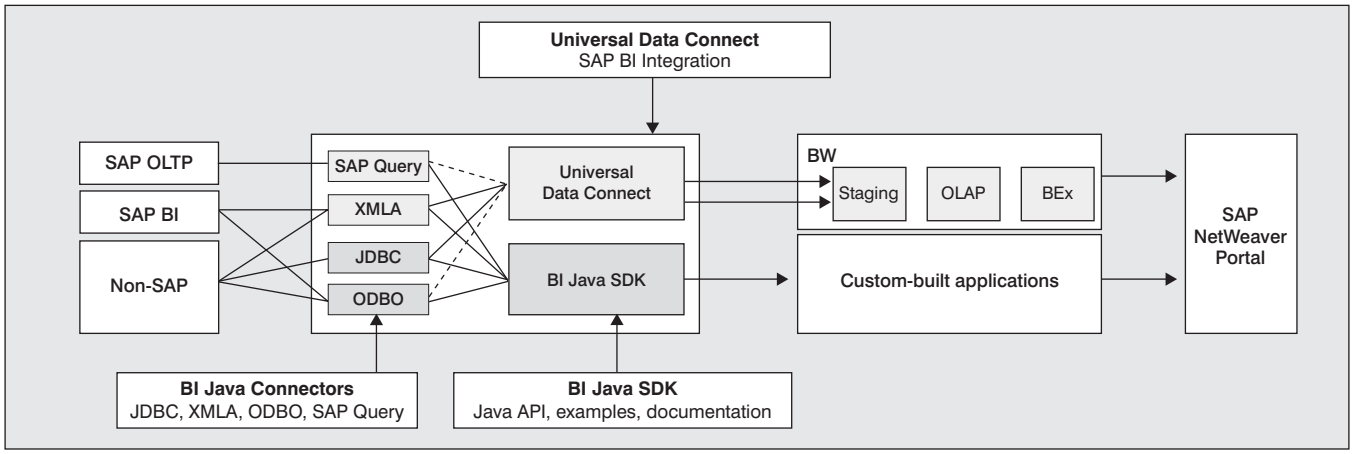


Figure 2 UD Connect in the SAP NetWeaver 7.0 architecture

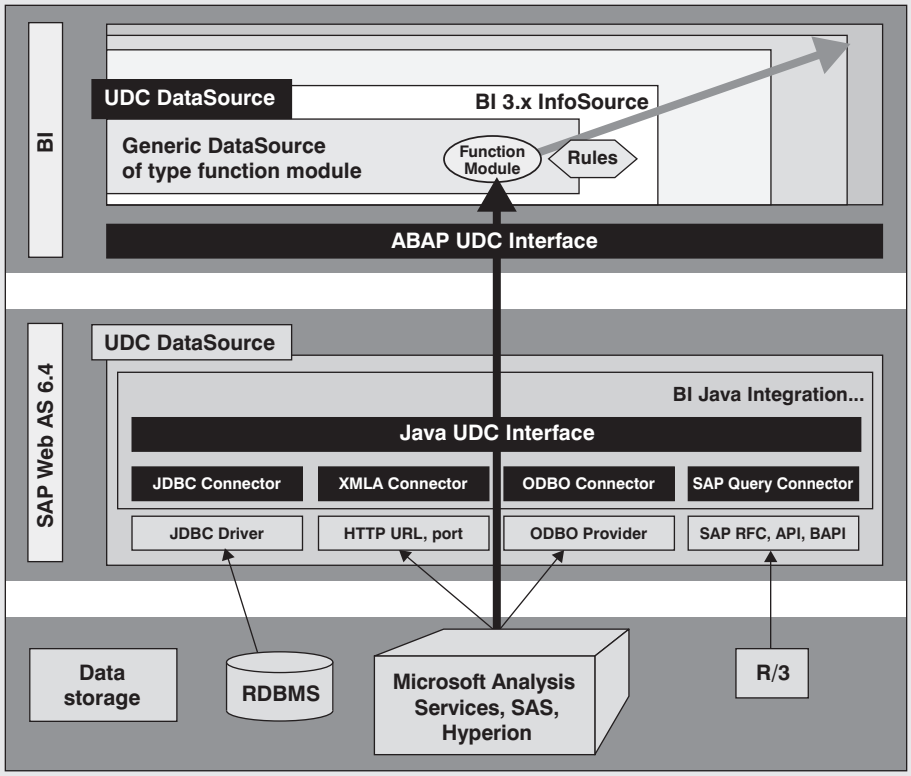


Figure 3 Location of the BI Java drivers (below the BI Java Connectors layer)

UDI uses the BI Java Connectors to move data into the data-warehousing layer. ODBO, JDBC, and XMLA are the three BI Java Connectors being used for non-SAP data. It's important to understand what external data is accessible and to have a firm grasp of the BI Java Connectors. For example, if you acquired a company that had an IBM DB2 database, how would you connect to it? Which BI Java Connector would you use: XMLA, JDBC, or ODBO? Do you need BI Java drivers? If so, which ones? These are extremely important questions to be able to answer when setting up UDI scenarios.

> ***Note!***
>
> For IBM DB2, you would use JDBC because DB2 is a relational data source (**Figure 1**). Because you're using JDBC, you also need BI Java drivers that you can get from IBM.

Figure 3 shows where the BI Java drivers are located (just below the BI Java Connectors layer). You need to load the BI Java drivers on the Java stack, not on the ABAP stack. It is important to understand this architecture to determine how to set up UD Connect. To configure the infrastructure for UD Connect, use the Java stack. If you're using SAP NetWeaver 2004, you will have the SAP Web Application Server (AS) 6.40, whereas if you're using SAP NetWeaver 7.0, you will use the SAP NetWeaver Web AS 7.0.

You configure the actual UD Connect scenario (i.e., the information about which data to move) on the ABAP side.

Note!

The BI Java drivers provide access to industry-standard drivers via the connection architecture on the J2EE Server (6.40 or 7.0, depending on your release). The J2EE Server is SAP's Java stack for deploying Java-based applications. The BI Java drivers are traditionally provided by third-party providers for external data and by SAP for internal connections.

Note!

The BI Java SDK is SAP's Java application programming interface (API) used to access the BI Java Connectors. Using this API, customers can build Java applications that connect to third-party data as UD Connect does.

Why use BI Java Connectors?

One of the biggest benefits to using BI Java Connectors is their universality. UD Connect uses these connectors, and any Java application you build with the BI Java SDK can also use them. The BI Java Connectors offer uniform connection management, monitoring, and a foundation for all connectors. This means that you can troubleshoot and monitor all applications using these connectors centrally on the SAP NetWeaver platform.

Now that you understand UDI, UD Connect, BI Java Connectors, and BI Java drivers, you need to know how to set up and configure these components. The next section gives you an overview of how to set up and configure UD Connect for use with the BI JDBC drivers for Microsoft SQL Server. For a list of how to configure all UD Connect connections, see Section 4 of the how-to guide on the SAP Developer Network (SDN) at www.sdn.sap.com.

Setting up and configuring UD Connect

Setting up UD Connect is fairly straightforward. You just need to complete three steps:

1. Install and configure the BI Java drivers.
2. Configure the UD Connect source system.
3. Build transient or persistent modeling.

To help illustrate each of these steps, let's assume you need access to a Microsoft SQL Server database. This type of database, like many other industry standards, is a relational database that provides JDBC drivers.

Step 1: Install and configure the BI Java drivers

To configure the BI Java drivers on the Java stack, you use the SAP NetWeaver Visual Administrator tool. Make sure there is a valid and functioning JDBC-compliant driver for the third-party database. Most databases have published JDBC-compliant drivers. This article uses the example of connecting to Microsoft SQL Server. Microsoft publishes JDBC service packs for its different database releases, and the drivers are available from the Microsoft download center.

Once the database drivers and all relevant service packs have been downloaded and the .jar files associated with the driver are accessible on your file system, you're ready to configure the drivers.

1. Log in to SAP NetWeaver Visual Administrator, and navigate to the JDBC Connector service from the Cluster tab. The JDBC Connector service is where you add the JDBC driver to the Java stack for SAP NetWeaver. To do this, all .jar files must be available on the file system. These .jar files make up the JDBC driver for the database in question. **Figure 4** shows that the JDBC driver contains three different files: msbase, mssqlserver, and msutil. The SAP NetWeaver stack prompts you to upload all files relevant to the driver after you give it an appropriate name, such as MSSQLDriver.

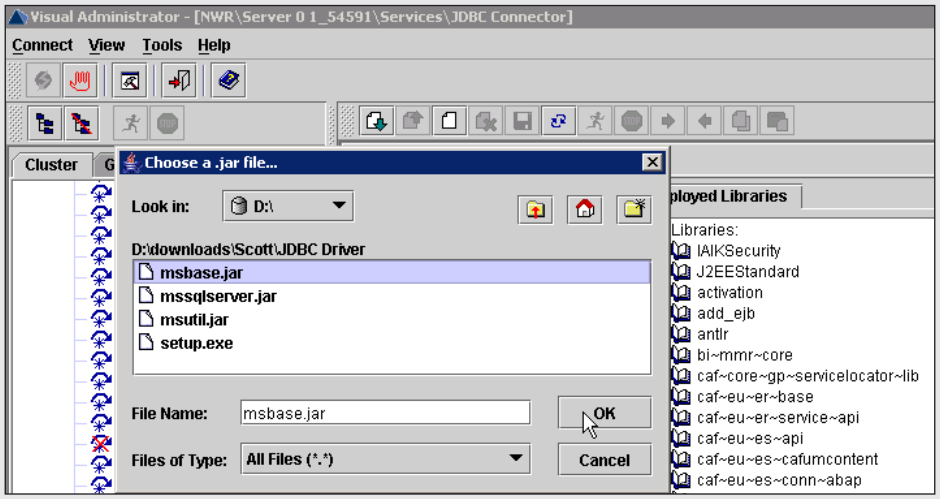


Figure 4 SAP NetWeaver Visual Administrator, JDBC Connector service

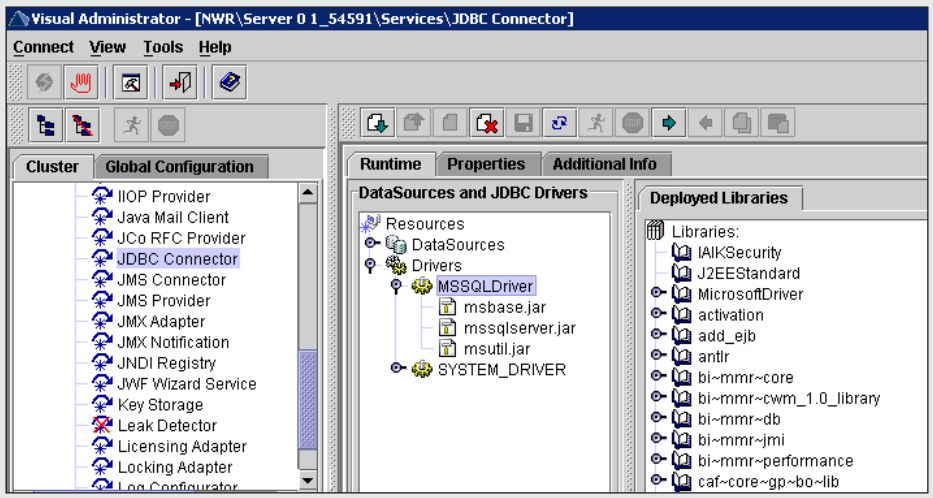


Figure 5 SAP NetWeaver Visual Administrator, JDBC Connector service complete

> ***Note!***
>
> The user interface (UI) of SAP NetWeaver Visual Administrator allows you to select only one file at a time. After each file uploads, it prompts you to either select additional files or complete the upload.

2. Ensure that all of the files you chose in the first step are present in the driver hierarchy that you named. For example, as shown in **Figure 5**, the files msbase.jar, mssqlserver.jar, and msutil.jar appear in the JDBC drivers' area under the name MSSQLDriver.
3. Now, you can set up the Java Connector because MSSQLDriver has been installed. On the Cluster tab, select the Connector Container service, as

shown in **Figure 6**, to display all the connectors installed on your Java stack.

By default, each connector type listed has its own single connector setup. You see in **Figure 7** that JDBC, XMLA, SAP Query, and ODBO all have connector types available.

Each connector appears with a default name that begins with the prefix SDK. These default connectors give you a starting point that you can clone and modify for your own use. There should always be a delivered SDK JNDI[3] name for you to copy. For example, the default SDK is "SDK_JDBC".

To set up the JDBC connector container for the Microsoft SQL Server 2000 database, you first need to clone the SDK connector container. To do this, click on the Clone button (), and enter the name of the resource adapter, as shown in **Figure 8**.

You give the connector container a unique name (in the example, SDK_MSSQL_JDBC), which specifies that you're connecting to Microsoft SQL Server (you can enter any unique name here that starts with SDK). To use a given connector with UD Connect, the name of the cloned connector must also begin with the prefix SDK. If it doesn't, BI will be unable to associate itself with the cloned connector.

4. After cloning the default container, you need to add the proper references to the new connector container so that both the connector and its driver load at the same time. To add the appropriate reference for the SDK_MSSQL_JDBC connector, switch to the Resource Adapter tab of the SDK_MSSQL_JDBC connector container. From there, you can add new loader references.

   As shown in **Figure 9**, you can add the references to the BI Java drivers (e.g., library:MSSQLDriver) to connect your JDBC driver (**Figure 4**) to the cloned connector container you are building.[4]

[3] JNDI stands for Java Naming and Directory Interface.

[4] To identify the appropriate references, please see the documentation provided with the driver from the third-party vendor.

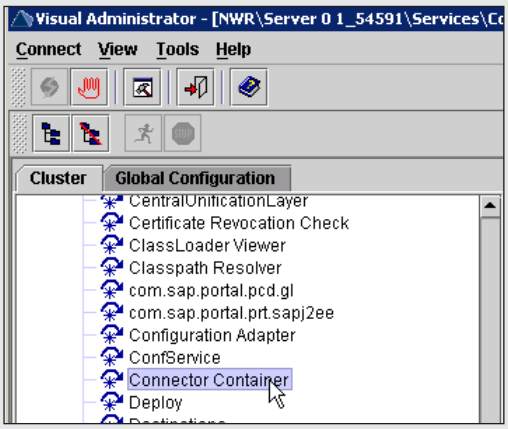


Figure 6 Selecting the cluster option "Connector Container"

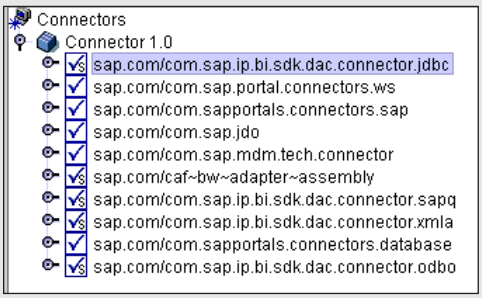


Figure 7 BI Java Connector types

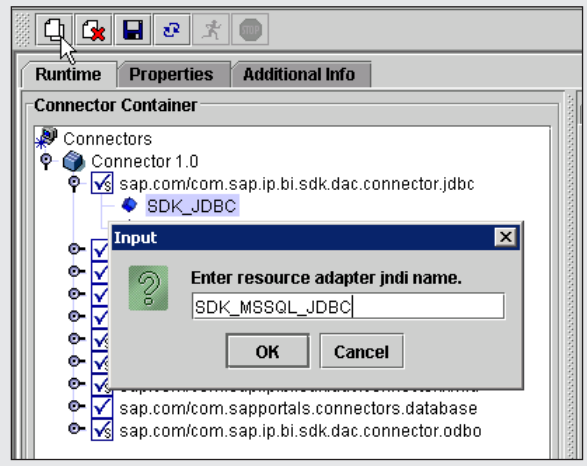


Figure 8 Cloning the template connector container to create your own SDK connection

By adding this reference, you associate the connector with the driver, which you already loaded to the J2EE engine (**Figure 5**). You still need to provide the connector with more information about the Java stack and the third-party database so that it can communicate successfully. We'll cover this in step 2.

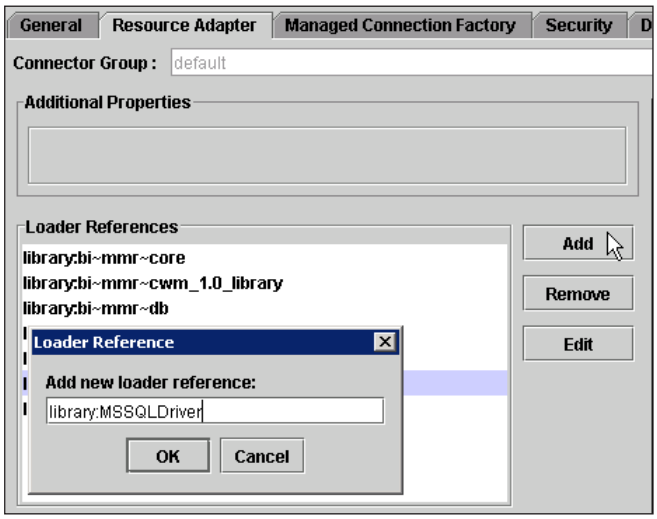


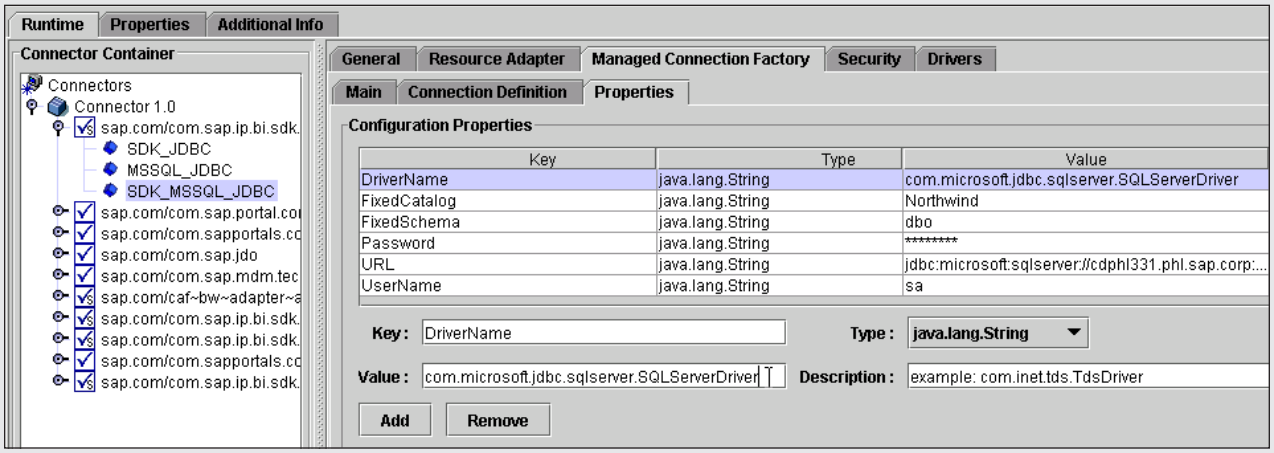
Figure 9 Adding a loader reference to let you associate the connector with the driver

When using connections from the connector container within SAP NetWeaver Portal, you cannot create user mapping; instead, you should populate the connector with authentication information from the properties of the connector container's "managed connection factory." Once you've populated the required information (the connection URL, user name, password, and driver name) into the Java stack, you will be able to use SAP NetWeaver BI to extract this third-party data.

Each of the four types of connectors that constitute BI UDI has its own test methods. Embedded within the Java stack is a group of servlets that allow Java stack administrators to run connection tests on the connector containers being configured. **Figure 10** (on the next page) provides a list of connectors and their test pages. In the Microsoft SQL Server database example, you would use http://<host>:<port>/TestJDBC_Web/TestJDBCPage.jsp where <host> equals the host of your BI Java Server, and <port> equals the port of your BI Java Server.

Note!

The authentication of the database to which you connect will require the user name and password, as shown in the screenshot below. This connection information should include the documentation you get from the third-party BI Java driver vendor.



If the settings you enter while configuring JDBC connectivity test successfully, the servlet returns a list of the tables available within the third-party database. A test of the SDK_JDBC connection shows you how to get the tables from a Microsoft SQL Server database, as shown in **Figure 11**.

Next, you need to configure the UD Connect source system on SAP NetWeaver BI's ABAP stack so you can make use of the new connector to employ your third-party data.

Step 2: Configure the UD Connect source system

To configure the UD Connect source system in SAP NetWeaver 7.0, run transaction RSA1 on the ABAP

Connector	URL	Successful result
BI JDBC Connector	http://<host>:<port>/TestJDBC_Web/TestJDBCPage.jsp	Displays a list of tables
BI ODBO Connector	http://<host>:<port>/TOdbo/servlet/TestOdbo	Displays a list of cubes
BI SAP Query Connector	http://<host>:<port>/TSapq/servlet/TestSapq	Displays a list of tables
BI XMLA Connector	http://<host>:<port>/TXmla/servlet/TestXmla	Displays a list of cubes

Figure 10 BI UDI test servlets for the four types of connectors

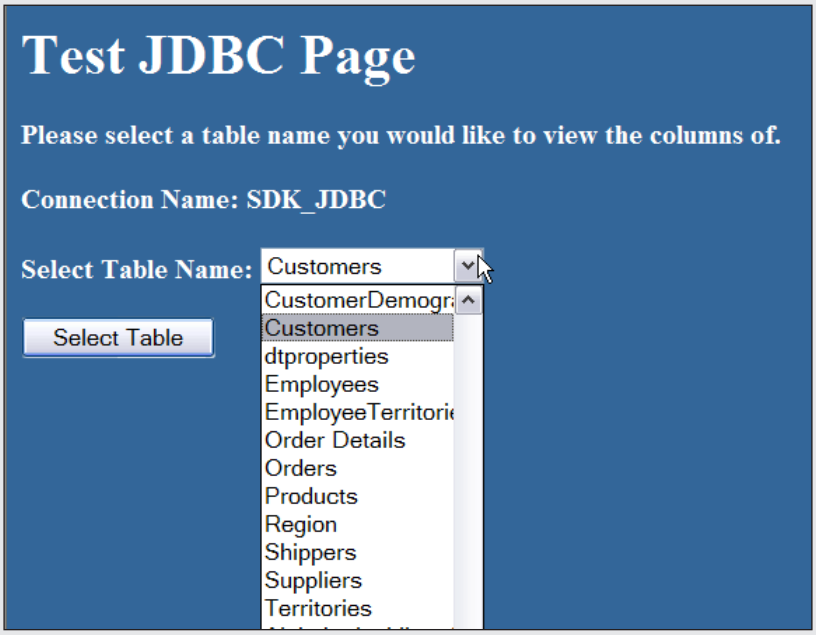


Figure 11 Successful connection test servlet

system. You can use the source system to stage the data from the third-party database. In **Figure 12**, select the UD Connect option, and then choose the connection that you created on the Java stack (that's why it was important to start this name with SDK; otherwise, it wouldn't show up on this list).

Next, you need to build either transient or persistent modeling so users can access the third-party data.

Step 3: Build transient or persistent modeling

Now that your source system is configured, it's time to build your dataflow. First, create a DataSource by right-clicking on your component and then selecting Create DataSource, as shown in **Figure 13**. This DataSource will utilize the SDK that we created earlier.

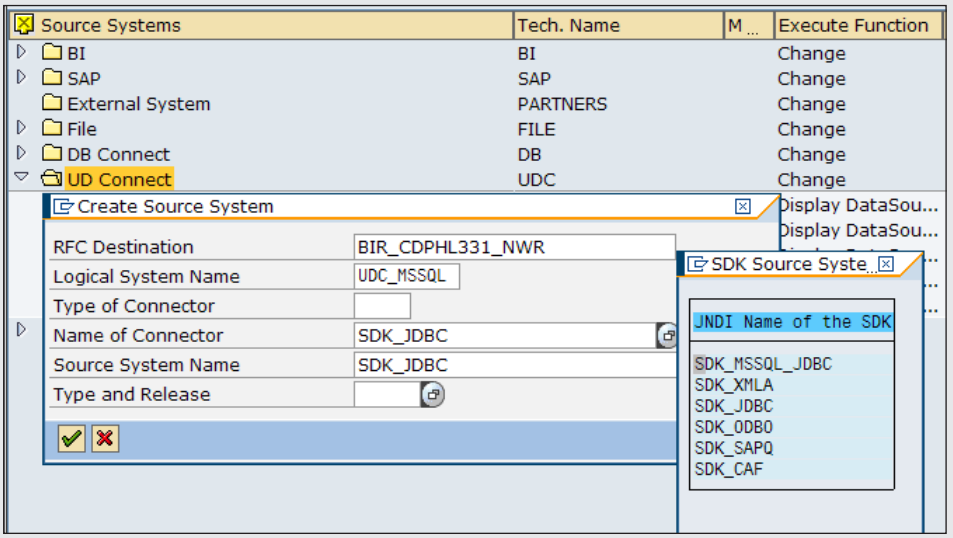


Figure 12 Creating and configuring the UDI source system

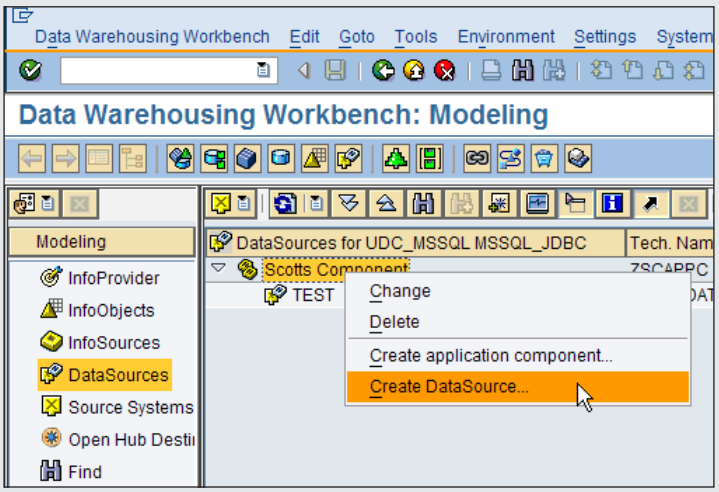


Figure 13 Creating a DataSource from RSA1

Then, from within your DataSource, select the UD Connect source object, for example, VINVOICES_00 as shown in **Figure 14**.

To choose the VINVOICES_00 object, you can use the help pop-up on the right side of screen. Here you can find all the tables and views available on the Microsoft SQL Server database. Microsoft, for example, delivers a demo database called Northwinds, which is the database you have connected to in this example.

Finally, activate your DataSource, and then you can use it for the mergers and acquisitions scenario described in the next section.

A mergers and acquisitions scenario

Now that you have configured UD Connect for use, let's look at a typical scenario in which you need to integrate data from different companies (using CompanyA and CompanyB as described earlier). After helping many SAP customers integrate data from a wide variety of sources, we have found that a four-phase approach works well for this kind of project. This approach employs key SAP NetWeaver capabilities, specifically UD Connect and SAP NetWeaver MDM:

- **Phase 1:** Building a direct UDI scenario from the front-end toolsets
- **Phase 2:** Developing a model UDI for transient data
- **Phase 3:** Staging persistent data warehousing through UD Connect within data extraction, transformation, and loading (ETL)
- **Phase 4:** Executing a data unification project

The primary reason for these four phases is to minimize disruption and allow the business users that need access to this third-party data to get it immediately after acquisition along with a step-by-step approach to aligning the IT infrastructures and the master data. Let's walk through the four phases in more detail.

Phase 1: Building a direct UDI scenario from the front-end toolsets

When faced with integrating heterogeneous data, companies want a familiar environment that addresses their specific business needs. In our example, for

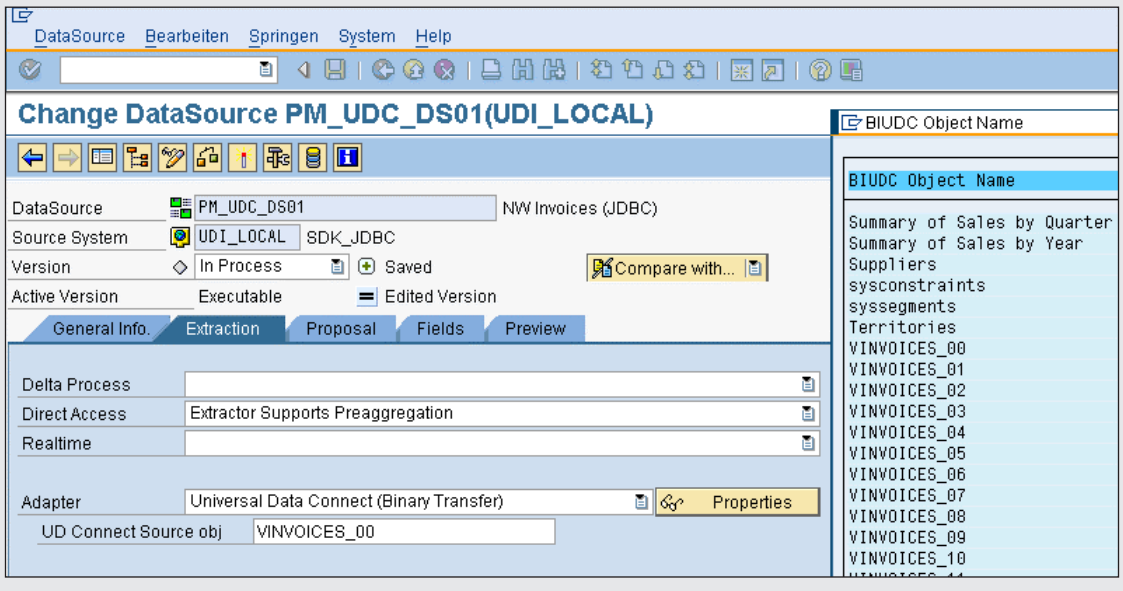


Figure 14 UD Connect source object

instance, CompanyA wants to have immediate access to CompanyB's data via tools with which they are familiar. They don't want to learn CompanyB's IT systems. You can expand this scenario to as many different third-party datasets as exist, but for the sake of simplicity, let's focus on just this one scenario: two different companies, each with a single set of IT systems.

In Phase 1, you select the UI and toolset you want to use for your integration project. SAP NetWeaver 7.0 provides two different toolsets in the front end that use UD Connect transiently: the BEx Web Analyzer and SAP NetWeaver Visual Composer.[5]

You can use BEx Web Analyzer to access data from either system. For example, CompanyA can run BEx Web Analyzer to choose a report from SAP NetWeaver BI or Microsoft Analysis Services.

If you want to run a report in BEx Web Analyzer, first you need to specify your source system as either SAP or non-SAP. As shown in **Figure 15**, we selected the BR9CLNT000 system as the source system for CompanyA. To get to this screen, launch BEx Web Analyzer and then open the dialog as described in SAP Help Portal at www.help.sap.com.

As of SAP NetWeaver 7.0, the BEx Web Analyzer allows you to use an ODBO system or an XMLA system, but not a JDBC system (relational data access). Visual Composer supports all the BI Java drivers. In **Figure 16** (on the next page), you can see how both the BEx Web Analyzer and Visual Composer can transiently access third-party data. (We'll refer back to how the data warehouse interacts with this third-party data in Phases 2 and 3.)

The benefit of this scenario is that it doesn't require moving any data. The data is persisted in the back-end source system, but these tools can access it without any staging work. For example, CompanyA uses the BEx Web Analyzer in SAP NetWeaver to access the data in Microsoft Analysis Services. Allowing CompanyA to use a familiar system offers them a consistent user-reporting experience. It is the quickest way to integrate data using a single toolset and, therefore, is Phase 1 of the integration roadmap.

The data from CompanyB hasn't yet been merged with CompanyA's existing data. Since the data is still in Microsoft Analysis Services and doesn't need to be moved to SAP NetWeaver, CompanyA users can use BEx Web Analyzer almost immediately after acquisition to access CompanyB's information with minimal

[5] Discussion of these applications goes beyond the scope of this article, but for a description of them, see the *SAP Insider* article "Visual Composer or BEx? Pick the Right Tool for the Right Application" (July-September 2006). Also, see the *SAP Professional Journal* articles "Get started creating SAP Enterprise Portal iViews with Visual Composer — a purely model-driven, code-free development approach" and "Advanced techniques for enhancing your SAP Enterprise Portal iViews with Visual Composer — a purely model-driven, code-free development approach" by Karl Kessler (November/December 2005).

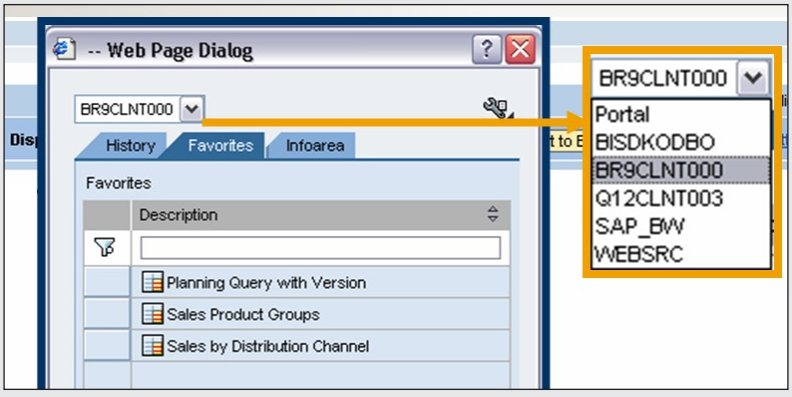


Figure 15 Selecting a source system in BEx Web Analyzer

effort. The business users can report directly from BEx Web Analyzer without burdening IT.

At this point, you already have a consistent user tool, as well as experience, for accessing the data for both CompanyA and CompanyB. The downside is that you can't merge the data from the two companies into one report. This is where Phase 2 comes in.

Phase 2: Developing a model UDI for transient data

This phase addresses the problem of integrating the metadata models of the two companies. For example, CompanyA might have an object called "material" while CompanyB calls the same object "product." At this stage, you can map these objects to a consistent metadata model. To do this, you actually create objects (InfoObjects and InfoCubes) in your SAP NetWeaver BI system. Phase 2 has quite a bit more work than Phase 1 does, but you have more time in which to do it because the business can report on the third-party data via the activities in Phase 1.

The basis of this scenario is a VirtualProvider, also referred to as a Virtual InfoProvider.[6] VirtualProviders are typically used to access data transiently. In the example scenario, CompanyA builds VirtualProviders to access the data from CompanyB via UD Connect. For example, if you want to use a financial cube from Microsoft Analysis Services at CompanyB, the cube will be built as a VirtualProvider in SAP NetWeaver BI. For each Microsoft Analysis Services cube you have in CompanyB, you build one VirtualProvider in SAP NetWeaver BI. Each VirtualProvider is then attached to its own MultiProvider,[7] and you build SAP NetWeaver BI queries on this MultiProvider, for example, a report on CompanyB's Financials.

The MultiProvider enables you to build consistent reporting using SAP NetWeaver BI tools, such as the BEx Query Designer. It's very

[6] For details on VirtualProviders, go to the SAP NetWeaver 7.0 (2004s) documentation on the SAP Help Portal at www.help.sap.com.

[7] For details about MultiProviders, go to the SAP NetWeaver 7.0 (2004s) documentation on the SAP Help Portal at www.help.sap.com.

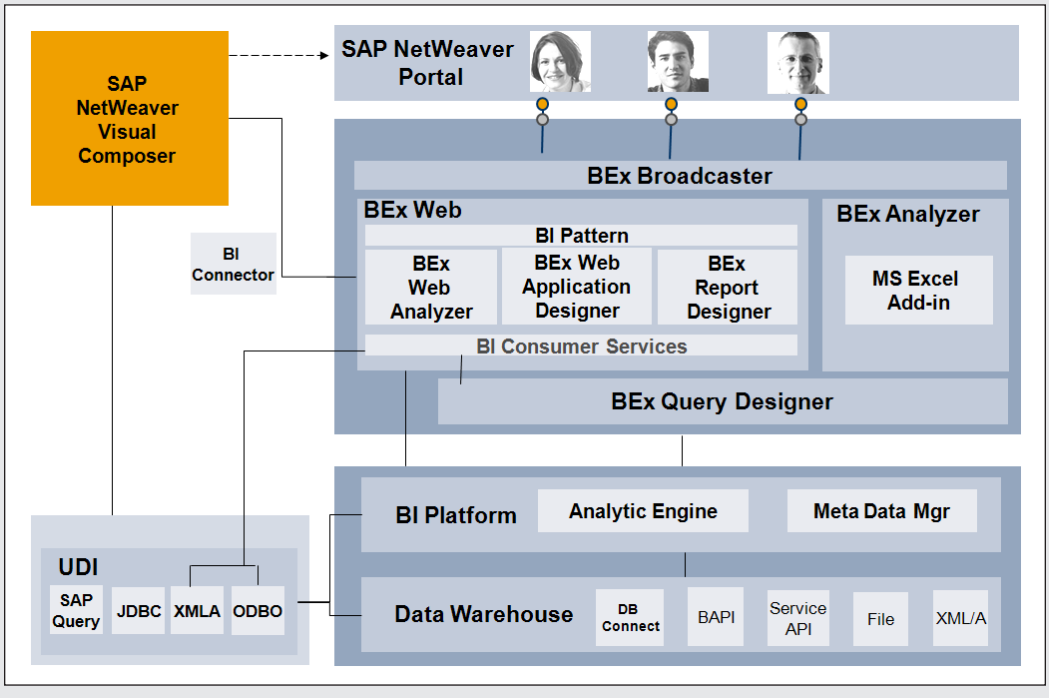


Figure 16 Connections to BI Java Connectors from front-end tools

important that you build these queries on the MultiProvider and not directly on the VirtualProvider (we'll explain why in Phase 3). Once you have the modeled layer and the query, you can access the data transiently. The data is stored in Microsoft Analysis Services until you move or load it to SAP NetWeaver.

VirtualProviders don't persist data. They allow you to create modeling with no necessary data ETL. Also, if you have master data that's unique to each system and doesn't overlap, you can have MultiProviders that bring together your SAP NetWeaver BI data and the external data in a single query. If the master data isn't unique, you will need to look at Phase 4 about Data Unification. The benefits of this approach are that you can build the framework for the next step and your users can get a greater advantage from using any of the BI tools (BEx Analyzer, Query Designer, Web Analyzer, Report Designer, or Web Application Designer) on top of this data.

In **Figure 17**, we see that the cube for CompanyA is in the SAP NetWeaver BI system. The cube for CompanyB is in Microsoft Analysis Services. The VirtualProvide is used via UD Connect to get data transiently from the CompanyB cube (third-party data) into our SAP NetWeaver BI reporting environment. Using the MultiProvider, you can merge data from your CompanyA cube and your VirtualProvider for CompanyB. Then you can build BEx Queries to report on data from both CompanyA and CompanyB. In this scenario, we have not ensured that our master data is consistent, so we still have separate queries for CompanyA and CompanyB data. The reporting tools and UI, however, are identical so the users can use one set of reporting tools.

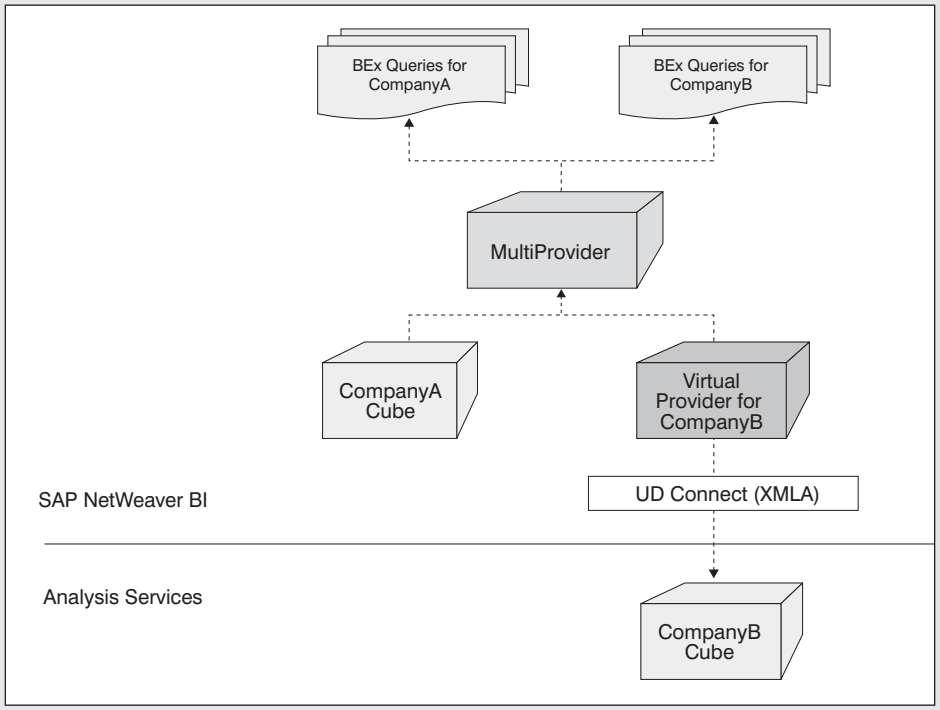


Figure 17 Data model for CompanyA and CompanyB in the mergers and acquisitions scenario

Phase 3: Staging persistent data warehousing (UD Connect within ETL)

In this phase, you finally build your data warehouse. You have set up the Virtual InfoProvider, and you have your queries and reporting built on top of the MultiProvider. You can create a new InfoCube into which you load data via UD Connect, as shown in **Figure 18**. Then, you switch from using the VirtualProvider to using this new InfoCube by attaching it to the MultiProvider and deleting the VirtualProvider. This requires no rewriting of reports and allows you to disconnect the legacy system from the company you acquired. All the information is now in your central data-warehousing system.

The primary reason for building your reports on top of the MultiProvider is so you won't have to change your BEx Queries. As you can see in **Figure 17**, you have a VirtualProvider. If you had built the BEx Queries on top of this VirtualProvider, then when you moved to persistent staging and removed the VirtualProvider, those queries would no longer work. Because you built the BEx Queries on top of a MultiProvider, you just have to point the MultiProvider to the new CompanyB cube instead of the VirtualProvider, and all the BEx Queries will continue to work. Those users who have built queries for CompanyA's and CompanyB's information don't have to change them.

Phase 4: Executing a data unification project

In the end, if you bring together data from different companies, you're going to have to consider a data unification project, merging the master and metadata from the new system with the master data of your existing system. This is the final step in building a true enterprise data warehouse.

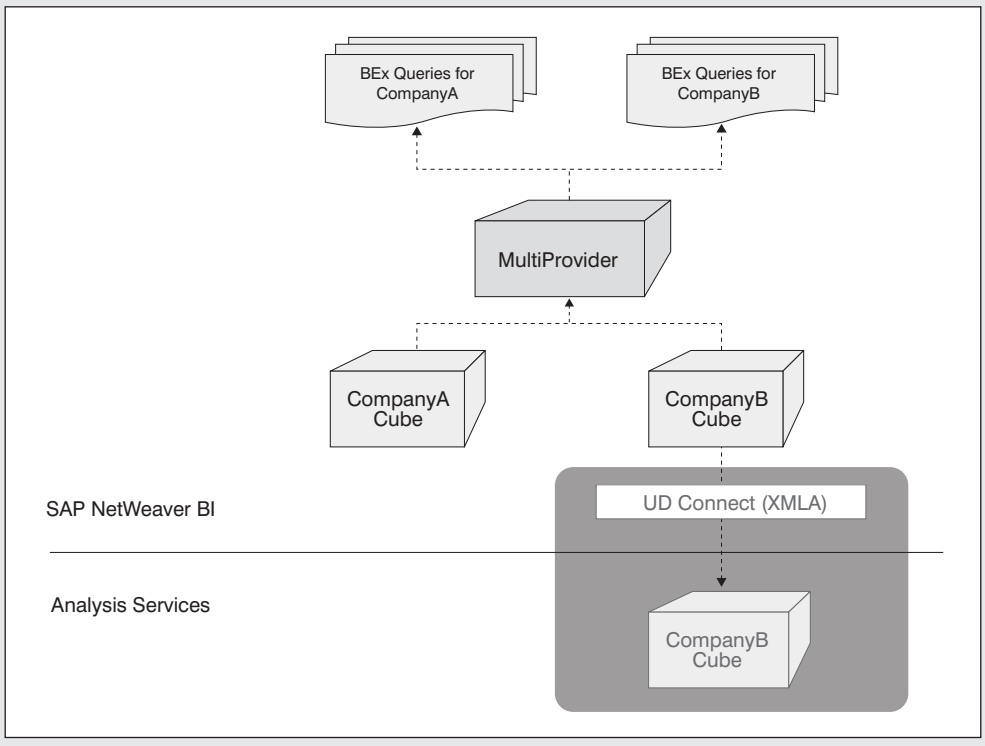


Figure 18 Replaced VirtualProvider with CompanyB cube and removed Microsoft Analysis Services cube

Every time a company is acquired, one of the tasks considered most daunting is integrating the new metadata and master data into the preexisting data warehouse, which already contains all its own metadata and master data. While SAP NetWeaver offers a tool, MDM, to help ease this task, it is still difficult. Although many IT departments carry the burden of organizing and integrating this new information, the business has to play a central part in this process to succeed. When dealing with new systems in an existing environment, always start with consolidation. The best way to ensure an ongoing flow of consistent data is to have good data governance policies in place.

To integrate new data into an existing centralized Master Data Management, or MDM[8] system, you need to complete a three-step process, as illustrated in **Figure 19**:

1. Consolidation
2. Harmonization
3. Central MDM

Let's walk through these steps.

1. Consolidation

Initially, within the process of data consolidation, you need to check whether the preexisting data inside your MDM system repositories matches any of the new data coming from the newly acquired company. MDM contains repositories, which are simply a set of database tables that represent all of your master data for a given object (e.g., product, business partner). You might have business partners in common, vendors or suppliers that the two companies refer to differently, or similar products with different brand names. These differences need to be addressed and consolidated to control redundancy and duplication at least for internal reporting. Once this information is consolidated in each MDM repository object, such as product or business partner, the repository will contain few, if any, duplicates.

[8] A detailed discussion of this three-step process of MDM goes beyond the scope of this article, so for more information, we recommend that you read Klaus David's article, "Centralize, harmonize, and distribute your master data with SAP NetWeaver Master Data Management (MDM)" (*SAP Professional Journal*, May/June 2006).

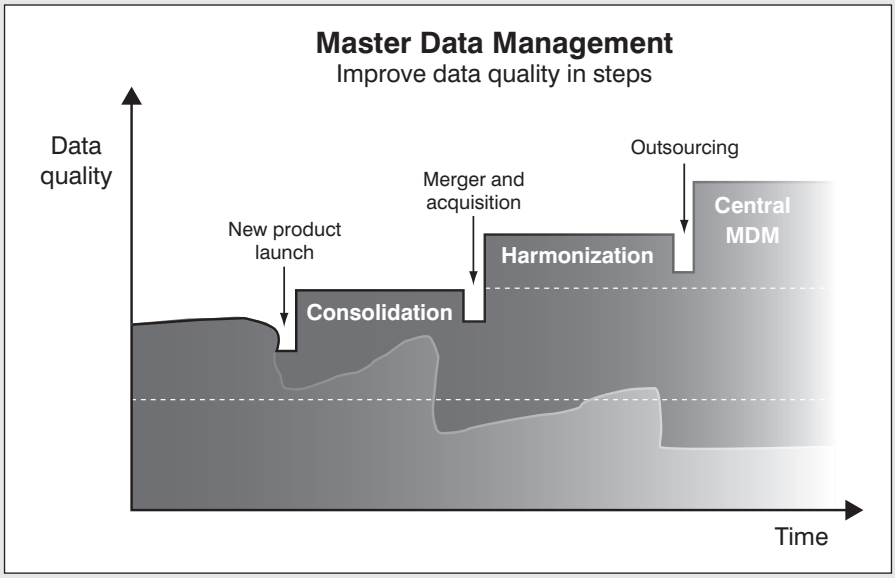


Figure 19 The three steps of MDM

UD Connect helps you bring your data together, but it doesn't resolve any overlaps in master data between CompanyA and CompanyB (i.e., the facial tissue/water bottle example). You won't be able to report consistently across the details by business partner or product. Another potential problem is where you have common customers. If both CompanyA and CompanyB worked with Wal-Mart (which is highly likely in the CPG space), Wal-Mart should have the same customer number in both sets of data. It probably doesn't, and you'll need to resolve this. SAP NetWeaver MDM 5.5 can help you here.

To help you with both the technical and semantic portions of integrating the data, you can use the tools provided in SAP NetWeaver MDM 5.5, such as the Import Manager. It allows you to define extensive mapping rules that delineate *n*-to-*n* relationships between a source of information and its target object or repository within SAP NetWeaver MDM.

You can also use the Import Manager to create a map file that describes the relationship between fields and values. For example, CompanyA may have a field called "material" and CompanyB may have a field called "product." Both of these fields could refer to products the two companies create and sell. In addition to the information stored in the map file, you might need to specify the type of matching that you want performed when the imported data is stored. For example, what should happen to an existing CompanyA record if a match to an imported CompanyB record is found? Should the record be updated, overwritten, or skipped? Configure your map file in such a way that it handles as many different mapping scenarios as possible.[9]

In **Figure 20**, you can see what the MDM Import Manager looks like.

During data consolidation the main goal of the import process is to get all the information into the repository with as little duplication as possible. If after the import there are still records with similar semantic meanings

[9] For details on how to configure and build the map file and how to work with the Import Manager, see the details in the "MDM Import Manager Reference Guide" on the SAP Help Portal at www. help.sap.com.

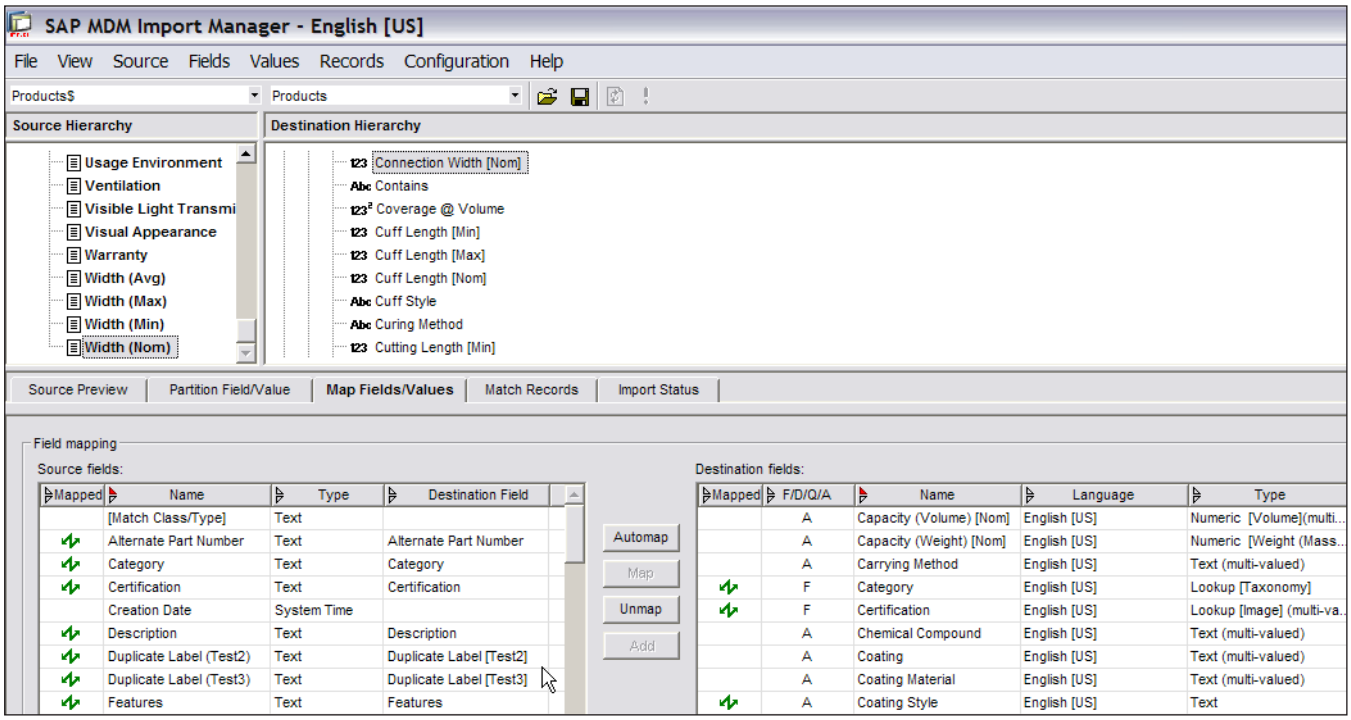


Figure 20 MDM Import Manager, which is used to map fields and values

(e.g., "product" and "material"), SAP NetWeaver MDM contains a series of merging and matching rules that you can apply to the newly imported data to eliminate any existing redundancy that might exist in the data.

You configure the matching and merging rules from within the merging mode of the MDM Data Manager, as shown in **Figure 21**. After a successful import from a spreadsheet, a database, or another source system, you can apply merging and matching rules against the newly imported records. This merging mode in MDM enables you to apply a matching strategy against a dataset. A matching strategy consists of rules and transformations.

For example, take the strategy MATCH_NAME_ADDRESS: This particular strategy is used to match names and addresses of newly imported records to those already existing within the repository. The strategy is made up of a series of scoring rules:

- NAME
- POSTAL CODE
- CITY
- COUNTRY
- STREET

Each rule contains a scoring mechanism that is later used to evaluate each individual record and give it a probability of being a match.

As you can see in **Figure 22** (on the next page), the Transformations screen shows the vendor names that were imported with this dataset. You need to "transform" (that is, cleanse) this data before you can compare it with the records in the existing repository. You want to drop the corporation designation, for example, "corp," "inc," "llc," etc, which may follow a company name. Once the data is cleansed, you can apply rules to this new dataset to compare it with the existing records.

In **Figure 23** (on the next page), the Rules screen shows the rule that is applied based on the function value. The rule's function value will determine whether NAME is compared as a whole or if each individual token within NAME is tested for a match. The Equals function performs better here because the

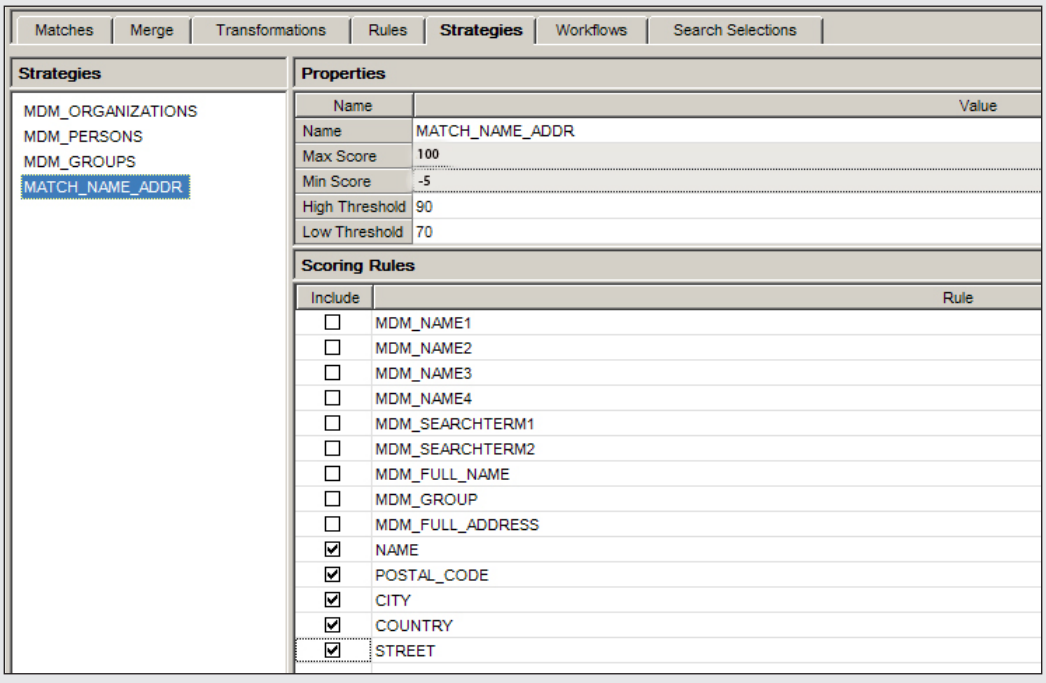


Figure 21 Strategies screen in the MDM Data Manager

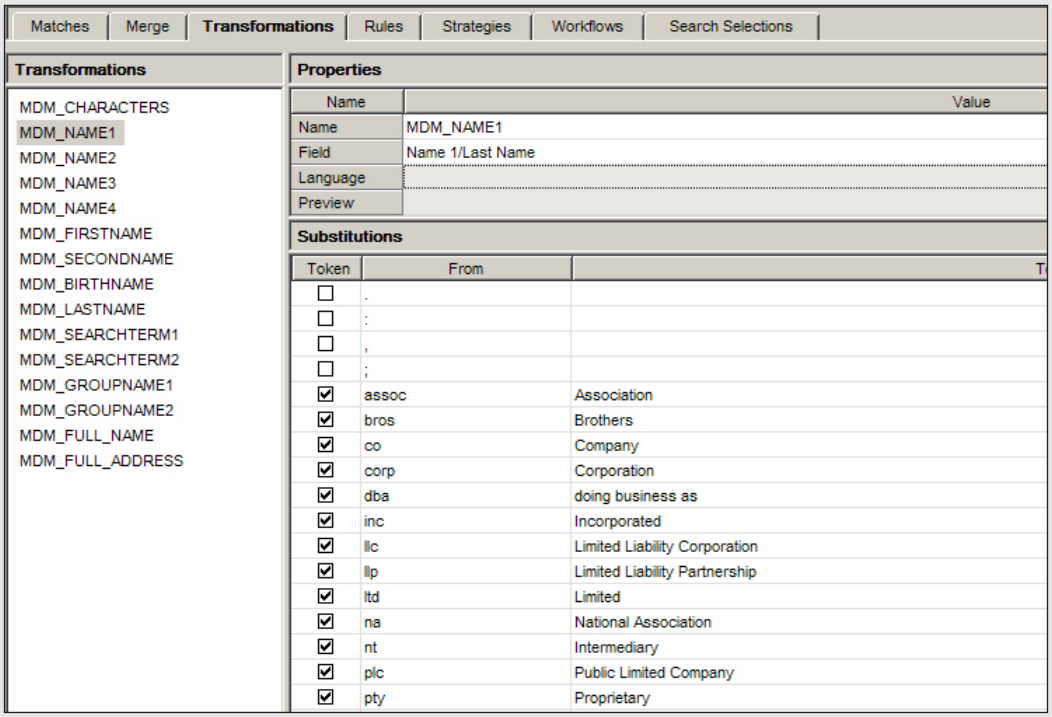


Figure 22 Transformations screen in the MDM Data Manager

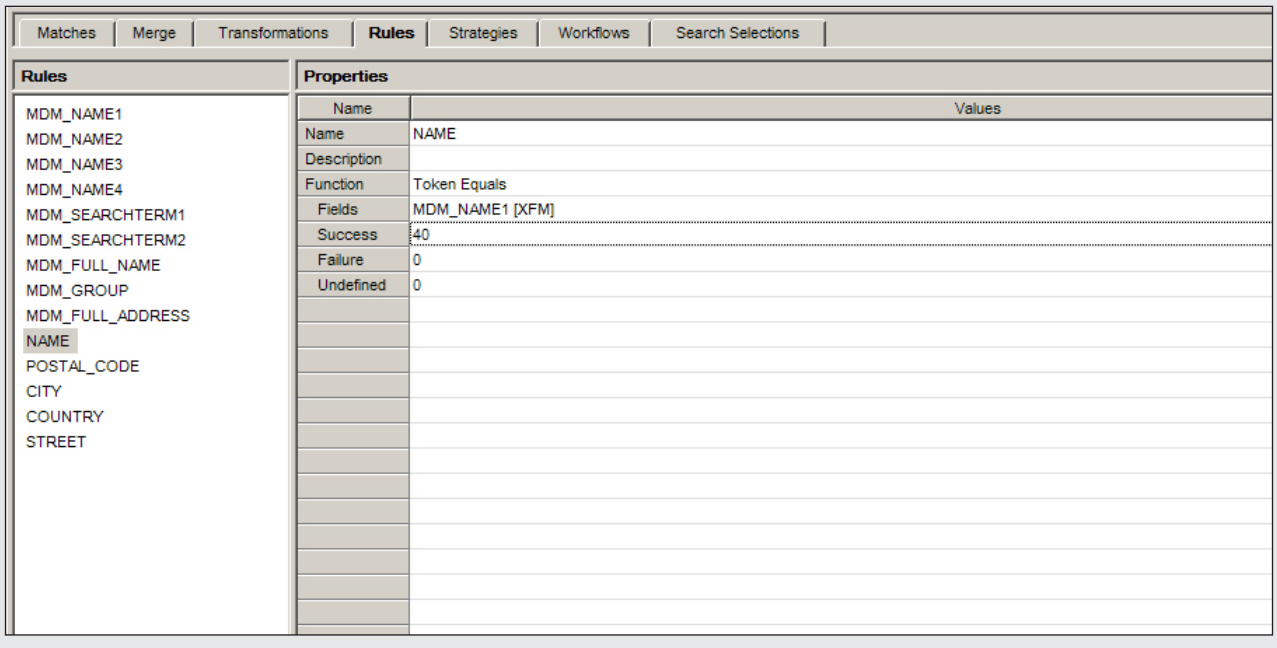


Figure 23 Rules screen in the MDM Data Manager

Token Equals function needs to tokenize the NAME string before it performs the comparison.

When comparing the NAME fields, the rule will receive a score based on whether the match is successful. Fuzzy logic is used to determine the success of a match. The possible outcomes for a function match per record are:

- Success
- Failure
- Undefined

Partial success scores are given using fuzzy logic, depending on how close the match is. A matching function score is equal to "Success" (a user-specified value, 10, 20, etc.) multiplied by a number of unique matching tokens and then divided by the total number of unique tokens.

The scores from the rules, in this case the NAME rule, are then rolled into strategies, which are defined as a group of rules (**Figure 21**). Each strategy contains a group of threshold scores. Each record is evaluated based on the cumulative score of all the rule scores given to it. Those scores roll into a rule, then all rules of the strategy being evaluated will roll into the strategy that contains the threshold scores. All records are then categorized into a match level based on these scores. A match level specifies the likelihood of a match. There are two possible match levels: High or Low.

These results will populate the Matches tab in merging and matching mode to help the user determine whether matches exist. This gives the users valuable information regarding the probability of a match, helping them tremendously.

After the strategies have been applied to the dataset being integrated with the existing MDM repositories, the user must evaluate which records to accept as matches and which records are in fact new. This process takes place within matching mode.

Once all the data has been imported, merged, and matched, consolidation is complete.

2. Harmonize

The next step in the process is to harmonize the data. Harmonization takes place after the initial data consolidation. The harmonization process is used to "push" out the new consolidation information to consumer systems. The system being consolidated was most likely feeding master data to additional systems. These other systems now begin to consume information from the consolidated MDM system. This newly consolidated data is distributed via the SAP NetWeaver MDM Syndicator.

The MDM Syndicator gives you the power of SAP NetWeaver MDM's parametric search to configure a subset of the repository and broadcast that dataset to any number of systems.

3. Centralize

The last step is to cut off the feeds to the existing MDM repositories that were set up during the consolidation phase. These systems have now been completely consolidated with SAP NetWeaver MDM. All new master data will be centrally created, allowing control over what new master data values are created. The corporate master data is in essence locked down and controlled. You can then broadcast the master data created within your system landscape to any and all consumer systems.

You have now unified your data. Here are the steps you followed:

1. You consolidated the master data from the newly acquired company into the existing MDM repositories via the Import Manager.
2. You applied master data strategies to the newly imported data to find potential matches between existing data and the data being consolidated.
3. You turned off additional systems consuming information from the legacy data warehouse being consolidated.
4. You set up additional systems as consumers of the SAP NetWeaver MDM repositories that contained the newly consolidated information.

5. You shut off legacy data warehouse systems and will create all future master data centrally within the SAP NetWeaver MDM solution.

You can now have a single BEx Query report on both CompanyA and CompanyB data since the data is now consistent and unified (no duplicate master data) with a MultiProvider that brings this data together (as shown in **Figure 24**).

Since the creation of master data is now centralized, you should start looking at how to effectively monitor the integration between CompanyA and CompanyB. The next section previews some of the capabilities available in the SAP CPM Suite. This suite can help immensely in mergers and acquisitions.

Post-mergers integration monitoring

Now that your data is unified, consistent, and in one data warehouse, it's time to explore the SAP CPM Suite for ongoing performance management of your post-merger integration (PMI) data. Within SAP's CPM Suite, Strategy Management enables you to create a PMI dashboard that shows all your PMI key performance indicators (KPIs) to measure the success of your acquisition. Some examples of KPIs for the PMI monitoring are: total revenue, customer retention percentage, key people retention percentage, and so on. An example of a scorecard showing some of these metrics is shown in **Figure 25**.

Determining the success of an acquisition isn't immediate. Companies may track acquisitions anywhere from 12 to 36 months to check their success. The UDI scenario discussed here should help you to integrate your data more quickly and to monitor the data from the acquisition with greater accuracy.

SAP Strategy Management allows you to track and measure the success of acquisitions. UD Connect creates a strong foundation for managing and merging data from an acquired company, whereas, SAP Strategy Management allows you to derive insight and effectively track and monitor your acquisitions.

You can also use the whole SAP CPM Suite for your post-merger scenario, as outlined in **Figure 26**.

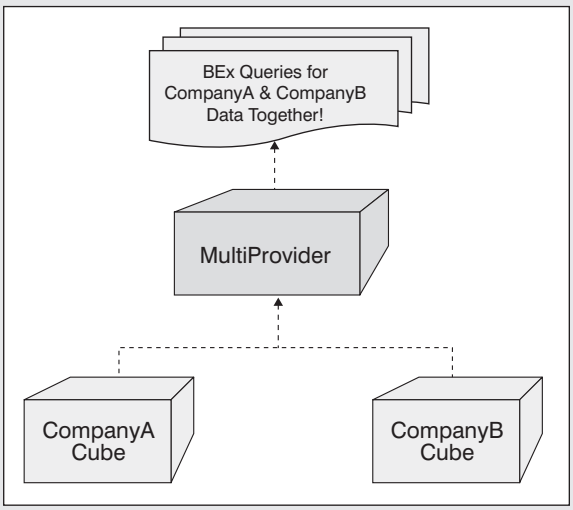


Figure 24 Merged data for CompanyA and CompanyB in a single BEx query

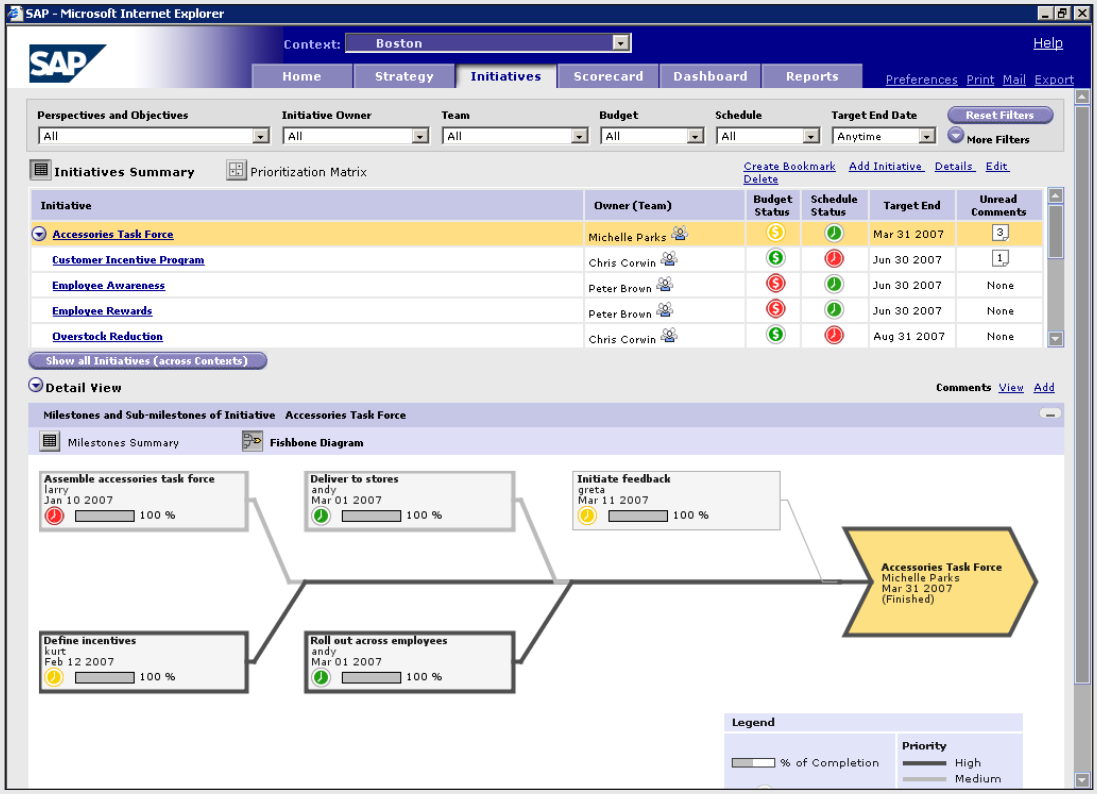


Figure 25 PMI scorecard using SAP Strategy Management

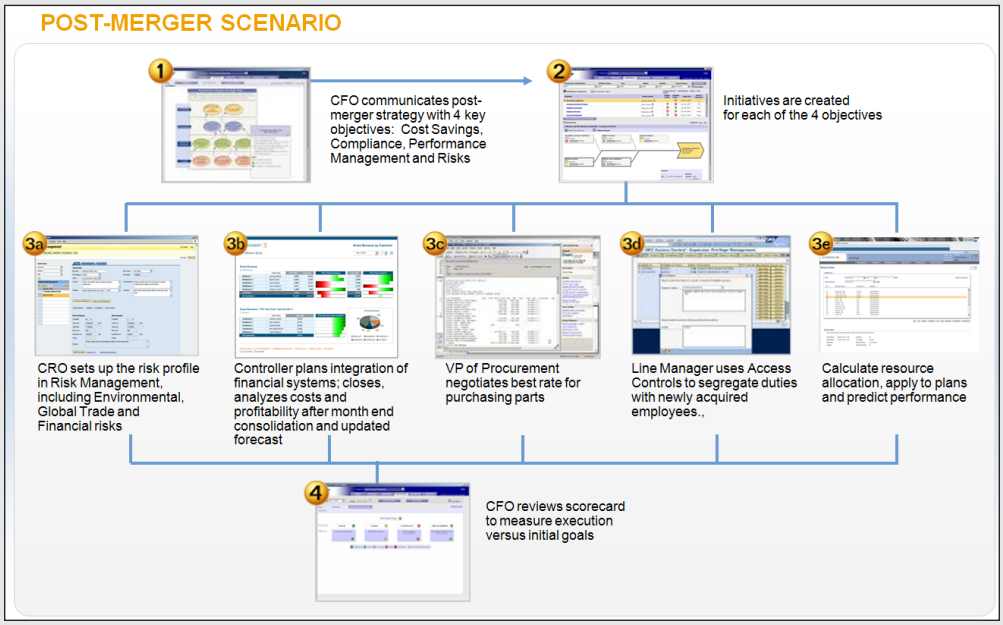


Figure 26 Post-merger scenario within SAP CPM Suite

Conclusion

UD Connect offers powerful functionalities that can aid in integrating data from multiple sources, both SAP and non-SAP. The mergers and acquisitions scenario we presented is just one example of the powerful uses of UD Connect. Using this phased approach, you can exploit the power of UD Connect to keep your business flexible. To use this data to drive insight, be sure to check out Strategy Management in the SAP CPM Suite.

Some best practices to keep in mind:

- Build BEx Queries on top of MultiProviders to help keep queries reusable despite underlying data model changes.
- Create a phased approach when working with third-party datasets, especially in mergers and acquisitions scenarios.
- Explore the SAP CPM Suite for further monitoring and KPIs.

Prakash Darji is currently a Product Manager in the Corporate Performance Management area within SAP Labs. He started his career in Operations Planning and Manufacturing and later moved to Plant Maintenance at Nalco Chemical Company. Prakash began working with SAP technology in 1999 and joined SAP in 2004 as a BI applications consultant. At SAP, Prakash has worked on numerous projects, including implementing key capabilities such as BI, Portals, SAP Analytics, Java applications, .NET connectivity, and SAP NetWeaver integration strategies. Following this, Prakash moved to the NetWeaver RIG as a specialist working on SAP NetWeaver Cross-Topics and integration, specifically in the areas of Visual Composer, Web Design, Portals, and BEx Analyzer. He then joined the SAP Corporate Strategy Group as a fellow where his primary responsibilities were in the area of mergers and acquisitions. His secondary responsibilities included work on SAP's Corporate Strategy as well as Product Roadmap. In his current role as a Product Manager, his responsibilities include working on the short- and long-term roadmap for SAP's Corporate Performance Management Suite. He is a recurring author for several publications and a speaker at SAP conferences around the world. You may contact him at prakash.darji@sap.com.

Scott Cairncross is a Platinum consultant with the SAP CPM Regional Implementation Group (RIG). Scott specializes in SAP NetWeaver BI and Business Planning. He has been working extensively with SAP NetWeaver BI analytics in the area of the SAP NetWeaver Visual Composer. Scott has also worked a great deal with SAP NetWeaver's Master Data Management solution. He has participated in multiple go-lives and has spoken at SAP TechEd, SAP Forum, ASUG, and the SAP NetWeaver/BI and Portals conference. Scott is also very strong in the area of custom development (ABAP). You may contact him at scott.cairncross@sap.com.

