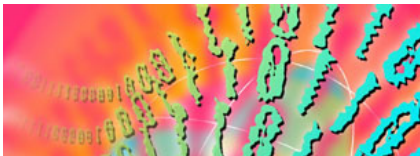


There are connectors and then there are connectors— and then there is so much more

Published: 22nd February 2011

In the database world there are essentially two types of connectors used by data integration and other suppliers: generic connectors such as ODBC and JDBC and native drivers that have specialised knowledge of the database technology to which they are connecting. The latter have the advantages that they are richer, have some understanding of the metadata used by the source/target and are typically more efficient and thus faster.



A similar principle applies to connecting to application software. Here the distinction between connectors that do and do not understand the source environment is much larger, especially for complex environments such as those provided by SAP or Oracle. This is why many vendors in the data integration and data quality spaces (Informatica and Trillium to take just two examples) have specialised connectors to interface with SAP ERP and similar environments.

However, the truth is that these connectors are still relatively simplistic. For example, suppose that you want to retrieve customer data from an ERP environment then you can use Informatica's connector to retrieve the relevant tables. That's all very well and good. However, what you will be presented with is a list of all the tables that relate to customers. Unfortunately there are over 1,000 of these: how do you decide which ones you want to access? The short answer is that you have to manually examine all the tables until you find the one or ones you want.

There is a further problem, which is that many connectors of this sort are based on templates that reflect the out-of-the-box product that SAP or Oracle will deliver

to you. Which is fine unless you customise the software. Which, of course, everybody does. Which means that you now have to customise (if you can) the connectors as well.

I have now met a company, Silwood Technology, which overcomes both of these issues (and a number of others). The way that the company's product (Saphir) works is that it reads the application's proprietary data dictionary (which will include any details of customisation), extracts that information into its own repository and then generates a complete data model of the current state of the application, based on the metadata it has retrieved. Having done that, you can filter, subset, search, slice and dice and report against the model. So, in the case of your 1,000+ customer tables you can ask the software (which is running against the Saphir repository and not the SAP or Oracle implementation) to list these tables in order of those with the most child relationships, for example. Logically, the customer master will have by far the most such relationships so it will be easy to identify.

So, first up, Saphir offers superior connectivity to SAP, Oracle, PeopleSoft, JDE and Siebel environments. But, of course, it is also doing a lot more than that. For example, suppose that you want to create a data mart: then you subset the data model that has been created for you, according to the data you are interested in, and there you have all the functions, relationships, tables, programs and so forth that you care about, ready for extraction. A similar consideration would apply to data migrations: either from one version of SAP to another or between SAP and Oracle or vice versa.

In addition, Saphir can be used to track the current state of the application and generate documentation for it, either directly or through export to a CSV file or

to any of the leading modelling tools including Erwin (CA, incidentally, resells Saphir), ER/Studio, PowerBuilder and so on. And, of course, it is complementary with all of these tools and with any other that requires a data model: enterprise architects should love Saphir, for example.

On a technical front there is one further point. You might ask if the product has to be updated every time that SAP or Oracle has a new release. Fear not, the short answer is no. This is because Saphir works at the level of the metamodel that underpins the application and this very rarely changes. Silwood tells me that since 2002, when the product was first introduced, there has only been a single change to the SAP metamodel and there have been none in any of the Oracle packages.

So, there you have it. This is seriously cool technology. Moreover, although you may not have heard of them, Silwood has a track record. It has been in business for a couple of decades, Saphir has been around for the best part of 10 years, the product has over 400 users (many leading companies) and it is resold not just by CA but by various other well-known suppliers. If you are a data modeller or an enterprise architect, or if you are building a data mart or involved in a data migration or archival project, or anything associated with these areas then you should take a serious look at Saphir.

Philip Howard
*Research Director - Data Management,
Bloor Research*



© Bloor Research 2011