R20
CONSULTANCY

Key Data Replication Criteria for Enabling Operational Reporting and Analytics

A Technical Whitepaper

Rick F. van der Lans
Independent Business Intelligence Analyst
R20/Consultancy

May 2013

Sponsored by
INFORMATICA®
The Data Integration Company™

Table of Contents

1 Management Summary

Operational business intelligence (OBI) is about supporting decision-making processes that require access to zero-latency data. Because these decisions are often urgent and time-sensitive, access to 'old' data is worthless. A key technology to implement operational BI is *data replication*. With this technology new data entered in production systems can be made available for reporting and analytics much quicker than with ETL. It allows decision makers to see what's happening right there and right then within the organization.

Organizations have to study whether they need data replication technology, and if so, which product to select. But what are critical criteria for selecting tools? What are, for example, important performance-related capabilities when used in a BI environment? Are transformation features important? To help out with evaluating data replication technology for developing OBI systems, this whitepaper lists and explains important criteria which are classified in five groups:

- Heterogeneous data source and data target access
- Non-intrusive data extraction
- Scalability and high-performance
- Easy to use and manage
- End-to-end data integration

In addition, this whitepaper describes which criteria are met by *Informatica Data Replication* (IDR).

2 Operational Business Intelligence: Easier Said Than Done

Business Intelligence and Car Windows and Mirrors – *Business intelligence* is the domain that focusses on supporting and improving the decision making processes of organizations. This is done by supplying the organization with *reporting* and *analytical* capabilities. Reporting is about presenting to users what has happened and analytics is about showing what may happen in the future.

A popular analogy used to illustrate the differences between reporting and analytics is driving a car. Reporting can be compared to looking through a car's rearview mirror, whereas analytics can be associated with looking through the front window. Reporting allows you to see what has happened and analytics helps you to predict what may happen.

Unfortunately, the analogy is only partially correct. If you look through the rearview mirror of your car, you see what's happening right at that moment. Unfortunately, this is not true for most reports. Most of them show what has happened a few hours ago, the previous day, or maybe even what has happened the previous month. In a way, the same is true for analytics, it should show what may happen. If you look through your car's front window, you see what's happening in front of you right then and there. You won't see what will happen after you have rounded the corner.

Bottom line: Looking through car windows and mirrors shows what's happening right there and then. This is difficult to relate to classic reporting and analytics. This is the domain of operational BI!

What is Operational BI? – In a nutshell, operational BI is about supporting decision-making processes with reporting and analytics that operate on *zero-latency data* (or *operational data*). Popular synonyms for OBI are *operational intelligence*, *real-time analytics*, and *real-time business intelligence*. Whatever name is used, as Wikipedia indicates: "The purpose of OBI is to monitor business activities and identify and detect situations relating to inefficiencies, opportunities, and threats and provide operational solutions."

In many organizations, OBI is already being used. For example, for a long time now credit card companies try to detect fraudulent payments immediately after they are made. The sooner a false payment is detected, the less the impact on the customer and the company itself. Another example is traffic control at airports. Traffic controllers, sitting in their high towers and managing incoming and outgoing airplanes, can only operate successfully when they are supplied with zero-latency data. Showing on their monitors where flights were 30 seconds ago, would not be very helpful.

To return to the car analogy. As drivers of cars, we are users of OBI. For example, the fuel gauge shows how much fuel is left in the tank. It would not be useful if it indicates how much fuel was available 30 minutes ago. The same applies to the speedometer. If a speedometer shows what our speed was 1 minute ago, we could end up with a mountain of speeding tickets. When you drive a car, all your instruments should show the situation as is.

The Aberdeen Group[1] identifies the following six "flavors" of OBI:

- transactional BI with analysis and reporting
- real-time analytics with business rules applied to data as it is captured
- near real-time analytics business rules
- operational reporting
- business activity monitoring or business process monitoring
- decision management based on business rules with integrated reporting and analytic applications

The Business Value of Operational BI – Many business processes and user groups can benefit dramatically from zero-latency data that shows what's happening right there and then. Operational management, the workforce, and external business parties, such as customers and suppliers, are all examples of user groups that can benefit from OBI. Organizations usually identify the following benefits for implementing operational BI: improved efficiency of individual business processes and the organization as a whole, reduction of operating costs, and increased customer satisfaction.

[1] The Aberdeen Group, *Operational BI: Getting Real Time About Performance*, December 2007, see https://www-304.ibm.com/jct03004c/tools/cpeportal/fileserve/download0/140686/AberdeenOperationBIperf.pdf?contentid=140686

For example, in a credit card company the costs of "fixing" fraudulent payments the minute it happens, is factors less than when those false payments are discovered a day later. The thief may have already made hundreds of payments. Another example is when visitors are browsing a website. Based on the style of browsing, certain advertisement should be presented. Again, which advertisement to show, depends on what the visitor is doing at that particular moment.

Operational BI: Easier Said Than Done – Most current business intelligence systems have an architecture consisting of a network of databases; see Figure 1. In this architecture, data is entered in the operational databases on the left-hand side of the diagram. ETL solutions are used to copy data from these operational databases to a staging area. Next, data is copied to a data warehouse. It's very likely that ETL technology is used to transform, standardize, integrate, and cleanse data during this process. Then, the data is copied to one or more data marts. In most of these data marts, data is organized in tables with a star schema or snowflake schema. Finally, data is copied to personal data stores (not in all cases). With every step, the form and quality level of the data is more in line with the requirements of the reporting and analytical tools.

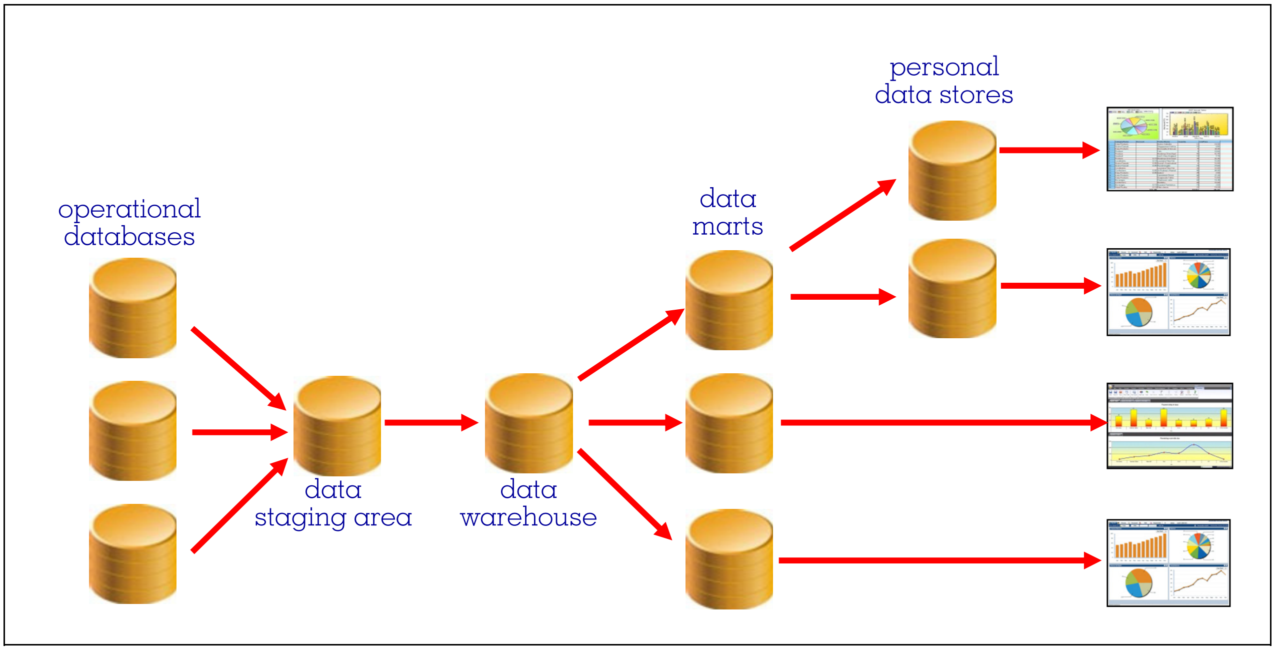


Figure 1 *Most business intelligence systems form a network of databases in which ETL processes are responsible for copying data between data stores.*

This step-wise approach takes time. It takes time before data, entered in an operational database, arrives in a data mart or personal data store and becomes available for reporting and analytics. Clearly, if users need access to zero-latency data, that data has to be moved through the network of databases in microseconds, which is a technological challenge for the following two technical reasons:

- First, to extract the new and changed data from a source system, an ETL tool requests the database server to retrieve that data from the database. The effect is that the database server has to spend time and resources on these requests, database files are accessed, and it creates extra workload on the operational system. The performance experienced by the production systems may go down temporarily, performance instabilities may arise, and so on. For this reason, in many organizations ETL tools are only allowed to retrieve data when the workload on the operational systems is minimal or when no users are active at all. Basically what this means is that data is extracted once per 24 hours: at night. Some organizations decide to postpone the ETL work until the weekend. The consequence is that data accessed by the reports have a latency of at minimum a day or week. All this is not in the interest of the OBI users.

- Second, to avoid the interference on operational systems as described above, ETL processes are scheduled. Periodically data is extracted from the source system and pumped bulk wise to a target system. So, there is no continuous copying of new data.

To summarize, implementing OBI in classic BI systems where scheduled ETL processes copy data from one database to another, is a challenge. A technology is required that is capable of copying new data right when it has been inserted, deleted, or updated, and that does not cause any interference on the operational systems. This is where data replication comes in.

3 The Need for Data Replication

Data Replication in a Nutshell – *Data replication* is an alternative technology for copying data from source to target systems. The main difference with a more classic ETL solution is that with data replication new data is continuously copied to the target system. When a new record is inserted, or an old one deleted or updated, that change is copied to the target system right away. So, there is no scheduling and no or a minimal delay. In microseconds, new data can become available in the target system.

To minimize interference on the operational systems, most data replication tools do not access the database files or request the database server for data, instead they retrieve (changed) data from the so-called log files. Right before database changes have been committed, they're also written by the database server to these log files. In other words, these log files contain all the data changes. Because data replication tools read data from these log files, there is almost no interference on the operational systems.

Data Replication in Business Intelligence Architectures – When data replication is deployed in classic BI systems, it can be used to make new data available more quickly making it possible to deliver zero-latency data and thus creating a foundation for OBI. Figure 2 shows an example of how data replication can be embedded in a BI system. In this example, data replication is used to copy changed data to a replicated database as quickly as possible. This replicated database can be used by reports that need zero-latency data. In addition, the data can also be copied from the replicated database to the staging area and the data warehouse to support the non-OBI reports.

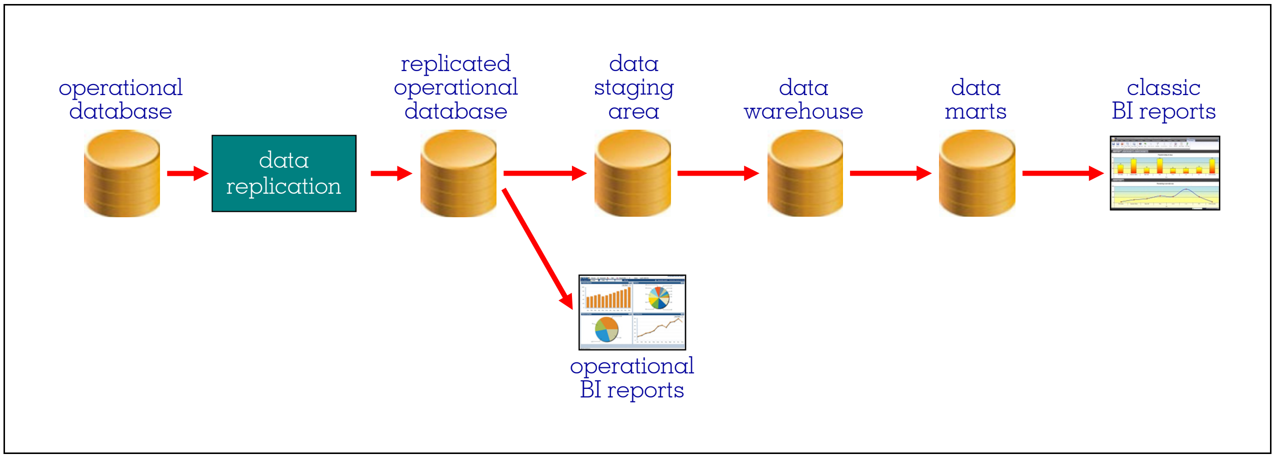


Figure 2 *Data replication can be used to make zero-latency data available in a BI environment.*

Evidently, this is only one way of embedding data replication in a BI system. An alternative would be to merge the replicated operational database and the staging area, or to replicate data straight into the data warehouse. All these alternatives have their advantages and disadvantages, and they all need data replication as an enabler.

For a more in-depth explanation of data replication, we refer to the whitepaper "Data Replication for Enabling Operational BI".[2]

In a nutshell, the *business value* of deploying data replication is to let business users operate with and to let them make decisions on zero-latency data. Replication makes it possible for BI to follow the speed of today's businesses. The business will not be constrained anymore because only yesterday's data can be obtained. Users can monitor and analyze business processes live.

Data Replication Criteria – Data replication tools have been around since the 1990s. Until recently they were primarily used in operational environments to increase availability or scalability. Now, with the increasing need for OBI, data replication is entering a new application domain.

In general, when a technology is used in a new application domain, some of the traditional capabilities are still needed, but other and new ones can become more important. To help out with evaluating data replication technology for developing OBI systems, the following sections list and explain important criteria which are classified in five groups

- Heterogeneous data source and data target access
- Non-intrusive data extraction
- Scalability and high-performance
- Easy to use and manage
- End-to-end data integration

[2] Rick F. van der Lans, *Data Replication for Enabling Operational BI*, June 2012, see http://vip.informatica.com/content/Downloads?docid=2086NA&lsc=NA-Ongoing-2012Q3-RH-IDR-Data_Replication_for_Enabling_Operational_BI_WP_www

In addition, these sections describe which criteria are met by *Informatica Data Replication* (IDR).

4 Heterogeneous Data Source and Data Target Access

It's very common that enterprise data is stored in a heterogeneous set of databases. For example, different database servers may be in use for different operational systems, and for data warehouses and data marts also other database servers may have been selected. Therefore, it's important that data replication technology supports a large set of database servers as source and as target. In addition, the extraction, transportation, and loading of data should be as efficient as possible. The reason is straightforward: data replication is deployed when time matters, when it's crucial that data is made available to reports as soon as possible. It's all about zero-latency data.

When evaluating a data replication technology to ensure Heterogeneous Data Source and Data Target Access, the criteria highlighted below should be considered.

Criteria 1: What is the list of data sources from which the data replication technology can extract data?

Especially in the world of business intelligence, more and more different database servers are being deployed. Therefore, it's important that the list of supported database servers by the data replication technology that can act as source is extensive.

IDR can extract data from the following SQL database servers:

- Oracle
- IBM DB2 for Linux, Unix, and Windows
- Microsoft SQL Server
- Sybase

Criteria 2: What is the list of data sources the data replication technology can load data into?

What applies to the source systems applies to the target systems as well: it's important that the list of supported database servers by the data replication technology that can act as target is extensive.

IDR can load data into the following SQL database servers:

- Oracle
- Microsoft SQL Server
- MySQL
- IBM DB2 for Linux, Unix, and Windows
- Sybase
- Teradata
- IBM/Netezza
- HP/Vertica
- EMC/Greenplum
- PostgreSQL

In addition, IDR can efficiently copy data from the data sources (mentioned above) via intermediate files to Hadoop. This can be useful for various reasons. The first reason is for copying structured data from production systems to an Hadoop environment. This enriches the data with more context. For example, if sensor data is stored in Hadoop, it's probably highly coded data. By adding more contextual data to those codes, the data becomes more valuable for reporting and analytics. Another reason is when large amounts of data have to be processed before they are stored in a classic database. In this case they can first be loaded into Hadoop (IDR uses intermediate files), then processed, and finally copied from Hadoop to a SQL database server.

Also, IDR can send data to message queue technology. The message bus can then pass that data to any type of application interested in this data. In a way, by combining IDR with message queues, data is pushed from the source system to the target system. For example, operational dashboards can really benefit from this.

Criteria 3: Besides replication of data, can the data replication technology also replicate database schema's and database schema changes?

IDR does notice when the schema of the database to be replicated has changed. Examples of database changes: new columns, dropped columns, and data types changes. Such changes have to be replicated as well, to keep the replicated database in synch. IDR can also replicate schemas in an heterogeneous environment.

Criteria 4: Is the data replication technology where possible *database agnostic* and are *cross-platform mappings* supported?

Developers use so-called *mappings* to indicate how data from a source system should be replicated to a target system. Some of those specifications are product-specific, such as the specifications that link the data replication technology to a source or target system. The rest of the mapping specifications, such as which source column should be mapped to which target column, which columns to skip, which records to filter, and so on, should be as database agnostic as possible. In other words, the mappings should be *cross-platform*. This allows for cross-platform development—the source and target systems don't have to be identical.

IDR can be classified as database agnostic and supports cross-platform mappings. Most of the specifications to be entered are not dependent on the source or target system used.

5 Non-Intrusive Data Extraction

Technologically, there are different styles to extract data from source systems. Some of these styles require substantial processing by the source systems and some require almost no processing. In BI environments, most of the source systems are production systems. Because data replication continuously extracts data, it's important that the replication process doesn't interfere with the production processing itself. In other words, the replication process should be as *non-intrusive* as possible.

Criteria 1: Is log-based CDC used by the data replication technology for *non-intrusive extraction* of data from the data sources mentioned in criteria 1?

To be as non-intrusive as possible, IDR uses log-based CDC for all four SQL database servers: Oracle, DB2 LUW, Microsoft SQL Server, and Sybase. For Oracle it extracts transactional changes from the online redo logs and/or the archive logs, for Microsoft SQL Server changes are extracted from the online or backup log files, for DB2 LUW the changes come from the UDB database log files, and for Sybase changes are extracted from the transaction log files. In all cases, the extraction process is non-intrusive and does not interfere with the production systems.

Criteria 2: Does the component of the data replication technology responsible for extracting data from the log files do most of its processing on the same machine on which the source system itself runs, or can some or all of the extraction work be offloaded to another machine?

IDR offers two styles of log-based CDC extraction. First, the *Extractor* component of IDR (responsible for capturing and extracting data) transfers the changed, transactional data to intermediate files. Next, these files are shipped over to the server on which the target system resides or to any other system. In this solution, the extraction process runs on the target system or a third system dedicated for this purpose.

Second, IDR also supports an option (called the Off-box option) where the archive logs from the source system are shipped over to another machine where it's processed by IDR. With this option, there is almost no work done on the source system side. The level of interference on the production systems is therefore very close to zero.

6 Scalability and High-Performance

Operational business intelligence requites zero-latency data. This implies that it is important that data is extracted, transformed, transported, and loaded fast. Performance is key here! In addition, the entire process should be robust. Decisions are made on the data in the reports. Serious business problems may occur if a user is not aware that the data is not correct.

Criteria 1: Which load mechanisms are used by the data replication technology to load new data efficiently in the data sources?

IDR provides three different forms of loading data:

- With *SQL Apply* SQL statements are applied to the target system that correspond with the SQL statements executed by the source system. For example, if a transaction on the source system consisted of some INSERT, UPDATE, and DELETE statements, then a set of INSERTs, UPDATEs, and DELETEs are executed on the target system that have the same effect on the data (this may not be the same set of SQL statements). To improve the load performance, and if possible, multiple transactions coming from the source system are bundled into one transaction on the target system.

- With *Merge Apply* data coming from the source system is first copied into staging tables in the target system. Periodically, the data in these staging tables is copied into the real target tables using a bulk load approach. This is the preferred approach when data is loaded into data warehouse appliances, such as EMC/Greenplum, HP/Vertica, IBM/Netezza, and Teradata. The bulk load approach makes it possible to deploy the native loading utility of these systems. This form of loading is much more efficient than using individual SQL statements.

- With *Audit Apply* each, each source table has a corresponding audit log table in the target system. These audit log tables record a row for every SQL update, insert, or delete on the source table. Each row contains a before- and an after-image of the data for each column in the table and for the columns that IDR adds to store metadata about the operation. This style of loading is very useful for auditors. It gives them full insight in what has been replicated and when.

Criteria 2: For which target systems does the data replication technology use a high-end load utility or feature of the target system?

As indicated in Criteria 1, IDR has a loading option called Merge Apply that optimizes loading data into data warehouse appliances and analytical database servers by using bulk loads. For example, for Teradata the utility called TPT (Teradata Parallel Transporter) is used.

Most of the data warehouse appliances allow for parallel loading of data. IDR provides the ability to configure the number of parallel threads during the loading process. This allows for parallel loading of data if the target system supports it and thus reducing the elapsed time necessary for loading data; see Figure 3.

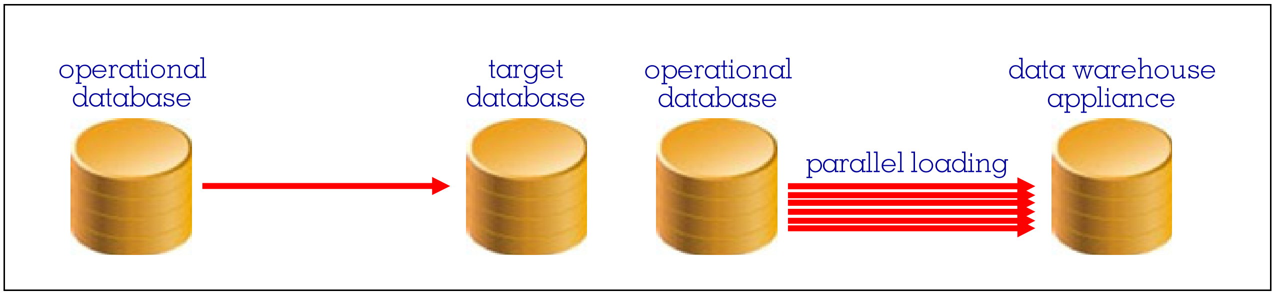


Figure 3 *IDR supports loading of data into data warehouse appliances and analytical database by deploying the parallel loading features servers of those products.*

Criteria 3: The internal architecture of data replication technology is crucial for the potential scalability, performance, and robustness. What does that the internal architecture look like?

Figure 4 shows the high-level architecture of IDR. IDR ships the files with data changes before everything has been committed. In other words, the changes are streamed to the target system on a continuous basis regardless of whether the transaction has been committed on the source. It's the Applier component that determines whether changes have really been committed. If not, they are not processed. This approach is sometimes referred to as *optimistic replication*. The advantage of this approach is that no additional memory or disc space is utilized to store the intermediate files on the source. In addition, since the intermediate files are continuously streamed, any latency associated with moving the intermediate files over to the target is eliminated.

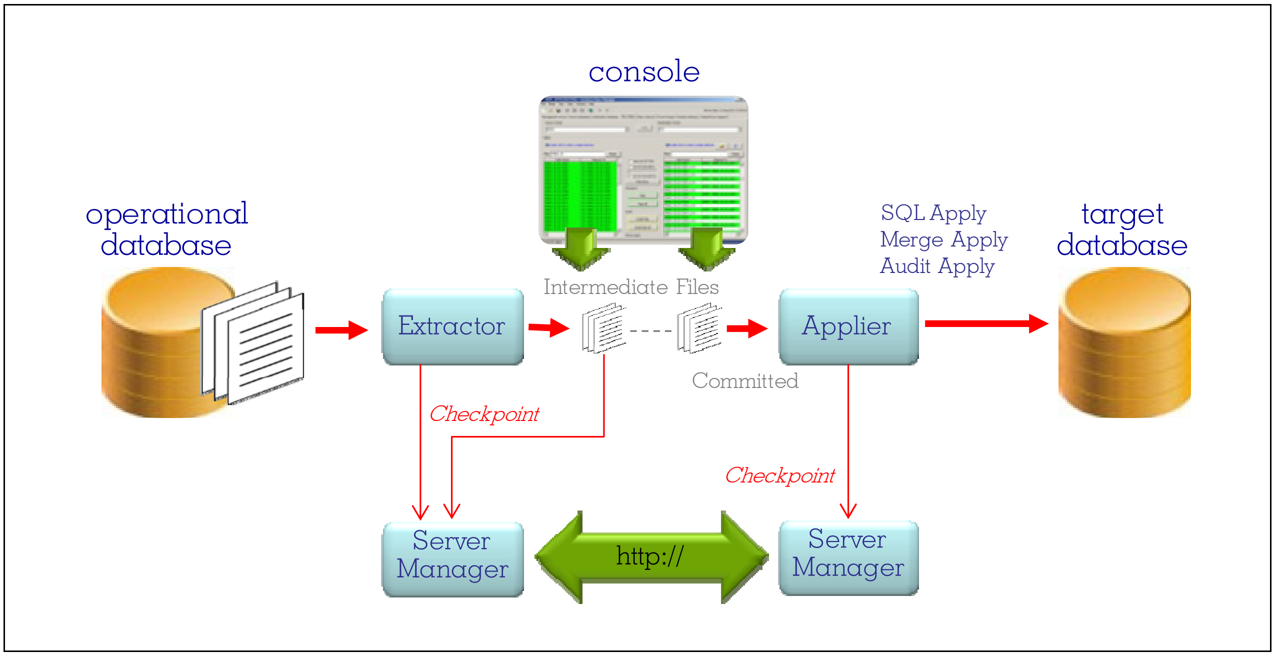


Figure 4 *High-level architecture of IDR.*

Criteria 4: How extensive are the data transformation capabilities of the data replication technology? For example, can new fields be added, can fields be dropped, can multiple fields be concatenated, can string, mathematical, and date-time operations be applied?

IDR supports basic transformations. For example, string and mathematical manipulations can be applied to columns, columns can be concatenated, virtual columns can be added, and existing ones can be skipped. IDR can also use any built-in database function to transform data as it is applied to the target system.

Criteria 5: When data is transported from a source to a target system, can it be compressed by the data replication technology?

The Extractor component of IDR stages all data changes in compressed files, reducing network overhead.

Criteria 6: When data is transported from a source to a target system, can it be encrypted by the data replication technology?

In IDR the intermediate files can be encrypted while in transit from the source to the target system.

Criteria 7: Can the data replication technology join data from multiple sources/tables into one table in the target system?

In data warehouse environments it's not uncommon that data from different systems has to be merged together in one table. For example, customer records from two production systems have to be brought together in one customer table in, for example, a data warehouse. In relational terminology, this is a union of rows.

In IDR, for consolidation purposes, data can be extracted from multiple data sources and loaded into one and the same table. For example, if multiple production systems contain customer data, and there is a need to create one table that shows the consolidation of all that customer data in a real-time fashion, this feature is strongly needed.

Criteria 8: What happens when a target system becomes unavailable?

To recover from the situation when a target system becomes unavailable, IDR provides a *checkpoint mechanism*. If, for any reason, the target system becomes unavailable, the Apply process stops. However, the checkpoint mechanism marks the exact recovery point. The other components continue to function unabated. Once the target system is back up again, the Apply process continues seamlessly, picking up from where it left off.

7 Easy to Use and Manage

When data replication technology is used to support business intelligence, it's very likely that specialists from the BI department or BI competence center (BICC) have to work with it. Because BI specialists don't always have detailed technical knowledge of the operational systems, and are not always skilled in the technologies deployed in operational systems, it's important that the data replication technology is easy to use and manage.

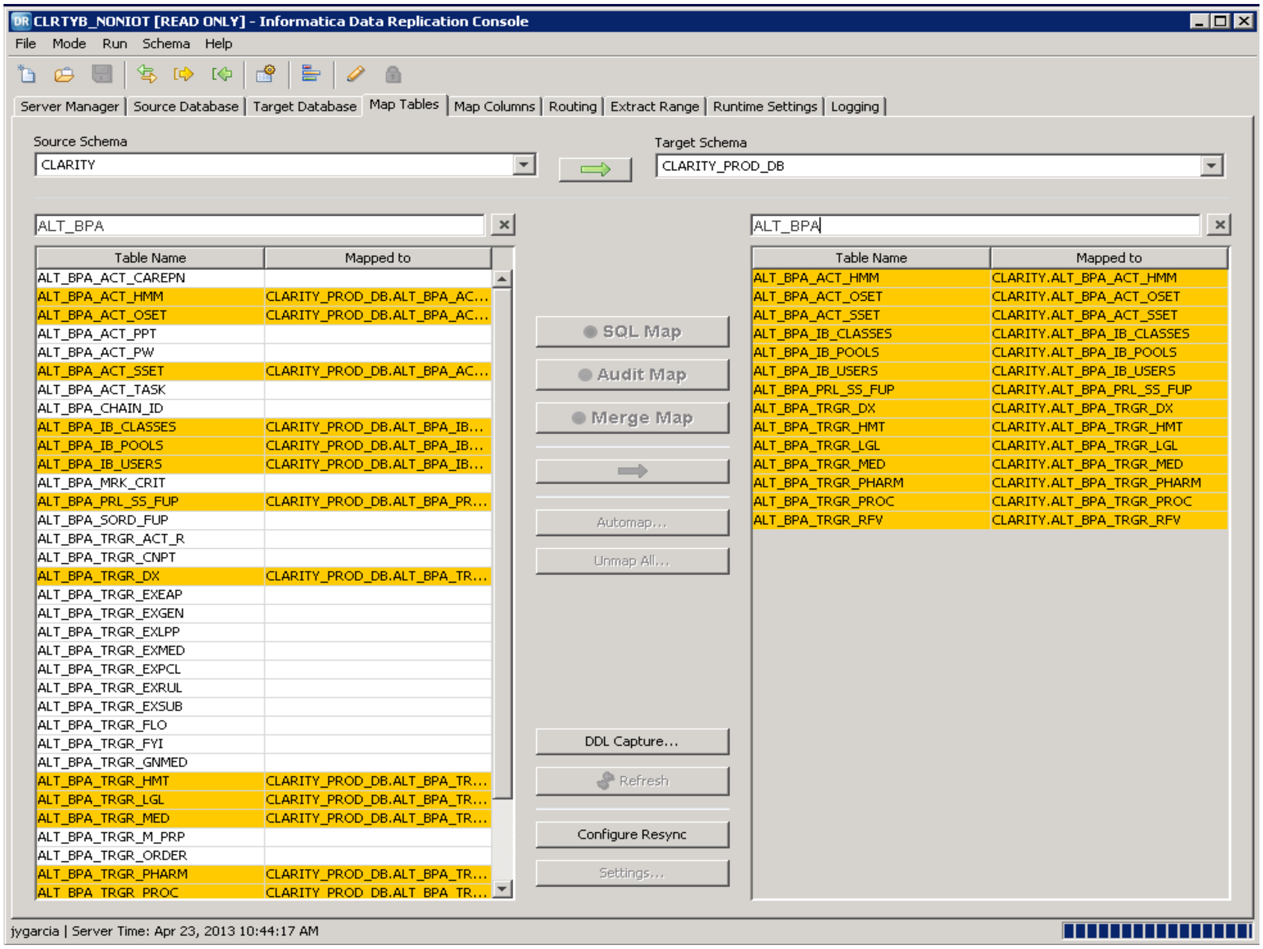
Criteria 1: Does the data replication technology support features for initially loading the target system with data that's already in the source system?

IDR comes with a module called InitialSync for initial synchronization of source and target systems.

In addition, Informatica offers a separate product called *Informatica Fast Clone* which performs high speed synchronization of databases. Even when the source database is in use, the entire database can be cloned. Fast Clone is not accessing the database server itself, but is accessing the database files directly. Currently, Fast Clone only supports Oracle databases as source. It supports several targets.

Criteria 2: Does the data replication technology support a graphical, user-friendly interface for entering and maintaining extraction, transformation, transportation, and loading specifications? Or do the developers need to use a technical scripting language?

IDR offers a graphical, user-friendly user-interface for configuration, management, and administration. This eliminates the need for manual scripting. Figure 5 shows an example of the interface of IDR.



Source Schema: CLARITY

Target Schema: CLARITY_PROD_DB

ALT_BPA

Table Name	Mapped to
ALT_BPA_ACT_CAREPN	
ALT_BPA_ACT_HMM	CLARITY_PROD_DB.ALT_BPA_AC...
ALT_BPA_ACT_OSET	CLARITY_PROD_DB.ALT_BPA_AC...
ALT_BPA_ACT_PPT	
ALT_BPA_ACT_PW	
ALT_BPA_ACT_SSET	CLARITY_PROD_DB.ALT_BPA_AC...
ALT_BPA_ACT_TASK	
ALT_BPA_CHAIN_ID	
ALT_BPA_IB_CLASSES	CLARITY_PROD_DB.ALT_BPA_IB...
ALT_BPA_IB_POOLS	CLARITY_PROD_DB.ALT_BPA_IB...
ALT_BPA_IB_USERS	CLARITY_PROD_DB.ALT_BPA_IB...
ALT_BPA_MRK_CRIT	
ALT_BPA_PRL_SS_FUP	CLARITY_PROD_DB.ALT_BPA_PR...
ALT_BPA_SORD_FUP	
ALT_BPA_TRGR_ACT_R	
ALT_BPA_TRGR_CNPT	
ALT_BPA_TRGR_DX	CLARITY_PROD_DB.ALT_BPA_TR...
ALT_BPA_TRGR_EXEAP	
ALT_BPA_TRGR_EXGEN	
ALT_BPA_TRGR_EXLPP	
ALT_BPA_TRGR_EXMED	
ALT_BPA_TRGR_EXPCL	
ALT_BPA_TRGR_EXRUL	
ALT_BPA_TRGR_EXSUB	
ALT_BPA_TRGR_FLO	
ALT_BPA_TRGR_FYI	
ALT_BPA_TRGR_GNMED	
ALT_BPA_TRGR_HMT	CLARITY_PROD_DB.ALT_BPA_TR...
ALT_BPA_TRGR_LGL	CLARITY_PROD_DB.ALT_BPA_TR...
ALT_BPA_TRGR_MED	CLARITY_PROD_DB.ALT_BPA_TR...
ALT_BPA_TRGR_M_PRP	
ALT_BPA_TRGR_ORDER	
ALT_BPA_TRGR_PHARM	CLARITY_PROD_DB.ALT_BPA_TR...
ALT_BPA_TRGR_PROC	CLARITY_PROD_DB.ALT_BPA_TR...

ALT_BPA

Table Name	Mapped to
ALT_BPA_ACT_HMM	CLARITY.ALT_BPA_ACT_HMM
ALT_BPA_ACT_OSET	CLARITY.ALT_BPA_ACT_OSET
ALT_BPA_ACT_SSET	CLARITY.ALT_BPA_ACT_SSET
ALT_BPA_IB_CLASSES	CLARITY.ALT_BPA_IB_CLASSES
ALT_BPA_IB_POOLS	CLARITY.ALT_BPA_IB_POOLS
ALT_BPA_IB_USERS	CLARITY.ALT_BPA_IB_USERS
ALT_BPA_PRL_SS_FUP	CLARITY.ALT_BPA_PRL_SS_FUP
ALT_BPA_TRGR_DX	CLARITY.ALT_BPA_TRGR_DX
ALT_BPA_TRGR_HMT	CLARITY.ALT_BPA_TRGR_HMT
ALT_BPA_TRGR_LGL	CLARITY.ALT_BPA_TRGR_LGL
ALT_BPA_TRGR_MED	CLARITY.ALT_BPA_TRGR_MED
ALT_BPA_TRGR_PHARM	CLARITY.ALT_BPA_TRGR_PHARM
ALT_BPA_TRGR_PROC	CLARITY.ALT_BPA_TRGR_PROC
ALT_BPA_TRGR_RFV	CLARITY.ALT_BPA_TRGR_RFV

jygarcia | Server Time: Apr 23, 2013 10:44:17 AM

Figure 5 *Screenshot showing the graphical, user-friendly interface of IDR.*

Criteria 3: Can data replication specifications of source and target systems be re-used?

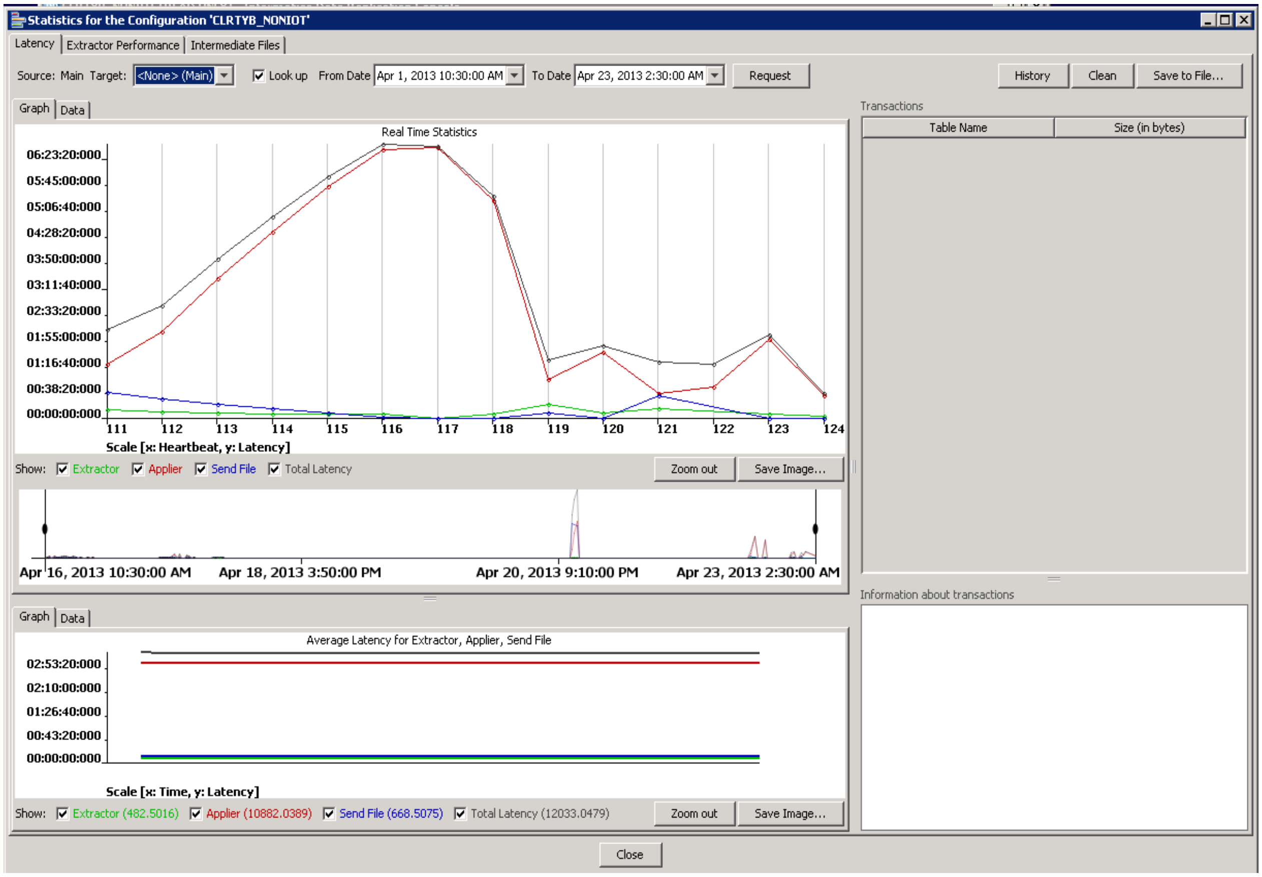
In IDR all these specifications can be re-used. There is no need to define a source system again, if the data in that system is used a second time to replicate to another target system.

Criteria 4: Does the data replication technology support a centralized monitor to show the entire data replication environment?

IDR offers a centralized GUI-based console. Figure 6 contains a screenshot of the monitor in action.

8 End-to-End Data Integration

The first sections show that data replication technology is an enabler for building OBI systems. Without it, having access to zero-latency data for reporting and analytics is hard to implement. But it's not the only technology needed to build OBI systems. Data replication is responsible for making new data entered in the operational systems available for reporting, analytics, and other tasks as quickly as possible. Next, data may have to be transformed, cleansed, and integrated before it can be used in an operational reporting environment. It may also be necessary that the data is processed by a *complex event processing* (CEP) engine, that applies rules to the data for making embedded business decisions.


Statistics for the Configuration 'CLRTYB_NONIOT'

Source: Main Target: <None> (Main) ☑ Look up From Date Apr 1, 2013 10:30:00 AM To Date Apr 23, 2013 2:30:00 AM

Show: ☑ Extractor (482.5016) ☑ Applier (10882.0389) ☑ Send File (668.5075) ☑ Total Latency (12033.0479)

Figure 6 *Screenshot showing the monitor of IDR.*

In other words, to implement operational analytics and reporting, data replication alone is not sufficient. Besides data replication, a data integration platform should also support ETL functionality for advanced batch-oriented transformation, integration, and cleansing functionality, it should support on-demand transformation, integration, and cleansing functionality (data virtualization), and it should offer CEP functionality to implement embedded BI. Together these functionalities form an end-to-end data integration platform.

Criteria 1: Is ETL functionality supported for advanced data transformation, integration, and cleansing functionality in a batch-oriented and scheduled style?

IDR compliments the Informatica Platform, which includes a popular and powerful ETL product called *PowerCenter*. IDR easily interoperates with PowerCenter. The latter can be used to transform, integrate, aggregate, clean the data to bring it the right form the BI reports need.

Criteria 2: Is data virtualization functionality supported for advanced data transformation, integration, and cleansing functionality in an on-demand style?

Informatica supports a data virtualization product, called *Informatica Data Services* which uses the same repository and development language as Informatica PowerCenter. Like IDR, IDS is a component of the Informatica Platform. These two tools can cooperate in different ways. For example, when IDR has pushed data to a replicated database, IDS can be used to transform, aggregate, and integrate the data in an on-demand style, delivering zero-latency data in the right form to the reports. Another example is, that when data is cached in IDS, that data can be replicated to other environments.

Criteria 3: Is complex event processing (CEP) functionality supported for embedded analytics?

The goal of CEP is to identify meaningful events (such as opportunities or threats) and respond to them instantaneously. With CEP, data is processed continuously. When data is processed, it's held against relevant rules. If a rule is true, an action is taken. Examples of such rules are: Is this the second payment with a particular card within 1 minute? Did the visitor jump from webpage X to Y directly? When data flows in and these rules are true, an application or user can make a decision on what to do what that? In case of the credit card payment, the second one may be a fraudulent one, and must be stopped.

Informatica's CEP product is called *RulePoint* with which CEP rules can be specified. IDR can act as a source that continuously streams data to the RulePoint CEP engine. Together, IDR and RulePoint form a foundation for developing embedded OBI systems.

About the Author Rick F. van der Lans

Rick F. van der Lans is an independent analyst, consultant, author, and lecturer specializing in data warehousing, business intelligence, service oriented architectures, and database technology. He works for R20/Consultancy (www.r20.nl), a consultancy company he founded in 1987.

Rick is chairman of the annual European Data Warehouse and Business Intelligence Conference (organized in London). He writes for the eminent B-eye-Network[3] and other websites. He introduced the business intelligence architecture called the *Data Delivery Platform* in 2009 in a number of articles[4] all published at BeyeNetwork.com.

He has written several books on SQL. His popular *Introduction to SQL*[5] was the first English book on the market in 1987 devoted entirely to SQL. After more than twenty years, this book is still being sold, and has been translated in several languages, including Chinese, German, and Italian.

For more information please visit www.r20.nl, or email to rick@r20.nl. You can also get in touch with him via LinkedIn (http://www.linkedin.com/pub/rick-van-der-lans/9/207/223) and via Twitter @Rick_vanderlans.

About Informatica Corporation

Informatica Corporation is the world's number one independent provider of data integration software. Organizations around the world gain a competitive advantage in today's global information economy with timely, relevant and trustworthy data for their top business imperatives. More than 4,100 enterprises worldwide rely on Informatica to access, integrate, and trust their information assets held in the traditional enterprise, off premise, and in the Cloud. For more information, call +1 650-385-5000 (1-800-653-3871 in the U.S.), or visit www.informatica.com. Connect with Informatica at http://www.facebook.com/InformaticaCorporation, http://www.linkedin.com/companies/3858, or follow on Twitter @InformaticaCorp.

[3] See http://www.b-eye-network.com/channels/5087/articles/

[4] See http://www.b-eye-network.com/channels/5087/view/12495

[5] R.F. van der Lans, *Introduction to SQL; Mastering the Relational Database Language*, fourth edition, Addison-Wesley, 2007.