

A. Tsyganov, S. Petit, A. Suwalska, CERN

# Oracle Text at the CERN Engineering and Equipment Data Management System search engine

CERN, the European Laboratory for Particle Physics, has recently started the Large Hadron Collider (LHC), a 27 km particle accelerator. The equipment data life-cycle management of this project is provided by the Engineering and Equipment Data Management System (EDMS [1] [2] [3]). Using an Oracle database as data repository, the service supports document and equipment data management throughout the entire life cycle of the LHC project: design, manufacturing, installation, commissioning and maintenance. The data collection phase, carried out by specialists, is now being replaced by a phase during which data will be consulted on an extensive basis by non-expert users.

The EDMS Portal (fig. 1) has been designed to address this new challenge.

Aimed at simplifying access to the information, one of its key functionalities is its search engine, based on Oracle Text technology [4]. Its design benefits from the experience and knowledge on previous search engines [5] [6].

This paper describes the architecture of the new search engine and the organisation of its data. It presents the structure of the Oracle Text index and the implementation of search requests. Some problems that have been faced during implementation of the search engine are also discussed.



Figure 1. The EDMS Portal

## Data Architecture and Index Structure

A starting point of the search engine is a definition of types of data on which searches are conducted. Key types of EDMS objects were selected: documents, equipment, projects etc... To build the Oracle Text search index, it is necessary to collect data into one storage place. All data is merged into a specially designed table for metadata (MTABLE). The primary key of this table consists of the object identifier and the object type. The data which must be indexed and on which searches must be performed are stored in an «XML like» string column. This string is divided into sections that match the objects' metadata (e.g.: «title», «author», «registration date»...). The sections are delimited by tags (<author>John Smith</author>). It is to be noted that every object type can have its own set of sections, depending on its own specific metadata. MTABLE is partitioned by object type.

From the many possibilities of available indices in the Oracle Text technology, the CONTEXT Index [6] was chosen, this is the standard index type for traditional full-text searches. This

index type allows an efficient access to large amounts of information. The important parameters for index creation are:

```
WILDCARD_MAXTERMS = 15000
PREFIX_INDEX = YES
SUBSTRING_INDEX = YES
```

One of the key points of the search is performance for the left-truncated (<%hello>), right-truncated (<hello%>) and double-truncated (<%hello %>) wildcard queries. Defining the SUBSTRING\_INDEX parameter is boosting query performance for double side truncated wildcard queries. Similar but less flexible behaviour is obtained by defining PREFIX\_INDEX parameter when only right-truncated wildcard queries are considered. Even though creating the substring index can be few times slower and significantly larger than prefix index, it seems to be a good compromise between initial investment and gains at the queries response time.

The following error: «Oracle Text error DRG-51030 (wildcard query expansion resulted in too many terms)» happens when the amount of records found is above a given value. By default, this value is 5000. The parameter WILDCARD\_MAXTERMS allows setting another threshold.

### A script for Oracle text index creation is listed below:

```
CREATE INDEX C_GMETADATA_INDX
ON METADATA_TABLE(c_text)
INDEXTYPE IS ctxsys.context
PARAMETERS(
  storage EDMS_ORACLETEXT_STORAGE_PREF
  stoplist EDMS_ORACLETEXT_STOPLIST_PREF
  lexer EDMS_ORACLETEXT_LEXER_PREF
  wordlist EDMS_ORACLETEXT_WORDLIST_PREF
  section group EDMS_ORACLETEXT_SECTION'
)
```

There are a few special configuration parameters to set for the index. These are set with specially created preferences (blocks of registered Oracle Text parameters)

- Storage – sets up storage parameters for the index.
- Stoplist – is a list of words which will not be indexed («and», «in», «an», etc...).
- Lexer – specifies the language of the text to be indexed (in our case we used BASIC\_LEXER) and how text will be tokenized for indexing, defines which characters will be marked as whitespaces etc...
- Wordlist – this parameter sets general rules for index creation (WILDCARD\_MAXTERMS, PREFIX\_INDEX etc...).
- Section group – this parameter defines the list of searchable sections, i.e. how the data describing the objects are structured; the XML tags mentioned above have a one to one mapping with the sections.

Even though the MTABLE table is organised with partitions, the search engine index is non-partitioned. Dynamically changing of a list of registered searchable sections on a partitioned index, requires the index to be recreated to make a search based on this section possible (see chapter 6 about limitations).

Every object type has its own searchable information. For instance for documents, it is: document number, title, authors, description, document version, document status etc. Here is a small example of an «XML like» string, which is stored in MTABLE table:

```
<ALL>
  <A>LHC Parameter and Layout Committee</A>
  <TKD>
  <T>Parameters for MCIBH</T>
  ...
  </TKD>
  <D_ID>104304</D_ID>
  <DOC_VERSION>0</DOC_VERSION>
  ...
  <STATUS_TYPE>Released</STATUS_TYPE>
  <DOC_TYPE_NAME>Engineering Parameters</DOC_TYPE_NAME>
  <CPD_ID>LHC-MCIBH-EP-0001</CPD_ID>
</ALL>
```

The XML string consists of tags on which Oracle Text can execute its search process. There is one global tag <ALL> for the entire structure, searching on this includes all information from the XML string.

## Data Control and Collection

In order to keep the MTABLE table consistent with the data it refers to, some triggers are used. These triggers call a common procedure, which populates the MTABLE table. This procedure is located in a dedicated PL/SQL package (fig. 2).

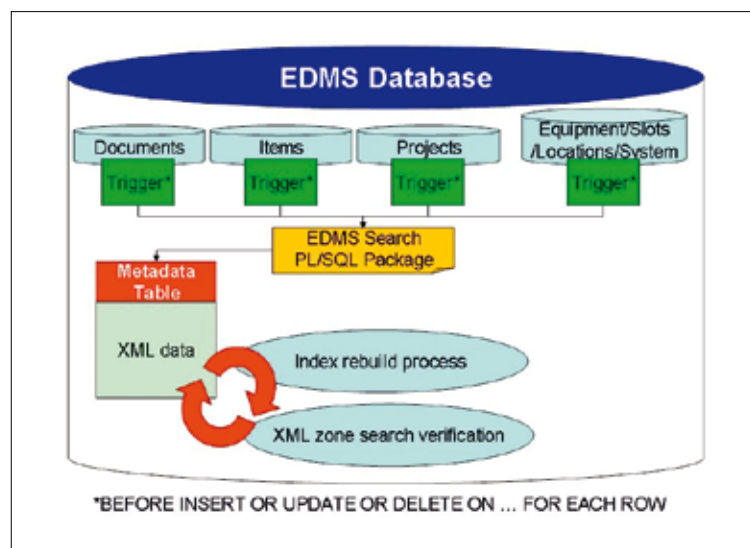


Figure 2. Data control and collection for the EDMS search engine

Triggers are configured to take effect before insert, update or delete (DML) operation on the source objects of EDMS. Therefore, a DML operation will force changes in the MTABLE and impact the Oracle Text index. It is known that modifying a record that has some references in an Oracle Text index has a side effect: it makes this record non-searchable from the index point of view. In order to cope with this issue, two batch processes are completing the triggers to enable the system to keep the index up to date. The first process «index rebuild process» (see fig. 2) rebuilds the Oracle Text index every 10 minutes. The second «XML zone search verification» (see fig. 2) process looks for any new searchable section and adds it automatically to the search list.

For operational reasons, the MTABLE is abstracted by a synonym. A temporary MTABLE can be created while maintenance operations are carried out on the real table one. The synonym allows a seamless switch during the time of the maintenance intervention.

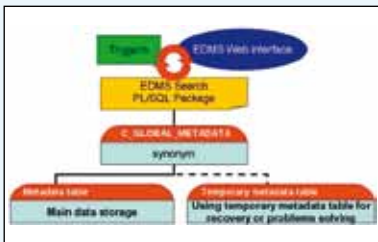


Figure 3. Tables architecture in EDMS Portal search

## Search Engine

To use the Oracle Text index feature it is necessary to call special SQL functions. For the CONTEXT index, the function to use is CONTAINS [7]. This function is used in the WHERE clause of a SELECT statement to force the use of an Oracle Text index. It also returns a relevance score for every row selected. All queries in the search engine are generated dynamically, and the result of their execution is a reference cursor, which is then used to fetch data as needed.

The main search request task for the EDMS Portal can be formulated as follows:

«Search for the given criteria on all types of objects. For every document, check if a particular user is allowed to see the document retrieved, and stop searching when 1000 such documents are found. For all other types of objects, find no more that 1000 rows (see MAX\_SEARCH\_RETURN on fig. 4). Display 200 rows for each type of object (see MAX\_ROWS\_RETURN of fig. 4). Displayed rows must be the most relevant ones». Figure 4 shows an SQL query, generated by the EDMS search engine, for this request.

```

WITH metadata_view AS (
SELECT t2.*
      ,ROW_NUMBER() OVER ( PARTITION BY obj_type ORDER BY hit_ratio DESC
                          ) AS group_rownum
FROM(
  SELECT
    SCORE(10) hit_ratio
  ,t1.*
  ,CASE WHEN obj_type <> ,D' THEN 1
        WHEN obj_type = ,D'
          AND EDBPKORACLETEXT.STF$GET_GARRAY_COUNT < MAX_SEARCH_RETURN THEN
            EDBPKORACLETEXT.STF$CHK_DISPLAYABLE_DOC(...)
          ELSE 0
        END disp
  FROM c_global_metadata t1
  WHERE CONTAINS (c_text, SEARCH_CRITERION, 10) > 0
) t2
WHERE disp > 0
)

SELECT *
FROM(
  SELECT t4.*
      ,ROW_NUMBER () OVER (PARTITION BY obj_type
                          ORDER BY global_hit_ratio DESC
                          ) AS display_group_rownum
  FROM(
    SELECT hit_ratio *
          EDBPKORACLETEXT.STF$GET_RANK_ADJUSTMENT(...) AS global_hit_ratio
    ,obj_type
    ,obj_id
    ,obj_link
    ,c_text
    ,CASE WHEN group_rownum + 1 = MAX_SEARCH_RETURN THEN
          EDBPKORACLETEXT.STF$PUT_VAR_TO_CONTEXT(...)
        ELSE NULL
        END AS obj_type_limit
    FROM( SELECT *
          FROM metadata_view
          WHERE 1 = 1 AND group_rownum < MAX_SEARCH_RETURN
        ) t3
    ) t4
  )
WHERE 1=1 AND display_group_rownum < MAX_ROWS_RETURN
ORDER BY obj_type, display_group_rownum

with
SEARCH_CRITERION : sys_context(,MetadataSearchContext', 'SearchParam')
MAX_SEARCH_RETURN : sys_context(,MetadataSearchContext', 'MAX_SEARCH_RETURN')
MAX_ROWS_RETURN : sys_context(,MetadataSearchContext', 'MAX_ROWS_RETURN')
  
```

Figure 4. EDMS portal search query using ROW\_NUMBER function

ROW\_NUMBER [8] is an Oracle analytic function that can be used to get the number of a row with respect to a group of rows. The Syntax is:

ROW\_NUMBER() OVER ( PARTITION BY obj\_type ORDER BY hit\_ratio DESC).

The ROW\_NUMBER function is used to obtain the number of a row in a group of rows based on its relevance. Then it is possible to cut result rows by their number in the group (for instance to get most 100 relevant from group(s) of objects).

The SQL query starts from the WITH block, where data is searched and rows are ranked for the given object type. The function STF\$CHK\_DISPLAYABLE\_DOC (fig. 4) is responsible for excluding documents which the user cannot see due to access restrictions. The main SELECT block uses a WITH block as a view and performs the next steps for obtaining the correct search results.

By default, relevance is calculated by Oracle using the SCORE function, which uses an inverse frequency algorithm based on Salton's formula [9]. The following example demonstrates this:

Searching for document number 1234567 returns two rows:

Obj_id	C_text
100	<ALL> <D_ID>1234567</D_ID> <D>test document</D> </ALL>
101	<ALL> <D_ID>9876543</D_ID> <D>test 1234567 and 1234567</D> </ALL>

Tag <D\_ID> stores document number

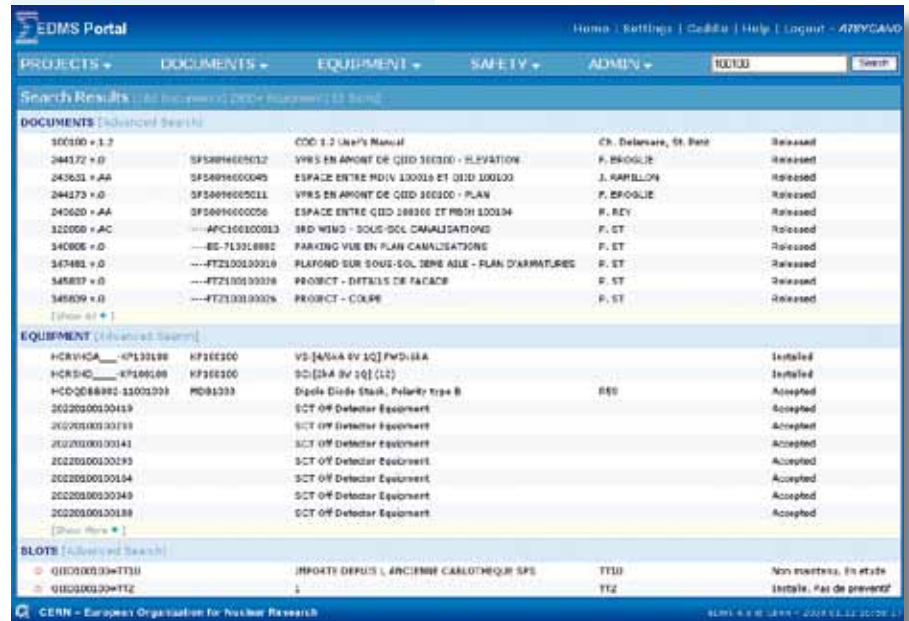
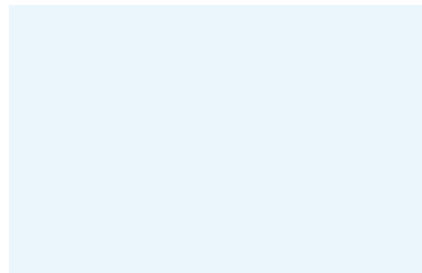


Figure 5. EDMS Portal search results display

The row with obj\_id = 101 will have a higher score, because it contains the string «1234567» twice. This is not a good result in our case. For the EDMS search, the document number is a very important criterion. If the search criterion is a document number, then the result rows for all documents with this number (it is possible that several versions of a document exist) must have the highest relevance result. So, to calculate the additional relevance score, the special function STF\$GET\_RANK\_ADJUSTMENT (fig. 4) applies the specific set of rules. The display of the search results on the EDMS Portal is shown on fig. 5.

Because of the complexity of the dynamically created SQL queries, straightforward use of bind variables is not efficient. To ensure the consistency of the SQL texts generated for similar requests, the search engine uses a specially created Oracle context. This context is a functionality provided by Oracle, which allows the storage of information during an Oracle session. All necessary parameters are kept in this context during SQL generation. The context parameters are unique for the given session and are maintained by the Oracle database. To obtain information from this context there is a special function sys\_context, provided by Oracle. The following is an example of a WHERE clause for the search criterion of an SQL statement:

```
WHERE CONTAINS (
    c_text
    , sys_context(, MetadataSearchContext', 'SearchParam')
    , 10
) > 0
```

Where «MetadataSearchContext» is a name of the context and «SearchParam» contains searched expression.

In addition, it is interesting to focus on the part of SQL where the row «obj\_type\_limit» is processed (fig. 4). This piece of code is used when there are more rows for a particular object type than the expected search limit indicates. An example of such a search result is shown in fig. 6.

DOCUMENTS					
10000	3.0		SYNCHRONIC STUDIES OF THE UHC 15 DIPOLE CHROMAT... AND A CORRELATION ...	Bazsin D., Calvez G.	Released
20079	3.0	PHYSICS	SUPPORT DIPOLE / DIPOLE SUPPORT	S. BAKULEV	Released
10704	3.0		BENDING TESTS ON 20 M AND 25 M UHC DIPOLE COOL PIPES	Rozema A. B. et al.	Released
10704	3.0	UHC00AA020	DIPOLE BUSBAR INSTALLATION - EXTERIOR DIPOLE BUSBAR	E. GEDOROV	Released
20498	3.0	UHC00A0106	FLEXIBLE BUSBAR DIPOLE DIPOLE BUSBAR FLEXIBLE	V. KLIZMENOV	Released
20498	3.0	UHC00A0103	SUPPORT LYRE DIPOLE SUPPORT FOR DIPOLE LYRE	V. KLIZMENOV	Released
20381	3.0	UHC00A0204	INTERNAL DIPOLE SVT TRAVERSANT - INT. TRAVERSANT DIPOLE BUSBAR	V. KLIZMENOV	Released
20379	3.0	UHC00A0302	CRU / DIPOLE SW. CONNECTION - FLANGE HEATING SUBAS... DIPOLE SW.	S. BOKLJIC	Released
10007	3.0		INSTALLATION FIRETHROUGHS FOR THE UHC 1500 RING DIPOLE (15) AND ...	Stark P., Williams L.P.	Released
20497	3.0	UHC00A0202	TRIP CIRCUITRY FOR DIPOLE - SW. CRU SUPPORT DIPOLE	V. SAVOT	Released

EQUIPMENT					
CRU-0479			RING BAR/COOL DIPOLE SW	130	Installed
HCD00A_001-8001018	3/518		Internal Dipole Bus Bar Type A	831	Accepted
HCD00A_001-8001019	3/518		Internal Dipole Bus Bar Type A	828	Accepted
HCD00A_001-8001020	3/518		Internal Dipole Bus Bar Type B	843	Accepted
HCD00A_001-8001021	3/518		Internal Dipole Bus Bar Type B	820	Accepted
HCD00A_001-8001022	3/518		Internal Dipole Bus Bar Type B	836	Accepted
HCD00A_001-8001023	3/518		Internal Dipole Bus Bar Type B	842	Accepted
HCD00A_001-8001024	3/518		Internal Dipole Bus Bar Type B	843	Accepted
HCD00A_001-8001025	3/518		Internal Dipole Bus Bar Type B	843	Accepted
HCD00A_001-8001026	3/518		Internal Dipole Bus Bar Type B	832	Accepted

Figure 6. EDMS Portal search results with limit row search

When the search limit is reached for the given criterion, the search engine puts some special information into the Oracle Context. This is possible even during the execution of the SQL statement (in fig. 4, the use of the Oracle Context is embedded in the call to EDBPKORACLETEXT.STF\$PUT\_VAR\_TO\_CONTEXT). Once the search is complete, the information stored in the Oracle context indicates whether the limit was reached or not during the execution (fig. 6).

### Special Syntax

There are many default syntax functions provided by the Oracle Text functionality. Using these functions

and applying them to developed rules for writing search requests, a flexible searching functionality is provided to users. The search engine recognises the following reserved characters:

- «+» – AND
- «|» – OR
- «"..."» – Exact match
- «!», «?» – Special symbols (explained below)

By default, the search engine works in «AND» mode, whitespace is a string delimiter and exact match for a particular word has a higher relevance score. For example, a user searches for the string: «100100 amont QIID», the search request generated by search engine will be:

```
((%100100% AND %AMONT% AND QIID%) WITHIN ALL)
```

For all search words with a length of more than 4 characters, the search engine adds a «%» sign to the left and right sides. For words of 4 characters, the «%» sign is added to the right side only. For words in «...» or shorter than 4 characters, no «%» sign is added at all. This behaviour is based on the analysis of EDMS statistics on how the search is actually used.

Prefix indexing in combination with reserved character makes possible various different syntax possibilities for search requests. The following example illustrates this:

```
"sector main test"|"100100"+QIID|12345|124546|QRL+MAIN
```

The generated search string is:

```
( ("SECTOR MAIN TEST") OR ("100100" AND QIID%  
OR %12345% OR %124546% OR (QRL AND MAIN%  
) WITHIN ALL
```

Moreover it is possible to use «!» and «?» signs at the beginning of the search request string. The «!» sign informs the search engine that a list of identifiers has been entered. Search words must be separated with a white-space or comma sign. In this case, the search engine automatically uses «OR» and exact match search criteria to all search words.

When the user puts a «?» sign before the search string, it is assumed that the rest of the string is written using native Oracle Text syntax for the CONTAINS command [10]. In this case the search string is fed into SQL query after removing the «?» sign. There is no risk of SQL Injection (malicious SQL text replacing what a standard search criterion should be) because the search request is parsed as a non-executable string and is used only with the CONTAINS function.

## Limitations

Though there are many advantages in using the Oracle Text technology, some problems during the development process were experienced. The most important problem faced was the Oracle error DRG-51030: wildcard query expansion resulted in too many terms. Oracle generates this error when the search process exceeds the value set by the WILDCARD\_MAXTERMS parameter. Moreover, when Oracle Text generates this error, the SQL query returns no results. It would have been very useful to have a parameter to set the maximum number of rows that the Oracle Text engine can return.

The second problem relates to adding a new search zone to a partitioned context index. It appears that when such an index is created it is not possible to add a new searchable section without recreating the whole index [11]. On the other hand, this is possible for non-partitioned Oracle Text indices. Non-partitioned indices for the EDMS portal were chosen for this reason.

## Conclusion

The new search engine opens new horizons for CERN EDMS. Currently, a central module gathers a major part of the information stored in EDMS and provides flexible search functionality.

With Oracle Text, a solid foundation for providing broad searching functionality for different kinds of information has been built and is ready for future challenges, as they arise in the coming life-cycle phases of the LHC. ■

## Contact

CERN

Anna Suwalska

E-Mail: Anna.Suwalska@cern.ch

## References

- [1] The EDMS service Web site URL, <https://edms.cern.ch>
- [2] C. Boyer, C. Delamare, S. Mallon-Amerigo, E. Manola-Poggioli, P. Martel, M. Mottier, J. Muller, T. Pettersson, B. Rousseau, S. Petit, A. Suwalska, D. Widegren  
«The CERN EDMS: An Engineering and Equipment Data Management System»  
Proceedings of EPAC 2002, Paris, France  
<http://accelconf.web.cern.ch/AccelConf/e02/PAPERS/TUPDO027.pdf>
- [3] T. Pettersson, 2003 <https://edms.cern.ch/file/370320/1/micado1.doc>
- [4] Oracle® Database Concepts 10g Release 2 (10.2) Part Number B14220-02  
[http://download.oracle.com/docs/cd/B19306\\_01/server.102/b14220/content.htm#sthref2643](http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/content.htm#sthref2643)
- [5] A. Suwalska «Oracle Text Search saves your time»  
Presentation at Oracle World Paris 2003.  
[https://edms.cern.ch/file/402581/1/Oracle\\_Text\\_OracleWorld2003.ppt](https://edms.cern.ch/file/402581/1/Oracle_Text_OracleWorld2003.ppt)
- [6] A. Tsyganov, S. Mallón Amérigo, S. Petit, T. Pettersson, A. Suwalska  
«A Search Engine for the Engineering and Equipment Data Management System (EDMS) at CERN» / Journal of Physics: Conf. Ser. V. 119 042029 (5pp), 2008
- [7] Oracle® Database Concepts 10g Release 2 (10.2) Part Number B14218-01  
[http://download.oracle.com/docs/cd/B19306\\_01/text.102/b14218/csql.htm#CCREF0104](http://download.oracle.com/docs/cd/B19306_01/text.102/b14218/csql.htm#CCREF0104)
- [8] Oracle® Database Concepts 10g Release 2 (10.2) Part Number B14200-02  
[http://download.oracle.com/docs/cd/B19306\\_01/server.102/b14200/functions123.htm#SQLRF00690](http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/functions123.htm#SQLRF00690)
- [9] Salton's formula:  
[http://download.oracle.com/docs/cd/B19306\\_01/text.102/b14218/ascore.htm#sthref2581](http://download.oracle.com/docs/cd/B19306_01/text.102/b14218/ascore.htm#sthref2581)
- [10] Oracle® Database Concepts 10g Release 2 (10.2) Part Number B14218-01  
[http://download.oracle.com/docs/cd/B19306\\_01/text.102/b14218/cqoper.htm#i996733](http://download.oracle.com/docs/cd/B19306_01/text.102/b14218/cqoper.htm#i996733)
- [11] Oracle® Database Concepts 10g Release 2 (10.2) Part Number B14218-01  
[http://download.oracle.com/docs/cd/B19306\\_01/text.102/b14218/csql.htm#CCREF0101](http://download.oracle.com/docs/cd/B19306_01/text.102/b14218/csql.htm#CCREF0101)



## Oracle Application Integration Architecture Global Customer Momentum and Success Continues to Grow

The demand and adoption of Oracle Application Integration Architecture (AIA) continues to gain momentum in 2009. Oracle customers have attained the benefits of Oracle AIA, including recent implementations at leading companies such as Alticor Inc. (Amway), BaneTele, Ciena, Com Hem, Lyse Energi, Pella, Vodafone Qatar and Zebra Technologies. Oracle AIA is a pre-built, open and complete architecture for orches-

trating agile, user-centric business processes across enterprise applications. Benefits customers have seen, include the ability to connect end to end business processes across multiple applications, simplify common business process bringing together their disparate systems, rapidly respond to changing customer and business demands and lower overall total cost of ownership. Powered by Oracle® Fusion Mid-

dleware, Oracle AIA's application independent framework enables organisations to utilise the applications of their choice to create composite business processes unique to their business on a flexible service-oriented architecture. As a result, Oracle AIA helps turn rigid IT landscapes into flexible, integrated environments that can adapt and scale more quickly to business needs.