Open Ingres IMA - Myth and Reality

By Dennis Adams

This article discusses the IMA facility, which was introduced as part of CA-OpenIngres version 1.1 in 1997. The article is now slightly dated. The latest version of Ingres is INGRES II, introduced in 1999. However, the basic principles still apply. (See footnote: [1])

Many CA-Ingres users will by now have received their GA version of CA-OpenIngres 1.1, which includes the new Information Management Architecture (IMA). Even at this early stage, there seems to be a mystique growing up around this feature. In reality, IMA is a straightforward gateway product with wide-reaching implications.

What is IMA ?

Suppose that you want to find out something about the internals of your CA-Ingres installation in order to understand how it works, and how to tune it. For example, you may want to find the size of the QSF pool, and how much it is being used. One way to do this would be to load the SQL terminal monitor and enter the following commands:

```
            set trace point qs501;
            set trace point qs502;
```

This returns a simple report which shows the amount of qsf memory, number of objects etc. Some problems with this approach are ...

1. the trace points are not officially supported, and
2. the output is not in tabular format, so cannot easily be imported to another program such as a spreadsheet for analysis or graphing.

It would be far easier if an SQL "select" statement could be used to extract the same information. Then you could enter:

```
            select mem_left, tot_mem, qsf_objects
            from qsf_pool;
```

... to retrieve the same information. That is precisely the functionality which IMA provides.

What can (& can't) IMA do ?

IMA basically provides a way of using SQL select statements to extract the same information which is currently available using trace points, or by examining the output from logstat, lockstat, iinamu, ipm etc.

IMA enables developers to write performance analysis software using embedded SQL within a user application; ie there is no need to parse output of utilities to get information required. For the first time, this information has been documented and will form a proper Management Information Base (MIB).

[1] This article was written whilst I was Senior Technical Consultant at Common Sense Computing, and originally appeared in the UserGroup Newsletter “Talking Common Sense” in 1997.
Common Sense Computing was subsequently taken over by Quest Software.

CA themselves also use the IMA to access debug information on server performance. It is possible, with care, to update some of this information using an SQL "insert" statement. This enables some management functions (such as closing down a DBMS server) to be achieved within SQL.

IMA is only a gateway. There is no logging system. Once information has been extracted, there is no way of (directly) processing it, or exporting it using SNMP,. for example. It is up to third-party software developers to determine how to extract and use this information

One other feature of IMA is that all select statements are "dirty reads" - they are a snapshot of memory locations at that time. There is no transaction control over IMA "tables". Similarly, the "commit" statement has no meaning.

How do I Extract the Data ?

Once developers have determined which information is required, they need to register a table which maps onto the IMA MIB structures. The syntax for this is

```
            register table ...
                  (columns...... )
            as import...
            with dbms=IMA;
            \go
```

This syntax should be familiar to users of any of the CA-Ingres gateway products. In fact, IMA can be regarded as just another gateway. The difference is that the IMA "tables" are not, in reality, database tables. They simply represent a convenient way to pointing to values held in memory structures in DBMS servers, name servers, etc. Special system catalogs hold the registration information.

Table names can be any name you like, although as always, consistent and sensible naming conventions will help make the data more accessible. I/Watch creates the tables with the prefix "iw_*" and a meaningful name describing the data extracted.

IMA tables can be joined, and views can be created on them. For example, you can register a table which maps onto the Server lock-lists, and join it to another table which maps to the Server locks or resources.

What can it tell me ?

Very simply, almost everything !

The IMA provides information on CA-Ingres internals which is currently available using standard utilities (lockstat, logstat, ipm etc), via simple SQL "select" statements. This includes the following...

- DMF cache
- Logging
- Locking
- GCA
- QSF/QSR
- System Control Facility

It would be possible for a knowledgable developer to re-write all of the standard Ingres utilities, using IMA calls.

Documentation - Please !

IMA documentation can be a bit sparse.

Its worth mentioning that the data structures are all "experimental". This means that the naming may change in a future release of OpenIngres. However, if tables registration is planned carefully, any future changes of datanames can be dealt with simply by changing and re-registering the tables.

The System Reference Guide has an appendix (Appendix E) which contains a good introduction to people wanting to find out about IMA. Those with access to Advisor can also extract more detailed and up to date documentation.

What of the Future ?

The future may see many other products like I/Watch taking advantage of the new open-ness of Ingres, to extract and analysing performance monitoring information. Developers may also want to use it within housekeeping utilities. For example, you could determine the available size of the log file if running a large SQL statement.

Conclusions

With the introduction of OpenIngres 1.1, developers can access the Management Information Base by registering tables. This is a major improvement on previous ways of extracting information, which were often unsupported and inefficient. However, the information is only useful if it is being processed effectively. We need to extract this data, log and process it, and then display it in a way that is meaningful for the end-user.

In practice, the real value of the data lies in the quality of the application which interprets and presents it. IMA greatly enhances the value of performance monitoring tools like Common Sense Computing's I/Watch

Dennis Adams is a Database Systems Consultant. A member of the British Computer Society, he can be contacted via email at Dennis.Adams@bcs.org.uk.