CAFS IN ACTION

Working party report

November 1985

ICL COMPUTER USERS ASSOCIATION (UK)

CAFS SIG

1985

Typeset on a Monotype Lasercomp at the Oxford University Computing Service

Contents

Preface

1 Management Summary

1.1 CAFS
1.2 Background to report
1.3 Working party terms of reference
1.4 Report contents
1.5 Report conclusions

2 The History of CAFS

2.1 Introduction
2.2 The sixties
2.3 The seventies
2.4 Into the eighties

3 CAFS User Survey

3.1 Analysis of questionnaire responses
3.2 General comments
3.3 Quotations

4 Case Studies

4.1 Berkshire County Council
4.2 Central Electricity Generating Board
4.3 The Inland Revenue
4.4 Logica
4.5 Northampton Borough Council
4.6 Oxford University Computing Service
4.7 Southern Water Authority

5 The Current CAFS System

5.1 The CAFS engine
5.2 The CAFS Search Option
5.3 The Direct CAFS Interface
5.4 QUERYMASTER
5.5 The Relational CAFS Interface
5.6 The Relational System
5.7 The CAFS context
5.8 Connectivity, co-existence and performance

6 Using CAFS

6.1 Introduction
6.2 Existing systems and data files
6.3 The evolution of existing systems
6.4 Future systems—general
6.5 Future ISAM systems
6.6 Future IDMSX systems
6.7 Text systems

6.8 Future batch systems
6.9 Using RCI
6.10 Using DCI
6.11 CAFS search options

7 CAFS and the End User

7.1 CAFS/QUERYMASTER
7.2 CAFS/QUERYMASTER training
7.3 Data training
7.4 Computer concepts training
7.5 User support
7.6 CAFS/CSO, RCI, DCI

8 CAFS—the Cost Case

8.1 Introduction
8.2 Hardware requirements
8.3 Software requirements
8.4 Installation and service development
8.5 The end-user area
8.6 Charging for CAFS services
8.7 Conclusion

9 The Future of CAFS

9.1 The working party view
9.2 The future of CAFS—an ICL view

Bibliography

Preface

This 'CAFS in Action' report has been produced by a working party consisting of the following members:

Eric Shaw Phillips (Chair)	–	Burton Group PLC
Paul Vaesen (Secretary)	–	Southern Water Authority
Phil Brown	–	National Computing Centre
Lou Burnard	–	Oxford University
Hamish Carmichael	–	ICL
Ian Carpenter	–	North Thames Gas
Rosemary Hardie	–	Northampton Borough Council
Guy Haworth	–	ICL
Martyn Kenyon	–	North West Gas
John Newton	–	CEGB (North West)

I would like to express my thanks to all members of the working party, each of whom has produced a chapter of the report, especially Paul Vaesen who has acted as secretary, and Lou Burnard who has edited the report. In addition, I would like to thank ICL for providing us with confidential advanced information of new products and product releases, and the various organisations who have contributed the reports included in the Case Studies chapter. Finally, I would like to thank everyone who has co-operated with the written and telephone questionaires we have used to gather information from the CAFS community.

The views expressed in this report are those of the working party members and do not necessarily represent those of their organisations, nor those of ICL or the ICLCUA (U.K.).

ERIC SHAW PHILLIPS
The Burton Group PLC

1 Management Summary

1.1 CAFS

For those few readers who do not know, CAFS is a system developed by ICL to search through data at speeds of several million characters a second.

Its full name is Content Addressable File Store Information Search Processor, CAFS-ISP or CAFS for short. It is an intelligent hardware-based searching engine, currently available with both ICL's 2966 family of computers and the recently announced Series 39, operating within the VME environment. It uses content-addressing techniques to perform fast searches of data or text stored on discs: almost all fields are equally accessible as search keys. Software in the mainframe generates a search task; the CAFS hardware performs the search, and returns the hit records to the mainframe. Because special hardware is used, the searching process is very much more efficient than searching performed by any software method.

Various software interfaces are available which allow CAFS to be used in many different situations. CAFS can be used with existing systems without significant change. It can be used to make online enquiries of mainframe files or databases or directly from user written high level language programs. These interfaces are outlined in the body of the report.

1.2 Background to report

Following the formation of the ICLCUA CAFS Special Interest Group in 1981 a Working Party was set up to examine how CAFS could be exploited in a Data Processing Installation. This first Working Party culminated in the production of a report entitled 'Exploiting CAFS-ISP' which was published in July 1984, rapidly followed by a second edition in August 1984.

This first report (of which some 4,000 copies have been issued) was produced at a time when there were only a few live CAFS installations. Of necessity, therefore, much of its content was based upon the theory of how CAFS *could* be exploited rather than how it *was* being exploited. It was decided, therefore, to set up a second Working Party to examine the impact of CAFS in practice. This resulted in the production of this second report, 'CAFS in Action'.

This report can and should be read in isolation from the first report as all relevant material from the first report has been included in it.

1.3 Working party terms of reference

The Working Party's terms of reference which were agreed at its first meeting in October 1984 were as follows:

- To co-operate with the early users of CAFS and from them collect evidence of the use made of CAFS, of system performance and of any effect on existing methods and systems.
- To analyse from experience how best to adapt existing systems and software to make use of CAFS.
- To devise hints to good practices and techniques relevant to the design and implementation of new systems incorporating CAFS.
- To identify CAFS implications for site management in terms of costs, benefits and strategy; DP and user training, resources (VMS, store, disc channels, comms, networking) and resilience.
- To advise ICL of future requirements and current shortcomings in the CAFS offering.
- To publish the findings of the Working Party.

This report attempts to address these terms of reference.

1.4 Report contents

Following this management summary, the second chapter traces the history of CAFS from its conception in the early sixties through CAFS 800 to the present day CAFS-ISP.

The third chapter presents the results of a survey of all current CAFS-ISP users carried out by the Working Party between November 1984 and April 1985. Questionnaires were sent to all known CAFS sites and followed up, where possible, by telephone contact. This chapter analyses the responses.

The fourth chapter looks at some leading sites where CAFS has been in operation, and presents their own conclusions.

The fifth chapter gives a technical overview of CAFS, including its hardware and software requirements and dependencies.

The sixth chapter explains how CAFS can be used and its implications on system and file design. Both conventional file systems and IDMSX database systems are discussed, as well as future systems such as text processing systems.

The seventh chapter examines the effect of CAFS on the End-User.

The eighth chapter looks at the cost implications of CAFS and attempts to explain how a cost-benefit analysis might be carried out.

The final chapter looks at the likely future directions of CAFS, both as ICL envisage it and as the Working Party would like to see it. To arrive at their conclusion the working party compiled a list of all known errors, omissions, shortfalls and genuine new requirements and held detailed discussions with an ICL technical panel.

1.5 Report conclusions

The working party arrived at the following significant conclusions in the course of their investigations.

ICL definitely have developed a system which can give most large computer users significant benefits in terms of facilities and cost effectiveness over other mainframe computer suppliers.

However, there is still a certain lack of awareness of the potential uses of CAFS, not only in the computer community at large but also within ICL and its current users. To overcome this ICL need to continue their push to educate their own staff in the benefits, uses, problems etc. of the total system. On the user front the recently developed course 'Exploiting CAFS' goes a long way to educating the user community; users who wish to gain the most benefits from CAFS would gain much from attendance on this course.

The apparent ICL marketing strategy of 'CAFS with everything' seems to be having an effect. At the time of writing over 500 CAFS units have been installed, with many more orders or potential orders. It must be said, however, that many of these installed units are not yet being used 'for real'. Hopefully the above mentioned ICL and user education will overcome this problem—if not, DP management will question why they have CAFS units.

If ICL wish to maintain their undoubted lead in information searching then the CAFS product must continue to be enhanced. CAFS must be seen by all sections of ICL as a product of central strategic importance. The enhancements outlined in the last chapter of this report should form a good basis for its enhancement. Also, it is critical that ICL continue to learn and take heed of their more advanced users; future requirements are always changing.

It would seem sensible that a CAFS Working Party continue to exist in the future to help shape these requirements.

2 The History of CAFS

2.1 Introduction

The conceptual origin of CAFS is simply the explicit recognition that the activities of a living organisation of co-operating people are not totally ordered, so that their interactions must be described as the interplay of order and disorder. The use of information to maintain their organised behaviour is therefore also characterised by an element of disorder, in its fundamental sense of intrinsic unpredictability, and so, whenever a large amount of data is stored on magnetic media, some applications will require that it be accessed by search. The traditional approach to solving this problem has been to build indexes to cope with the more predictable searches. This is fine as far as it goes, but not all searches can be predicted, and if one tries to build indexes to cope with every eventuality—by, in effect, totally inverting the file—the task of managing the indexes, particularly when the data is volatile, can become intolerable.

Two very relevant quotations from Vic Maller (*ICL Technical Journal*, November 1979) are:

> There are instances where the indexes can occupy between two and four times the volume of the data to which they refer—a perfectly absurd situation!
>
> It should not be assumed that indexable operations cover the totality of useful functions ... Indexes are really very primitive projections of files and consequently their utility should not always be taken for granted.

Content addressing, or associative processing, is a natural way of trying to circumvent these difficulties. Very natural, in fact, since we are all very familiar with the marvellous effects we can achieve using these techniques in retrieval from the memories in our heads. Laymen have always assumed that computers work this way; look at the computers in *2001*, *Doctor Who*, or *Blake's Seven* to see how ordinary people think they ought to behave.

2.2 The sixties

In the early 1960s there was much speculation in the academic world on the potential value of associative stores, but few suggestions regarding practicable techniques for making such a store of adequate capacity at acceptable cost. In 1962 Gordon Scarrott, working with Roy Mitchell in the Ferranti computer department, was stimulated by this situation to point out to the Ferranti computer department management that if a store, of adequate capacity to be useful, conceptually rotates as a consequence of its operating principle—for example a magnetic drum or a delay line—then the mean time required for access by search is half the revolution time, exactly the same as for access to a known address. Hence associative access can be achieved with existing technology, without an additional access time penalty. However, in 1962 the term 'Data Base' had not entered the jargon and consequently the need for store access by automated search was not widely recognised. Moreover there were no resources available for exploratory development, and so no physical work was initiated at the time.

In 1969 George Coulouris, then lecturer in computer science at Imperial College, suggested to Gordon Scarrott a joint project with ICL's Research and Advanced Development Centre (RADC) to identify the intrinsic requirements for for file storage and propose ways for meeting such requirements—a suggestion that was immediately accepted. The late Roy Mitchell, then a senior member of the staff of RADC, took responsibility for the project, collaborating with John Evans, one of George Coulouris' senior students.

At that time moving head disc storage devices were coming into use, and by

December 1969 Roy Mitchell had proposed a combination of disc store, key store and comparator, search evaluation unit and retrieval unit, and coined the term 'CAFS' to refer to the system, a name by which this combination of mechanisms is still known.

Initially there was some confusion about whether the 'A' stood for 'Addressable' or 'Addressed', but this issue has long been resolved in favour of 'Addressable' since the system permits addressing by content but does not enforce it.

It is noteworthy that most of the individual events (i.e. comparisons) in the CAFS searching system take place at a time determined by the of availability data from the disc store, so that Roy Mitchell's design was a 'Data Flow' system as now defined. But of course the term 'Data Flow' was not used in 1969.

Since CAFS is an engineering innovation, its most basic feature—the autonomous searching mechanism—should be regarded as a novel and advantageous synthesis of ends and means. Gordon Scarrott and Roy Mitchell proposed the means in 1962, George Coulouris and John Evans recognised the ends in 1969, and Roy Mitchell proposed the synthesis in 1969. On this foundation others, notably Ed Babb, proposed additional valuable features during the development, while Vic Maller maintained the momentum of the project over a difficult period when ICL was preoccupied with defining and launching the 2900.

2.3 The seventies

During 1970 proposals for investigating and evaluating the characteristics of a CAFS device progressively took shape. Coulouris and Evans undertook technical research in the United States, leading to the publication of CAFS Report No. 1, 'Some Characteristics of Real Time Data Management Systems'. The specification of a joint project between ICL and Imperial College was submitted for possible support to the Advanced Computer Technology Project (ACTP). In June CAFS Report No. 2 gave 'An Outline of the CAFS Programming Interface'.

Coincidentally, in the same month Ted Codd published his seminal paper 'A Relational Model of Data for Large Shared Data Banks'. ICL has sometimes been criticised for not implementing CAFS more speedily; it is interesting to note that a similar length of time has elapsed before the relational model has achieved practical respectability.

In 1971 Roy Mitchell's design for an experimental machine was adopted in a project for the construction of a prototype at RADC, funded under an ACTP contract. The principles of the machine which became known as CAFS Mark 1 were published at IFIP 71 in a paper by Coulouris, Evans, and Mitchell entitled 'An Approach to Content-Addressing in Data Bases'.

Also during 1971 Vic Maller joined Roy Mitchell at Stevenage, to undertake applications studies, software management, and experimental evaluation. A large number of potential applications within ICL itself were readily identified, including systems in Group Purchasing, Spares Inventory, Personnel, and Production Scheduling. On a wider scale, major possibilities were identified, even at this early stage, in police systems, in the hospital service, in matching jobs and applicants, in services provided by libraries and information publishers, and in the preparation of government and trade statistical analyses.

1972 saw the completion of the Mark 1 prototype, and the start of an extensive testing programme. It was soon confirmed that data stored on discs could be searched and retrieved by the unique hardware developed by the RADC team at speeds far in excess of those possible using traditional software techniques. The Mark 1 machine used an 8-megabyte exchangeable disc attached to a 1900 series host, and already had the characteristic separation of functions between Key Store and Comparator Unit, Search Evaluation Unit, and Retrieval Unit. The Journal of the British Computer Society published another paper by Coulouris, Evans, and Mitchell entitled 'Towards Content-Addressing in Data Bases'. The success of trials at this stage justified initial contacts with the Post Office, for a study of the Directory Enquiries problem, and with the IEEE's INSPEC service as an example of a document storage and retrieval application.

By 1973 enough experience had been acquired for the design of a replacement machine to be started. This was intended to be a further research and experimental device, but already considerable debate had started within the Company on not only how, but also whether, CAFS could eventually be incorporated into the official Company product line; and what procedures and standards should be followed in order to translate information smoothly between the research and manufacturing processes. Meanwhile, the Mark 1 machine was successfully demonstrated to the Post Office Directory Enquiries management, and the outline of a possible PODQ trial was agreed.

1974 saw the construction of two prototype Mark 2 CAFS machines. Although it was clear that the major future market would be associated with what at that time was called the 'New Range'—subsequently the 2900 series—it was inevitable that the CAFS developments should be continued on the stable foundation of 1900 architecture. The Mark 2 CAFS was designed to search data held on EDS 60 discs, using multiple read heads and amplifiers to read and search up to 10 tracks at a time; (this multiplicity, which has sometimes been mistaken for a permanent characteristic of a CAFS device, was in fact adopted as a means of boosting the data transfer rate to a speed which matched that of the CAFS searching and evaluation mechanism). A modified 7503 was used as the control processor which interfaced with the host and directed the activity of the underlying CAFS components. A VDU was installed at Hemel Hempstead for preliminary experiments on Directory Enquiries; not surprisingly, these highlighted many requirements for enhancements and corrections to the pilot software, but also generated sufficient enthusiasm to justify progress to an extended trial.

In 1975 a CAFS Exploitation Review Committee was established to co-ordinate future policy. It considered how to take advantage of the Department of Industry's procedures for assistance under the pre-production orders scheme. Guarded interest was aroused in the CCA (predecessor of the present CCTA) for the possible application of CAFS in public service applications in, for example, PRISM, DHSS, Home Office. An initial submission was also made to the European Commission. A PODQ Project Team was established within ICL.

Throughout these years of the mid-70s, the CAFS team at RADC made continual progress in the development of software products and techniques to enable them to explore fully the possible ways of exploiting a CAFS search engine. Major advances were registered by Ed Babb and Len Crockford in the definition of data models to describe an online database in a manner which permitted relational processing with optimal efficiency. The facilities of the enquiry language also underwent continual pragmatic refinement, with particular attention being paid to psychological factors affecting the usability of the enquiry commands. The implications of CAFS for database design and structure provoked much lively debate, between proponents of IDMSX on the one hand and CAFS on the other; there was not much prevision of the eventual mutually beneficial synthesis of the two approaches!

During 1976 and 1977, as such development work proceeded, a rational programme for the transition of CAFS to full product status was developed. The prototype applications should be consolidated, and practical experience in live application conditions should be acquired at a limited number of active co-operating sites. This would initially require the manufacture of a small batch of machines, following the architecture of the Mark 2 engines but conforming more closely to normal manufacturing conventions. A CAFS Marketing Manager was appointed; formal disclosure procedures were set up to control the flow of information from the centre, through approved units of the Sales organisations, to candidate customer pilot sites. The PODQ trial also advanced, with the transfer of the CAFS enquiry software from Stevenage to the PODQ Project Team at Bracknell, where it was modified to incorporate the application-specific requirements that the pioneering experiments had defined.

During 1978 the main Post Office Directory Enquiry trials took place. Terminals in the Telephone Exchanges at Leeds and Leatherhead were connected to the CAFS machine installed at ICL Bracknell, where the records of 7,000,000 subscribers were held (this represented about half of the national directory). All the technical criteria for the trial were triumphantly met; for example, an average response time of 2.5 seconds had been specified, and over the whole trial the observed average was 1.7. There were also noteworthy and very positive feedback messages from the operators of the trial service. They very quickly found the best ways to use the facilities—how much or how little search information to enter, which terms were the most successful search keys, how best to proceed with incomplete information, and so forth. They also uniformly reported that the system was much more enjoyable and satisfying to operate than the previous 'conventional' techniques. All these reactions have now become commonplace, but at the time they were very welcome confirmation that we had been proceeding on the right lines.

2.4 Into the eighties

The main event of 1979, from the CAFS point of view, was the announcement of CAFS 800—the public version of the CAFS Mark 2 machine—as an official ICL product. One of the first of these was installed in ICL's own Group Information Systems, where the first application implemented on live data was the Personnel system. Over the next two years, a total of 10 such machines were installed, in very different organisations, and covering an astonishly wide range of applications. At mid-1985, 7 of these machines are still in very successful active service. These machines naturally attracted a great deal of interest, and were generally seen as providing sound confirmation of the claims for the principles of CAFS which ICL had developed over the years. On the other hand, with a complicated architecture and somewhat outmoded technology, they were adjudged to be expensive—at some 250,000 each! Nevertheless, although people found it difficult to justify such expenditure in advance, by conventional measures, nearly all users reported very satisfactory—though hard to quantify—'savings' as a result of using a CAFS service, whether in reduced tactical and ad hoc programming, improved quality of data, or greater accessibility of information.

The impetus derived from the CAFS 800 programme led naturally to the development and implementation of CAFS-ISP, bringing the proven techniques of CAFS into the mainstream of ICL's product line. From now on CAFS was to be progressively incorporated in the standard environment of VME, to become automatically associated with standard file and database structures, and to be available in the standard hardware of the 2966 family of systems. CAFS-ISP was officially launched at a well-attended open meeting of the CAFS User Group (perhaps it should then have been called the CAFS Intending User Group) in April 1982.

The CAFS-ISP hardware became available for Field Trial with QUERYMASTER during the second half of 1983. General Release followed in March 1984. In the year to March 1985 over 350 CAFS-ISP modules were installed, on well over 200 systems, and in 11 countries around the world. The same rate has continued: by the end of June 1985 the 500th order for CAFS-ISP on a 2900 system was being celebrated. In addition it had been announced that CAFS-ISP would be an automatic constituent of every Series 39 system.

This evidence confirms that CAFS has come of age. Its success is due to the combined efforts of many people—too many for them all to be cited. So let this review conclude by crediting the main groups only:

- the pioneers in RADC, without whose flair, vision, and pertinacity over many years there would have been no CAFS product;
- the integrators within Mainframe Systems, who have so completely incorporated CAFS in VME and the hardware of 2900 and Series 39 systems;
- the facilitators within Applied Systems, who have made available QUERYMASTER, RCI, and DCI, the three prime routes for the exploitation of CAFS;
- and the CAFS User Group, whose serious commitment to CAFS (as shown by this Report and its predecessor) has convinced the world of its importance.

3 CAFS User Survey

The working party decided that a survey should be conducted in order to identify the number of CAFS users, combined with such aspects of their type of applications and the sort of problems they were experiencing.

As a means of collecting the necessary base data, the working party designed a questionnaire (reproduced at the end of this chapter) and in November 1984, sent it to every registered U.K. CAFS site. Of some 136 questionnaires despatched, a total of 71 were completed and returned.

Those users who had provided a positive response, i.e. were using, or were planning to use CAFS, and who had indicated that they were willing to participate further in the analysis, were contacted by telephone several months later in order that an updated picture could be obtained, and certain details clarified. From these telephone conversations, working party members completed the second questionnaire reproduced below.

The results obtained from all the questionnaires were subsequently amalgamated and are presented here.

3.1 Analysis of questionnaire responses

Table 1 Questionnaire Response Rate

no. despatched	136
no. returned	71
response %	52

Table 2 Number of CAFS Units

	1	2	3	>3	Total
no. of sites	37	17	2	5	61
no. of units	37	34	6	34	111
% total (sites)	61	28	3	8	
% total (units)	33	31	5	31	

The high proportion of sites with only one unit at the time of the survey (December 1984) reflects the fact that many sites acquire a CAFS unit either with no firm plan for its use or as part of a planned upgrade.

Table 3 Machine Types

	2953	2957	2958	2966	2988	Unknown	Total
no. of m/c's	3	7	7	40	11	8	76
no. of CAFS	3	9	7	67	14	11	111

This spread of CAFS units reflects ICL's population of mainframes.

Table 4 CAFS Usage by Site

	Jan 85	Apr 85
no plans to use CAFS	4	5
intending to use CAFS	27	22
developing a CAFS service	16	14
running a live CAFS service	18	25

These figures indicate that at the time of the survey a quarter of the sites were using CAFS in a production service.

Table 5 Software Usage by Site

	Jan 85	Apr 85	planned
using QUERYMASTER	26	35	27
using RCI	0	4	31
using CSO	0	0	21
using DCI	0	2	6

These figures would indicate that although QUERYMASTER is a popular product, it is not sufficient on its own to meet the needs of all sites—many customers are waiting for other products that will utilise CAFS.

Table 6 Respondent Profile

Analysis by business type:

government	2
local authority	16
health	4
education	2
utilities	9
manufacturing	10
retail	3
financial	6
other	10

Analysis by application:

	current	planned
financial	17	12
admin	14	8
stock	6	6
research	3	3
other	12	15

Of all the sites represented in this analysis, half are in the public sector.

The analysis by application clearly shows that CAFS can be used across a wide range of systems.

Table 7 User Profile

	current	planned
DP	3	2
end users	7	1
both	22	17
don't know	0	5

Table 8 System Profile

Analysis by file type:

	current	planned
IDMSX	13	19
ISAM	26	16
serial	1	10

The planned figures seem to show that IDMSX and ISAM are viewed as alternatives, whereas currently there are far more ISAM-based systems. This reflects the widespread opinion that it is easier to apply CAFS to ISAM than to IDMSX.

Analysis by file size:

Smallest file in use	current	planned	*Largest file in use*	current	planned
< 10 Mb	18	7	<1 × 640	16	9
<160 Mb	8	2	1–5 × 640	5	4
>160 Mb	4	5	>5 × 640	5	2
don't know		3	don't know		3

These figures would indicate that the majority of live CAFS services are being operated on less than one FDS640.

3.2 General comments

During the course of analysing the returned questionnaires and the enusing telephone conversations, it became apparent to the working party members that there were a number of common areas of likes and dislikes among respondents, mainly concerning CAFS-related software.

A number of sites did not get started due to the necessity of going to a new release of VME (8.01) and upgrading certain items of software e.g. TPMS and IDMSX, with the inherent work that such an upgrade involves. A great number of respondents were interested in PSAM and RCI but were awaiting SV211 before making any serious evaluations.

A high proportion of the sites would appear to have acquired CAFS almost by accident as part of a machine upgrade, and a number of them were also in the throes of moving from DME to VME. In these cases the introduction of CAFS services was being delayed until VME operation was established. Again, those already running under VME often quoted the move to SV211 as a reason for delay, not wanting to introduce too many new things at once.

A lot of users felt that the availability of software to make use of CAFS was lagging too far behind the hardware. This had hindered much early development and use of the product. A perceived shortcoming of CAFS at the turn of 1984/85 was the dependency on QUERYMASTER, with a number of respondents flatly refusing to consider using this product, maintaining that it was too unfriendly and too restrictive in report facilities. Unfavourable comparisons were made with Filetab, Deke and various unspecified micro-based systems.

There was a certain (misplaced) caution amongst some users in going too far in letting their end-users use QUERYMASTER.

Almost everyone using CAFS was very happy with it, although some users found it difficult to obtain adequate software support from ICL. CAFS expertise within ICL seems to be distributed unevenly. Some sites reported far better local support for CAFS than others. To some extent this is a subjective judgement and may reflect particular sites' relationships with their local ICL support staff. However, a recurring theme was the lack of first line support for CAFS within ICL over the period of the survey. This has improved since, but ICL must ensure that their local support staff are kept up-to-date with the latest developments.

Although most sites reported that the CAFS marketing was being done well, they were often less sanguine about the associated software. There have been delivery delays with some software products and the long wait for the new release of QUERYMASTER did not impress.

Many sites indicated that they would be willing to pay for short, on-site courses on CAFS exploitation.

3.3 Quotations

'CAFS is a tremendous idea but the interfaces need improvement.'

'I would hope that 1985 sees a major advance in access methods to CAFS. It promises to be so, but we've had promises before.'

'Any future system must use CAFS.'

'CAFS/QUERYMASTER is currently used as an add-on to present systems and this gives simple value-added bonus. These are reduction in job costs by a factor of 10, saving in user-time waiting for search, and users more willing to use the system thus releasing development staff.'

'We require COBOL interface to fully exploit CAFS as a prototype device and general purpose system development tool.'

'As on-line ad hoc use grows CAFS will certainly alleviate the otherwise intolerable growth in processing/disc IO power needed.'

One user whose files were mostly small said that they 'accepted the theoretical advantages of CAFS for large files but found that [for them] it was a solution looking for a problem.'

Another user has an enquiry application where 'usage is not heavy but where it would be impossible to meet performance requirements without CAFS.'

CAFS - ISP WORKING PARTY QUESTIONNAIRE

Please return this form after completion to:-

Eric Shaw-Phillips,
The Burton Group PLC,
Hudson Road,
Leeds LS9 7DN.

NAME: JOB TITLE:

ORGANISATION: BUSINESS:

ADDRESS: TELEPHONE NO:

COMPUTER INSTALLATION:

OCP(s) - Main Store -

No. of Disc Controllers - No. and Type of Discs -

No. of CAFS Units - Date first CAFS Unit Installed -

CAFS USEAGE:

Do you support (or plan to) a live CAFS-QM service? -

When was it first (or will it be) operational? -

When was this service (or will it be) provided to non-DP Users? -

Do you currently (or plan to) use the Relational CAFS Interface (RCI)? -

When did you (will you) commence use of RCI? -

Do you currently (or plan to) use the VME CAFS Search Option (CSO)? -

When did you (will you) commence use of CSO (Previously 'PSAM')? -

What have you found the major current shortcomings of CAFS to be and what are your major future requirements for CAFS?

Can you quantify the cost benefit of CAFS and if so what cost benefit (in £ p.a.) can you attribute to it?

Any Further Comments?

Are you prepared to participate in further discussions on your use and experiences of CAFS with the working party?

Many thanks for completing this questionnaire.

CAFS-ISP WORKING PARTY - TELEPHONE QUESTIONNAIRE (3)

WP MEMBER NAME   PHONE DATE

CONTACT ORGANISATION

TYPE OF BUSINESS (e.g. government, manufacturing etc. - see list)

DO YOU CURRENTLY USE CAFS?   IF NOT WHY & WHEN?

WHAT TYPE OF CAFS APPLICATION HAVE YOU (e.g. finance, stock etc. - see list)

WHAT LEVEL OF USEAGE IS ENVISAGED (No. of enquiries, users, terminals)
- both now & future

CAFS FILES USED (now & future)

IDMS/ISAM/Serial
Live Data/File Copies/Snapshot (e.g. monthly)

Smallest File	: <10 mb	: <160 mb	: >160 mb
Largest File	: <1 x 640	: 1 - 5 x 640	: >5 x 640

WHICH CAFS FACILITIES USED (now & future)

QM/ RCI/ CSO / DCI

Now
Future

WHAT FACTORS MITIGATE AGAINST CAFS USEAGE (e.g. CAFS weaknesses & bugs, lack of CPU power, cost, staff shortages etc.)

WHAT OTHER WAYS DO YOU SUPPORT AD-HOC INFORMATION REQUESTS (e.g. QM without CAFS, RM, AM, other packages, file extracts etc.)

OTHER COMMENTS

4 Case Studies

This section describes in some detail the experience of seven early CAFS users. The sites whose reports are included here vary considerably both in the scale and nature of their operations and in the types of application area which they have identified as benefiting from CAFS power.

To a large extent, the individual case studies speak for themselves. However, some recurrent themes may be identified. These are:

(a) the importance of CAFS as a simple way of providing end user facilities;
(b) the extent to which the demand for CAFS-based end user services has exceeded expectation;
(c) increased information productivity, both as a result of the freeing-up of dp staff resources and as a result of the new kinds of analysis made possible by CAFS.

The sites whose reports follow are in alphabetical order:

Berkshire County Council

One of the most recent CAFS customers, and also one of the most innovative, BCC have installed in the new Reading Public Library, a viewdata system, based on CAFS and RCI, by which members of the public can interrogate the Library's catalogues.

Central Electricity Generating Board

The North West Region of CEGB is perhaps typical of many installations at which information is spread across several large ISAM files. The use of CAFS/QUERYMASTER has enabled these files to be combined into an integrated system for enquiry purpose without compromising their availability for online updating in a TP system and without an expensive conversion process.

Inland Revenue

Undoubtedly the biggest application for CAFS currently known of is the Inland Revenue's name and address tracing system. The application is simple, but the scale of the operation, involving a search through 63 million taxpayers' records, is daunting. The system is in the process of being converted to use DCI.

Logica

Logica was one of the first third-party software houses to recognise the importance of CAFS in bringing their products to the ICL marketplace. Their experience in implementing a version of the relational database management system RAPPORT which can exploit CAFS facilities provides impressive evidence of the flexibility of the programming interfaces now available to the CAFS system and of its beneficial effect on existing software.

Northampton Borough Council

Like many other small- to medium-sized installations, Northampton is making the transition from DME to VME. Many of its applications are still updated under DME, snapshots of the data being made accessible via ADRAM to a CAFS/QUERYMASTER service. Other applications began using PDS, making the transition to CAFS/QUERYMASTER when the files grew too large for PDS to handle conveniently. The impact on end user departments is typified by one early application in which a weekly report on expenditure previously circulated on paper is no longer produced at all, users preferring to interrogate the file directly via QUERYMASTER.

Oxford University Computing Service

OUCS provides computing facilities for academic researchers from a variety of disciplines, with applications ranging from library catalogues and statistical databases to catalogues of Greek vases and ancient Greek names. QUERYMASTER has proved invaluable as a way of providing quick access to IDMSX databases and as a means of dealing with overgrown PDS databases. OUCS also uses DCI, which is in many ways seen as more appropriate to the research environment than QUERYMASTER or RCI.

Southern Water Authority

One of the earliest CAFS customers, SWA have seen their confidence in the system fully vindicated. While 20% of their CAFS/QUERYMASTER applications are (online) replacements for what were formerly (batch) Filetab-type reports, 80% of the workload represents enquiries that would simply not have been attempted without CAFS.

4.1 Berkshire County Council

When it became apparent that a new central library for Reading was to be built, the Assistant County Librarian, John Hicks, approached the Computer Section with some ideas. The Library had had a computerised service for many years including the complete catalogue of books held in Berkshire, but access to this was available only at Headquarters, the libraries receiving the information on microfiche some weeks later. John Hicks thought it would be very useful if we could provide this service to the general public via terminals in the new library. Although a good idea conceptually, this presented technical problems: the general public usually enquire for books by making vague references to words in titles or by author. Author access was no problem as that already existed, but title access in a 'fuzzy' mode seemed impossible. However, at that time Berkshire County Council was about to take delivery of both BULLETIN and its first CAFS controller, following the acquisition of an ICL 2966 under a new leasing arrangement, and suddenly the solution to John Hicks's request was apparent. We would use the power of the CAFS-ISP to search a specially created title file and BULLETIN to present it in a friendly way to the public.

The implementation of this service necessitated the use of ICL's Data Dictionary System (DDS), a new product to us as we had previously developed our own Data Dictionary Language. Additionally, to use CAFS-ISP with a COBOL program required the Relational CAFS Interface (RCI) and a new COBOL compiler (i.e. C2COB130). ICL provided this software at short notice and work commenced on designing the required system. It was decided to create an extract file of the titles to pass through the Self Identifying Format (SIF) to be used by CAFS, rather than use the existing Catalogue file which was very large. There are some 273,000 titles in the Catalogue. The extract file also contains the ISBN to enable the program to access the Catalogue file, when the appropriate title had been selected by the reader. This file also had to be documented using DDS. The learning curve on this activity was very low, both products (i.e. DDS and RCI) being very straightforward to use.

Whilst waiting for ICL to deliver the RCI software, the screen design was undertaken and various options shown to the Library to arrive at the final design. We used QUERYMASTER, another new product for us, to verify the DDS entries and this had the further advantage of being able to tell whether the CAFS unit was working as well. QUERYMASTER was found to be quite easy to use although perhaps not as user-friendly as we would like. When the RCI software arrived, it took about a week to get it to interface with BULLETIN, and to provide a working program for the Library titles. Again the software was not difficult to install, allowing for the fact that ICL manuals are not easy to read.

The final system went on trial in the Leisure Library at Shire Hall for about 6 weeks before the opening, and was used in 'test mode' by members of the staff. A suggestions pad was attached to the terminals and as a result certain refinements were made. The service went live on the 1 July 1985 (on schedule) with the opening of the new Reading Library. Since its implementation, the service has been in constant use six days a week and has not yet suffered from a failure in either the CAFS unit or BULLETIN.

4.2 Central Electricity Generating Board

4.2.1 Introduction

CEGB-NWR commissioned their first CAFS-ISP unit in January 1984. During March, April and May of that year a series of trials were performed on CAFS to establish performance of the device in the context of applications running in the electricity generating industry. The trials were successful, and proved that significant performance benefits accrued from CAFS use. A full report was published, concerning the trial, to provide information for other parts of the electricity supply industry who were considering CAFS applications. Given the trial outcome, CEGB-NWR developed their first live CAFS enquiry service, using QUERYMASTER, on an application for Maintenance Recording and Work Control at large power stations (the MR System).

MR runs on one of the 2966 machines at CEGB-NWR's computer centre located near Stockport. The system is primarily online (based on TPMS) and is accessed 24 hours every day by various types of terminal equipment installed at user sites. Prior to the introduction of the CAFS service most non-TP file enquiries were handled by a combination of bespoke enquiry software (mostly COBOL) and FILETAB. The MR system is based on a number of inter-related conventional (ISAM) files, which together form a database reflecting the plant maintenance requirements of a power station.

4.2.2 Data sets held

The major data sets held concern Routine Work, Defects, Work in Progress and History.

The Routine Work data set holds details of recurring maintenance tasks, which are either time-based or 'as required'. Jobs range from minor tasks (e.g. relamping, painting) through to major overhaul exercises. The latter, in practice, consist of many linked smaller jobs.

The Defects data set contains data from the defect recording and processing system which controls records of plant defects, and other work requests of a non-recurring nature.

When future work plans are being assessed Routine and Defect work is combined to form a 'Work in Progress' file. As time goes on this file provides station engineers with a picture of past, present and future work load requirements. The WIP sub-system provides day-to-day work scheduling and control facilities as well as monitoring outstanding or overdue work and providing statistical breakdowns for station management.

A comprehensive history of past Routine and Defect maintenance is maintained in the History data set. Up to 3 years 'current' history is held online, older data being consigned to archive media.

Other files provide networking facilities for Routine jobs and, for nuclear sites, a Maintenance Schedule facility to ensure that work is carried out at the correct time and in the correct sequence.

The data held for each site is sizeable—Heysham Stage 1 has over 40,000 routine jobs on the system, has typically 7000–8000 entries on 'Work in Progress', covering current work and a five-week planning horizon, and the maintenance staff perform upwards of 1000 jobs per week. When overhauls are being performed these figures become far greater, as a large number of tasks have to be performed in a short period. Time off generation for high-merit plant is directly translatable into real financial costs for the CEGB.

4.2.3 The enquiry workload

As the system use has developed the enquiry workload has increased dramatically. The development of a CAFS/QUERYMASTER service was seen as a priority because:

- Ad hoc enquiry requests were straining the limits of the computer development team in providing either specific programs or FILETAB support.
- The large data searches were leading to long response times for users.
- Collation of data on the ISAM-based system was becoming a problem, Although the files were integrated using techniques such as key cross-referencing, this was causing throughput overheads and required skilled programming support. It was felt that the relational techniques available with CAFS/QUERYMASTER (e.g. value-based sets) could provide real benefits.
- A text-searching workload was building up, especially on historical data.

4.2.4 Some initial problems

The decision to 'go live' was taken in early May, and Heysham I started to use the CAFS/QUERYMASTER service in September 1984, although full introduction of the service, including user training, did not take place until October. The development of the service took longer than anticipated, primarily because:

- Some file designs were old, by modern standards, and extra Data Dictionary (DDS) work was necessary, including some record re-formatting, leading to a once-off cost in program modifications.
- Additional programming effort was required to convert text data into CAFS text (SIF) format. The History file used in the CAFS/QUERYMASTER service is a copy of the production file, converted to CAFS searchable form each day.
- User training was more demanding than anticipated. This was not so much in basic use of QUERYMASTER, but more so in imparting concepts such as value-based sets, and the relational view thus provided, and in developing techniques of enquiry specification to make optimum use of CAFS.

4.2.5 The current position

In May 1985 7 major power stations were using MR and 4 power stations were running live CAFS/QUERYMASTER services, with a fifth imminent.

The service is being extended, on a customised basis, to each MR user as quickly as resources permit. It has been found that the significant cost element was in the provision of the first user service, subsequent introduction spreading the one-off costs.

As far as the objectives in view were concerned, it can be fairly said that the CAFS/QUERYMASTER service has fulfilled them all. The level of user appreciation is extremely high, with quantifiable benefits being realised in such areas as improved responses, precision of enquiry and reduction in computer development workload. As a testament to the success of CAFS at CEGB-NWR its use has now been extended to three other applications, and work is in hand to provide CAFS access to mainstream IDMSX systems e.g. financial administrative. The results of CAFS access to these general commercial systems are awaited with interest.

4.3 Inland Revenue

4.3.1 Introduction

The Inland Revenue receives a large amount of correspondence from taxpayers and outside bodies. Each item of correspondence must be directed first of all to the district office which handles the particular taxpayer's affairs, and then to the Allocation Officer (AO) responsible for the taxpayer. At the time of the start of the computerisation of PAYE (COP) project, a large microfilm-based system was used to carry out the first of these tasks, while the second was achieved by storing manual records in alphabetical order within the tax office.

The COP system will involve a transfer of these manual records into a number of large IDMSX databases. Within the database a record is identified by the taxpayer's National Insurance Number (NINO), and this number will be quoted on all correspondence from the Inland Revenue to the taxpayer.

Much of the correspondence received by the Inland Revenue will not include the NINO, and many callers at tax offices will not have their NINOs to hand. Thus the Inland Revenue have embarked on a parallel project to implement a CAFS-based system which can be used to trace taxpayers' NINOS given their names and addresses.

4.3.2 Tracing at Centre 1

The Centre 1 office at East Kilbride in Scotland was selected for a pilot project of this type, as it has used computers for some years to keep track of the employees of Scottish-based employers. This pilot system has now been in use for some time and is extremely well received by the users at Centre 1. It contains records of around 4 million taxpayers, handles a peak load of around one trace every two seconds and gives an average response time of 3 seconds.

4.3.3 The tracing system

The tracing system runs on an ICL 2966 with a single CAFS unit. It is implemented in a TPMS service, and uses a database consisting of a single large Indexed Sequential file. The average record size is around 133 bytes, and a record is keyed by the first five characters of the taxpayer's surname followed by his NINO. The AO uses a TP transaction to enter the name, initials and other data relating to the taxpayer, and selected words from his address. Fuzzy characters can be used if required. The standard facilities of the ISAM RAM are used to restrict searches to the portion of the file containing the surname, and a CAFS search program constructed from the data given is used to select any relevant records. These are displayed to the AO, who can then choose the record in which he is interested.

Because ISAM facilities are used to restrict the scope of the search, the application rejects traces in which at least the first character of the surname is not present.

4.3.4 Updating

Updates to the ISAM file are carried out by a batch process which runs in the evening once the on-line service is closed down.

4.3.5 The future

The pilot tracing system has proved its worth and the Inland Revenue are now looking to implement a National Tracing System, which will be accessed via DTS and an X25 network from terminals in district offices via their own Regional

Processing Centre. As well as PAYE taxpayers this system will also allow tracing of self-employed (Schedule D) taxpayers.

The enormous size of the database required to support this many taxpayers, and the greatly increased number of traces compared with the pilot system, mean that special steps must be taken to handle the workload. The principal requirement is to severely restrict the amount of searching carried out per trace, and a system of secondary indexing is CAFS being implemented to allow this. Details of this approach are given in an article published in the November 1985 issue of the *ICL Technical Journal.*

4.4 Logica

4.4.1 Introduction

This note summarises the features of the RAPPORT-CAFS interface, our experience with it in the past year or so, and our plans for extending the implementation.

The whole feasibility of the RAPPORT-CAFS interface has depended on the availability and success of the ICL DCI product. A few remarks on our use of DCI and wishes for additional features are included at the end of this note.

4.4.2 RAPPORT users and CAFS

For end-users of RAPPORT applications, no extra learning is required over and above that required to use the ICL version of RAPPORT without CAFS. Such users will only notice the existence of the CAFS interface insofar as response times are shorter with certain types of database manipulation and the cost (when billed for OCP and input/output) can be considerably less.

4.4.3 RAPPORT internals

In standard RAPPORT the underlying file structure is very straightforward. It employs fixed-length blocks in VME Direct files and maps the logical records, as the user sees them, onto these blocks. Block spanning is not used. VME sees only the fixed length blocks, the buffering and mapping of logical records being handled internally by RAPPORT. Logical records are located in a file by the technique of open address hashing on the values of the prime key field(s) in each record. Indexes, also maintained by RAPPORT, may be specified by the DBA. RAPPORT determines the retrieval strategy at run time, without any explicit action by the user. If retrieval conditions are suitable, RAPPORT will use hashing to retrieve a record by the value of the prime hashing key; this normally costs at most one disc access. If prime key hashing cannot be used RAPPORT will look for an index and use that, costing normally two disc accesses at most. If neither method is possible, RAPPORT searches the file serially. It is this last method that is considerably accelerated by use of CAFS via DCI.

4.4.4 RAPPORT modifications for CAFS

Modifying RAPPORT to take advantage of CAFS was a relatively simple task. Few modifications to RAPPORT main-line code were required; these were in the areas of:

- replacing RAPPORT blocking of logical records by VME blocking;
- using ISAM rather than Direct files;
- implementing code to translate RAPPORT retrieval conditions into DCI-readable format. An additional buffering algorithm was added to ensure adequate performance for multi-user implementations and to cover the possibility of nested searches on several files at once (e.g. a relational join).

4.4.5 Performance

As long as the ISAM files were buffered adequately for non-CAFS work (i.e. between 3 and 5 buffers per VME file), the speed of CAFS over non-CAFS searches was about a factor of 3 in elapsed time. OCP time was ten times less with CAFS. Input/output was, of course, dependent on the hit-rate, but always less for CAFS than non-CAFS. This saving in resources is expected to improve throughput, especially in multi-user environments.

4.4.6 Users

An existing RAPPORT customer has now ordered the CAFS interface and it is currently being installed under VME release SV201. We have taken advantage of the intervening period to tidy up and productise the set-up procedures in order to ease the DBA's job. The next development will be to include the CAFS option under the existing RAPPORT-TPMS interface. The amount of extra work involved is fairly minimal and the full RAPPORT-TPMS-CAFS offering should be available as a supported product early in 1986.

4.4.7 Additional features

The new release of RAPPORT (version 5), is currently being ported to various machines, including ICL VME. It includes extended facilities for substring searches of data files containing text and will be taking as much advantage as possible of DCI capabilities in this area.

4.4.8 Problems with DCI and CAFS

On the whole we have been very happy in our experience with DCI. All those features of DCI we have used have performed flawlessly. Only in the following aspects did we wish for a bit more functionality:

- due to strong typing of data, it took us some time to discover which types of FORTRAN argument would work with each DCI call;
- in order to prevent lost updates we have had to ensure that all CAFS searches are preceded by a COBOL CLOSE/OPEN to flush the VME buffers to disc before a search begins;
- it would have been far more convenient, from our point of view, if we could have specified our own logical record layout to DCI (at run time), superimposed on the VME records/blocks. However, by generating suitably shaped input/output modules in COBOL we have got around the problem for the time being.

4.4.9 The future

We are hoping to make RAPPORT-CAFS the standard offering for ICL VME installations from SV211 and DCI 100.1. An active Logica-ICL Software Agency Agreement is in the process of being set up and development work is already under way on an informal basis using ICL's facilities at Feltham.

4.5 Northampton Borough Council

4.5.1 Introduction

Until November 1982 Northampton Borough Council ran a heavily overloaded 1902T with 30 terminals attached to two real time systems. A considerable amount of development resources were spent on tuning programs to fit the machine and writing ad hoc enquiry programs. As the backlog of work increased, end users became increasingly discontented with the computer service.

A new machine—a 2958—was installed in November 1982. As there was no possibility of increasing staffing levels to cut the backlog, a decision was made to make use of products such as QUERYMASTER, Reportmaster and PDS/QUERYMASTER.

The field trial release of QUERYMASTER 210 was delivered in July 1983. Whilst these trials went well it became clear that CAFS would not be supported on CME sites until a later date. The date slipped considerably and eventually the CAFS unit was delivered in May 1984.

4.5.2 QUERYMASTER

With limited development resources—10 including analysts, programmers and technical support—lengthy performance trials were not attempted. Within a couple of weeks of installing the field trial version of QUERYMASTER it was made available to end users.

The first systems to have QUERYMASTER access were Expenditure and Income, Housing Repairs and Sundry Debtors. All of these are DME systems. ADRAM is used to copy the data to VME after each update.

Users found QUERYMASTER easy to use and after a short training session they were able to make basic enquiries. A follow-up training session then dealt with macros and more complicated enquiries. Initially, users were not trained in the use of 'MAKE' and 'EXTEND', but now the majority of users (especially those who are familiar with PDS), are using these commands with confidence.

In the early stages of use the QUERYMASTER problem which affected most users was its slow run time. An enquiry on the annual expenditure and income file—a 60 Mbyte serial file—could take 4–5 hours for a single enquiry, or 7–10 hours if more than one user was making an enquiry on the same file.

No attempt was made to perform sophisticated time trials on an empty machine. Instead, a program was written to convert the expenditure and income file to an indexed file, so that it could be searched by QUERYMASTER using CAFS. The run time was reduced to 2–3 minutes. To say the end users were happy is an understatement! The Queryview for this system was made available to all departments of the Borough and production of the multiple copies of weekly tabulations was stopped. The users are happy to rely on CAFS/QUERYMASTER enquiries.

The main usability problems users have experienced with CAFS on ISAM files are as follows:

- the inability to handle zero dates consistently
- CAFS not being invoked for 'CONTAINS'
- reduced amount of work space for CAFS query views
- CAFS not available for Serial files.

The last problem causes a lot of frustration with users. They want 'CAFS with everything' and cannot understand that data captured by an old system may not use CAFS without system alterations.

4.5.3 CAFS on IDMSX databases

A new stores system was implemented in October 1984. This was written using TPMS and IDMSX. Infrequently used enquiries and inverted file enquiries were specifically excluded from the TP system design. It was decided that these should be performed by CAFS/QUERYMASTER.

The effect of CAFS on this IDMSX system has not proved as dramatic as on ISAM files. This is understandable bearing in mind that the database design causes sets to be read rather than a realm searched.

Subsequently two more IDMSX based systems have been implemented using Application Master.

Experience with all these systems has shown the benefit of duplicating key data and the simplification of the database design.

4.5.4 CAFS on PDS databases

Considerable use is made of PDS, both for prototyping and for users to develop their own systems.

Some of these systems hold large volumes of data. Definitions of some of the databases have been entered into the dictionary so that advantage can be taken of CAFS via a Queryview of the database.

4.5.5 Effect on the users

Users are not restricted to any one department of the Borough or only the managerial levels. QUERYMASTER/CAFS is being used by Managers, Accountants and Clerks. There are few people who find QUERYMASTER difficult to use.

However, before CAFS was available, the long run times discouraged some users from making frequent enquiries. Since CAFS has become available, 'browsing' through files has become a reality and several departments are no longer requiring their systems to produce regular reports but are happy to rely on QUERYMASTER enquiries.

Since CAFS has been available end users have made considerable use of QUERYMASTER as an aid to decision making.

The success of QUERYMASTER/CAFS is shown by the fact that the number of terminals has increased to 150. There are 65 separate Queryviews in use, three of which (Stores, Plant and Direct Debits) access online TPMS/IDMSX systems, while the remainder operate on ADRAM copies of DME data. The applications covered are:

Stores, Vehicle and Plant, Direct Debit, Income and Expenditure, Housing Repairs (including inspections, contractors and DLO jobs), Housing Rents, Housing Property Register, Housing Cash Income, Housing Rent Arrears, Unified Housing Benefits, Rent Roll Bills, Trade Refuse Bills, Betting Gaming and Lotteries Bills, Car Park Bills, Allotment Bills, Manual Bills, Engineering Repairs Master, Electrical Repairs, Mortgages, Technical Staff Costing, Payroll, Rating Master File Details.

4.5.6 Effect on the computer services section

The most noticeable effect has been a considerable reduction in requests for ad hoc enquiry programs. This has left development staff to work on major projects.

Computer staff need to be available to act in a consultancy and training role. Users need a quick response to problems and so an 'Action Desk' has been set up. This is manned on a rota basis by senior staff, who deal with all enquiries relating to QUERYMASTER, TP and general computer problems.

The CAFS unit has been reliable and the original objectives of giving end users quick access to their data without considerable effort from development resources has been met.

The main disappointment from the computer services point of view has been the lack of ICL documentation on CAFS and the general lack of expertise of CAFS, QUERYMASTER and PDS in our dealings with SSC.

4.5.7 Future plans

A second CAFS unit was installed in January 1985, and CAFS is seen as a vital part of future computer strategies.

Cafs Search Option and Relational Cafs Interface have been purchased. CSO will be used to enhance Reportmaster programs.

Developments are currently taking place to link QUERYMASTER and PDS output files to 8800 Word Processors.

4.6 Oxford University Computing Service

4.6.1 Introduction

OUCS provides computing facilities for all postgraduate research work at the University. We do not support university administration or any significant amount of teaching. We act as a bureau and an advisory centre for any research project within the university which needs computing facilities; our end users are thus very varied both in their level of computing expertise and in their academic backgrounds.

We run a 2988, with a user filestore comprising 8 FDS640 discs, only four of which are connected to a CAFS DCU2 disc controller at present. We are currently running under VME release 8.11. The CAFS unit was installed on an experimental basis during early 1984 and since then several CAFS applications, using IDMSX databases, serial files and indexed sequential files, have been implemented. We expect the number to increase significantly within the next year.

Until October 1984 the only software available capable of exploiting CAFS was QUERYMASTER 210.9. Since October we have tested a pre-release version of DCI (Direct Cafs Interface) and have now installed version 100. We also intended to take advantage of CSO (Cafs Search Option) when this became available but were disappointed to find that (despite earlier assurances) it was unusable in our non-Cobol environment.

In a service such as ours it is very difficult to predict the type of usage to which novel equipment will be put. This is one reason why we are concerned that all of the capabilities of the CAFS hardware should be available to our users, not just that subset which is currently accessible through QUERYMASTER. We see part of our role as being the development and exploitation of new CAFS-based research techniques.

To date, applications have fallen into three categories:

- IDMSX databases accessed by QUERYMASTER
- ISAM files and PDS databases accessed by QUERYMASTER
- serial and ISAM files accessed by DCI from Fortran77 and Algol68 programs.

4.6.2 IDMSX databases

Lexicon of Greek Personal Names Project (IDMSX database, 120 Mb.)

This is a major research project of the British Academy, the aim of which is to produce a five-volume record of all Greek personal names attested from earliest historical times up to the mid 7th century A.D. Typesetting of the work on the OUCS Lasercomp phototypesetter will be database driven. It is also intended that the database will offer a shared, online enquiry service to other interested scholars. QUERYMASTER and CAFS have two major roles to play in this project:

- As an interactive processor for ad hoc queries, e.g. 'How many women from 4th century A.D. Chios have names beginning with K?'
- As a simple front-end processor in the production of ad hoc reports. QUERYMASTER is used to generate a list of database keys of records satisfying a particular set of retrieval criteria. This list is then used to drive a Fortran DML reporting program which generates either a line printer listing or data for input to the typesetter.

Much of the design and implementation of this database predated the arrival of CAFS at OUCS. Nevertheless CAFS/QUERYMASTER has been a valuable add-on extra at little or no programming cost, particularly as a means of producing reports which would otherwise have required special programming.

Beazley Archive Database Project (IDMSX database, 50 Mb.)

This project, which is funded by the Getty foundation, aims to computerise the catalogue of Greek vases begun by Sir John Beazley in the 1950s. The catalogue enables scholars to trace the whereabouts and cataloguing histories of all surviving vases, indexed by their iconography and the artists to whom they have been attributed. As with the Greek Lexicon, the project has been in progress for several years, initially using a simple flatfile-based information retrieval system called Famulus.

The facilities offered are also very similar to those of the Greek Lexicon project: the database is updated by a simple interactive program using Fortran DML and interrogated by QUERYMASTER, which is also used to produce lists of hit records for input to a specialised report program. One novel feature is the use of CAFS text searching capabilities: a full text description of the iconography of each vase is held in the database, so that the QUERYMASTER user can request information about (say) vases decorated with pictures of Herakles and the lion; controlled keyword indexing is also used to ensure consistency.

The database was demonstrated (live) at a recent international conference of art historians in Pisa, where it reportedly aroused considerable interest.

The Ashmolean Project (prototype under development)

Following on the success of the above two projects, we have recently begun to design a full scale IDMSX/CAFS database system to support all aspects of the cataloguing and conservation records of the Ashmolean Museum. This is the first IDMSX database which we have designed specifically with CAFS in mind, and it will exploit such features as text quite heavily; this is seen as particularly appropriate for this application which is typified by partial and variable data.

Implementation of the pilot project, restricted to the Museum's ancient Egyptian collections, began in summer 1985, using Application Master, DCI and QUERYMASTER.

4.6.3 PDS/QUERYMASTER and ISAM files

The design and maintenance overheads of full scale IDMSX systems would not be appropriate for many of the research applications we support, most of which are single-user systems of limited duration. For such users, particularly those unfamiliar with data processing, we have found PDS to be very suitable entry-level software. In many cases however, the large quantities of data involved can lead to serious performance problems on our heavily loaded 2988. In the absence of any intention on the part of ICL to provide CAFS searching directly from PDS, we have therefore provided several of our larger PDS users with query views, enabling them to access their PDS databases via QUERYMASTER and CAFS.

This has proved to be a particularly worthwhile exercise. PDS can continue to be used for database updates whilst QUERYMASTER is used for fast interactive retrievals. These can also take advantage of the improved fuzzy matching capabilities and of the output record-type facility, which are supported by QUERYMASTER but not by PDS110.

The exercise has highlighted for us some irritating differences between the two basically very similar interfaces. In our opinion ICL should give high priority to providing a packaged PDS to QUERYMASTER bridge, and should also ensure that the retrieval facilities of the two products stay in step.

Some typical applications which have benefited from the PDS/CAFS/QUERYMASTER combination are briefly described below.

The Iowa Database (5 tables, 2.5 Mb.)

This user is investigating the growth of Republicanism in 19th century Iowa. The database contains information on 70,000 votes cast in the Iowa legislature over a period of 30 years. Multi-dimensional scaling and other techniques are used to analyse the voting patterns of representatives from different social and political backgrounds on a variety of issues. The project was the subject of a paper presented at the recent International Conference on Databases in the Humanities and Social Sciences in the U.S.

Old English Glossaries (3 tables, 4.5 Mb.)

This user is working on a large collection of Mediaeval manuscripts in which Latin and Old English words are glossed in various different ways. QUERYMASTER is used to select groups of words which have been glossed in a particular way, or to identify the different ways in which a particular word has been glossed. The information provides important evidence in historical linguistics.

Social Mobility Database (12 tables, 8 Mb.)

This user is attempting to quantify various assertions about social mobility and class divisions over a twenty year period in 19th century Philadelphia. Data about membership in various elites and ownership of property and businesses was taken from such sources as street directories and social registers and entered in a normalised form. Simple time series analyses are performed to demonstrate the changing social structure of particular wards. The unpredictable nature and complexity of the queries involved have justified our faith in both the flexibility of the PDS/QUERYMASTER relational view of data and the performance benefits of CAFS.

In-house Databases (8 tables, 1 ISAM file, 10 Mb.; 7 tables, 1 Mb.)

OUCS offers a number of facilities in support of computer aided research in the humanities. Administration of two of these, the Text Archive and the Lasercomp typesetting facility, is also supported by PDS/QUERYMASTER/CAFS databases.

TOMES, the Text Archive database is used to maintain information about the origins and usage of our extensive archive of machine readable texts; it was described in a paper presented at a recent international conference on the use of computers in literary studies. As well as the tables used to monitor issues and receipts of text copies and to produce the Archive's printed catalogue, the Tomes query view also includes bibliographic data for some of the texts, which is held as CAFS-searchable text in a conventional ISAM file.

LASER, the Lasercomp database, is used to maintain all the information needed to run the national typesetting service efficiently. Monthly accounts and project registration and usage figures are all produced from the same system. In both cases, the CAFS/QUERYMASTER combination has proved invaluable as a means of providing ad hoc reports and enquiries at minimal cost.

4.6.4 DCI Systems

Major perceived weaknesses of QUERYMASTER in our environment are its poor support for textual data, its dependence on Data Dictionary and the comparative difficulty of integrating it with existing scientific software. Many of our users write their own programs, usually in Fortran or Algol68; others rely upon widely used scientific and statistical software packages such as TSP, SPSS or SIR. Clearly interfacing all this software to the QUERYMASTER view of data is not a task to be undertaken lightly, if at all.

By contrast, the DCI product enables any competent programmer to tap CAFS search power directly from any high level language, making only minimal changes to existing code.

The three projects described below demonstrate three of the DCI product's attractive features: first, its performance advantages over even QUERYMASTER and CAFS for simple applications, secondly, its comparatively sophisticated range of text searching facilities, and thirdly, its flexibility.

The Futures Markets Database (4 tables, 4 Mb.)

This is used by an econometrician working with commodity market time-series data. The aim is to investigate the costs and benefits of different pricing strategies in international markets. PDS was originally used for database creation and management and, under this regime, extraction of one year's data for a single market took, typically, 240 seconds of OCP time. Provision of a query view and CAFS access by QUERYMASTER reduced this requirement to around 60 seconds. Since the arrival of DCI all data extraction has been by a Fortran77 subroutine called directly by the analysis package and this takes 5 seconds for the same extraction (involving 3 ISAM files). As a result of this the user is now able to pursue various subsidiary analyses which would have been prohibitively expensive without CAFS.

Text Systems

Included in the Text Archive mentioned under §4.6.2 are many major literary works, analysis of which has hitherto been performed only by using expensive indexing and concordance generation software. As part of our evaluation of DCI, a simple text retrieval package was written in Algol68 and made available for use with a variety of texts. Full details of this software and its applications are given in an article to appear in the *ICL Technical Journal.*

The largest text database accessible in this way is the Shakespeare corpus, about 15 Mb. of running text now held as a single CAFS searchable sequential file. Scholars working on the new critical edition of the canon have found this a particularly useful tool both in the quantification of existing hypotheses about the transmission of the text and in the formulation of new hypotheses.

Also accessible by the same software are two large library catalogues (6 and 8 Mb. sequential files) prepared in CAFS searchable form to enable Oxford's library community (now entering the throes of extensive computerisation) to assess for itself the potential of CAFS.

Performance of all these systems has proved very impressive, the response time at a MAC terminal being limited more by the number of hit records returned than the size of the file, provided this is less than 20 Mb. or so. It is our view that larger files will still require some degree of secondary indexing to achieve reasonable performance.

The CSO Macro-Economic Time Series Data Bank

This data which is freely available to academic economists from the ESRC Data Archive at Essex University comprises around 11,000 sets of time-series data relating to all facets of the U.K. economy. The data is held in variable length records, making it difficult to access from standard statistical software packages. The object of the project is to make easily available to a group of around 20 interested researchers datasets of particular importance to their own field of study. For example, find 'all quarterly sampled datasets at 1980 base prices relating to domestic fixed capital formation'. Extracted data will then be passed to specialist time-series analysis software for further processing.

It is anticipated that this database will be 15–30 Mb. in total size and will be updated quarterly. The retrieval package will use DCI from Fortran77.

4.7 Southern Water Authority

4.7.1 Introduction

Southern Water Authority (SWA) is one of ten regional water authorities set up in 1974, reporting ultimately to the Secretary of State for the Environment. The Authority covers an area of 42,000 square miles, in Hampshire, the Isle of Wight, Kent and Sussex within which it provides a sewerage service to four million users, and a water service to two million. These services rely on 6,745 miles of water mains and 9,542 miles of sewage mains, use 426 sewerage treatment plants and incorporate 1,720 miles of river. In 1984, the Authority's revenue expenditure was £150M, and capital investment £60M. There are 4,000 employees. Apart from the principal water services, the Authority also looks after sea defences and flood alleviation, and provides ancillary services in amenity, recreation and fisheries.

SWA have been interested in CAFS for several years. They helped to found the CAFS Special Interest Group of the ICL CUA and provided the Chairman for one of its first working parties. They have been active in the preparation of both SIG Reports on CAFS.

Their installation has grown from a 1902A in 1974 to the current main equipment of a dual 2966. As well as 13 EDS200s, this system has 20 FDS640 drives, of which one is normally reserved as a resilence unit, giving a total online capacity of 115,200 Megabytes, all of which is accessible through the CAFS service. The service supports some 210 VDUs, any of which may exploit CAFS.

4.7.2 Product exposure

During the field trial of RCI, SWA's prime objective was to confirm the search power of CAFS and to investigate very thoroughly its system performance implications. They looked particularly at what happens when several CAFS users are active at the same time, measuring the effect of varying the 'fragment size', both on CAFS tasks and on their interaction with a TP service.

After completion of installation testing, the first ever CAFS enquiries at SWA were made on 7 July 1983 and cautious progress was made towards the first CAFS enquiries during the prime shift of the live service, which took place on 27 November 1983. From then until 14 March 1984 a limited service use of CAFS was available, restricted to Computer Services staff and a few end users under their direction. The success of these enquiries during this first tentative phase led SWA to place an order for three further CAFS ISP modules.

Since 14 March 1984, with the installation of the full complement of the CAFS hardware, the CAFS service has been open without restriction to all end users. This open doors policy is the strongest possible proof of SWA's confidence in CAFS itself and also of its perceived importance to their end users.

Software reliability has been very creditable. During the first nine months after the RCI field trial, SWA submitted 21 bug reports on DDS, 23 on QUERYMASTER and 8 on IDMSX, categorised as 31 at C priority, 7 at D and 14 at U. This confirms the judgement that the products are fundamentally very sound, with improvements required mainly on matters of style and usability.

The hardware has also clocked up a very satisfactory record. There have been no recorded faults whatever with the four CAFS modules themselves, though the associated power supplies have been less reliable.

4.7.3 Impact on end users

Overall, the end user effects can be summed up as substituting new methods of working for old, and opening up the Authority's database for types of analysis which previously were not feasible, either on technical or on cost grounds. As one example, the previous practice of running many overnight Filetab analyses stopped very quickly, their place being taken by online daytime enquiries. The importance to the users of the new more interactive style is well indicated by the speed and vigour with which they now complain if, for any reason, the CAFS service is not available to them.

Which end users are obtaining the most benefit? It appears that many of them are professionals in their own disciplines—accountants, engineers, biologists, chemists etc.—while others are at the most senior clerical grades. These people share the characteristic that they have the best and most detailed understanding of the stored data and are the most skilful at formulating search strategies that will elicit useful information. Although they may not themselves be very high up the management hierarchy, their output is often of direct value to management. Both their own enthusiasm for CAFS assisted enquiries and their managers' perception of the resulting improvements in efficiency are expected to spread direct usage of CAFS progressively to higher echelons.

Over seventy query views have been generated, giving many different users suitably personalised access to appropriate parts of the stored data. Some query views cover a single application, while others extend over several. The following table indicates the spread of interest.

Application	*Organisation*	*Previous enquiry mode*
Personnel	IDMSX	TP
Plant	IDMSX	TP
Expenditure	ISAM	IDH
Unmeasured billing	ISAM	IDH
Measured billing	ISAM	IDH
Bank reconciliation	ISAM	IDH
Stock	ISAM	IDH
Creditors	ISAM	IDH
Payroll	ISAM	IDH
Water quality	ISAM	IDH
Rainfall	ISAM	IDH
Capital programme	IDMSX	TP/MAC
Emergency centre	IDMSX	TP

Examples of new work being undertaken include more effective debt collection, because it is easier to review outstanding accounts and determine where collection effort can be most usefully deployed.

Because users can now begin to express enquiries that span the whole of the Authority's data, increasing emphasis is being placed on standards—particularly for data names and the definitions of data elements. The organisational implications are very interesting—local managerial autonomy is not jeopardised, but it is seen that this must exist within a framework of stronger corporate data discipline.

Understanding of the data by end users themselves is enhanced by the intimate contact provided by CAFS and QUERYMASTER. On the other hand, a slight problem has arisen among more naive users, who are liable to believe that CAFS and QUERYMASTER can act as a magic wand, to satisfy all their data processing requirements.

QUERYMASTER itself has proved very easy to learn. SWA's technical experts reckon that the basic enquiry functions can be mastered by a non-technical user within an hour. On the other hand, users who have never previously experienced direct encounters with a database can take a lot longer to grasp the principles of database structure and understand their effect on response times.

Finally the users, as one would expect, have produced a list of demands for improvements to the service. Chief among these are requests to make some parts of the service run even faster and to translate into more user-friendly language some system messages which are considered to contain too much computer jargon. But the more encouraging kind of demand is typified by the requirement for more MAC streams and for the online availability of historical data, so that enquiries with a time dimension can be more easily performed.

4.7.4 Impact on Computer Services

The management of Computer Services have laid down a policy rule that wherever QUERYMASTER is used CAFS must also be used, unless there are other over-riding considerations, perhaps on aspects of privacy. The norm is that QUERYMASTER is allowed to make the choice and so exploits CAFS wherever it is profitable to do so.

The only people who have proved hesitant about using CAFS have tended to be technical staff within Computer Services, who have been liable to prefer to exploit their understanding of IDMSX structures and traditional DML programming. The management policy in favour of CAFS has proved necessary to prevent inefficiency caused by this attitude.

Hitherto there has been little or no impact on the operations function; the QUERYMASTER service is simply another online use of files which are already mounted for the TP service. Also, apart from the work involved in the field trial of RCI, there has so far been relatively little impact on programming staff. By contrast, systems analysts and technical support staff have been more heavily involved, mainly in the production of query views.

The line taken is that for new and recent applications any query view should be produced by the analyst concerned, so that the generation of query views is being treated as an automatic element of system completion and documentation. For established applications, the user support function generates query views, taking on average half a day for each one. In every case scrupulous attention is paid to rules about data accessibility and no enquiry service can be initiated without the approval of the owner of the data concerned. Charts are always provided with query views on IDMSX systems, so that the users can understand in outline the navigational implications of any enquiry; on ISAM files charts are sometimes provided; otherwise their place is perfectly adequately supplied by the QUERYMASTER DISPLAY function.

It is particularly interesting that SWA have adopted such a liberal attitude to the usage of CAFS within their live service, permitting it to become the norm and imposing no controls other than the number of MAC vms which can be supported—currently 33, all of which are allowed to run any kind of work.

SWA confidently report that they have seen no noticeable effect of CAFS on their TP service. This is undoubtedly related to their choice of the minimum size—a single track—for the CAFS task fragments; at this level, a CAFS fragment is indistinguishable from any other disc access task, and VME scheduling provides all the necessary control and inter-leaving. Other system variables are observed to cause greater fluctuation in TP response times than any effects of CAFS, among them such familiar items as DCU2 dumps, NPS incidents, comms line failures, TP AVMS recycling, TP services loading and VM dumps.

SWA have found it necessary, for privacy reasons, to set up a dedicated QUERYMASTER service running under packaged MAC. By this means they are able to prevent QUERYMASTER users from accessing sensitive data to which they are not entitled. This is achieved by forcing a QUERYMASTER MAC session to take note of SWA's own privacy access details (protected by password) before asking the user to provide the name of the query view to be used. The only query views which the user is then able to access are those for which he is a registered bona fide user.

In their general workload patterns, SWA report that there has already been some shift of activity from overnight batch mode to online daytime processing. This has assisted and accelerated an intended change to unattended operation during the evening shift.

4.7.5 Summary

User demands and the system management problems involved in meeting them with CAFS are not new; they will be familiar to any professional DP department. CAFS should not therefore be feared for its novelty so much as welcomed for the help it gives in satisfying known requirements.

The most cost-effective way of evaluating the CAFS hardware is to use it.

DP managers should define a simple overall policy for the use of CAFS, rather than leaving it to the varying technophile or technophobe attitudes of individual analysts or programmers.

Software houses and other third parties should be persuaded to exploit CAFS in any new applications packages by making such exploitation a condition in any tenders. If this stipulation is not made by the DP department, it may well be insisted on by the end users!

5 The Current CAFS System

The current CAFS system consists of five elements, one of hardware and four of software, as follows:

- the CAFS-ISP hardware search engine itself, sometimes called a CAFS module or a CAFS engine, but normally simply abbreviated to CAFS
- the VME CAFS Search Option, CSO
- the Direct CAFS Interface, DCI
- the general purpose online enquiry package QUERYMASTER, often abbreviated to QM
- the Relational CAFS Interface, RCI, a programming interface implemented as an extension to COBOL

In the following five sections, each of these elements is described in detail.

Within VME itself, awareness of CAFS or the ability to exploit CAFS has been incorporated in many functions. A CAFS Code Generator converts each task into a microcoded form that can be obeyed directly by the CAFS hardware. The IDMSX database manager and the RECMAN conventional file manager both contain enhancements concerned with CAFS. The scheduling and control of disc tasks takes proper account of the difference between a conventional disc access and a CAFS disc scan. VME also manages the buffers through which hit records are passed from CAFS to the application program. However, it is not necessary for the user to understand the details of these operating system functions.

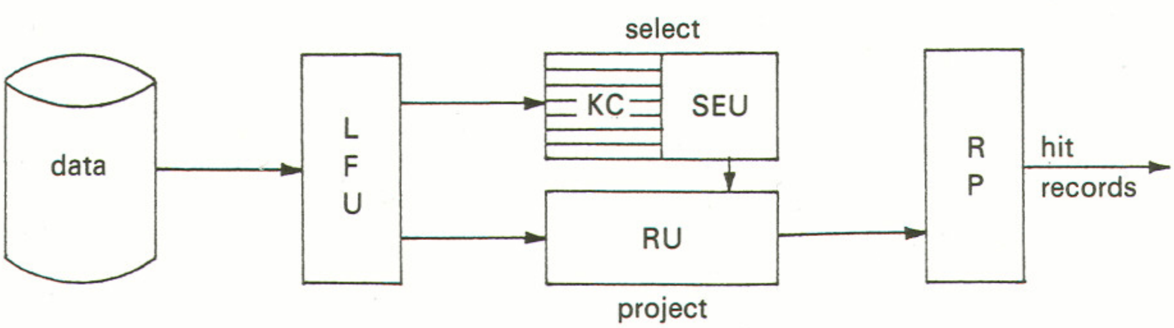
5.1 The CAFS Engine

This forms an extension to the hardware of a disc controller on both Series 39 and 2966-family mainframe systems. The controller retains the conventional ability to transfer data in either direction between disc and mainframe, a block at a time, and thus continues to perform all the functions of a traditional disc controller. But with the attachment of CAFS it also becomes capable of searching stored data.

In a CAFS search both parts of the disc controller hardware act together: the standard Disc Control Module moves the read heads to the right cylinder, senses the rotational position of the disc, and initiates the transfer of data; the CAFS engine searches the resulting data stream.

The CAFS engine as a whole is often described as consisting of an interpreter, a search mechanism, and a retrieval mechanism. All units operate under the control of a task specification, formulated in the host mainframe by a combination of application software and operating system facilities.

Figure 1 shows the main components of the engine in diagrammatic form.



5.1.1 The Logical Format Unit

The LFU acts as the interpreter within the engine. Its function is to identify significant points in the stream of data being transferred from disc. It is chiefly used to identify the start of a database page, the start and finish of a record, and the start of significant fields. Using this information it is able to issue control signals, telling the search engine when to start searching, when to stop, and when to evaluate whether or not the record constituted a 'hit'; and telling the retrieval engine when to start retrieving required fields, when to stop, and when to perform any post-retrieval processing functions.

The part of the task specification obeyed by the LFU contains a description of the layout of the records in the target file, together with a table of the addresses within each record at which searching and retrieval are to start and finish. Although a file being searched by CAFS may legitimately contain records of several different formats, only one record-type is scanned in each occurrence of a CAFS search, and so the task specification only defines a single record-type.

Fixed-length records are always searchable by CAFS, except in the rare case where they are shorter than 16 bytes in length. To be searchable by CAFS a variable-length record must also be greater than that length and must conform to a few further conventions. Its fixed-length header is unconstrained, but the trailing section must be one of the following:

- a single field containing a variable-length string of characters;
- an array consisting of one or more fixed-length fields repeated a variable number of times (although such data can be searched by CAFS, some subsidiary rules must be observed in order to avoid ambiguous results);
- a variable-length field containing text in the special CAFS-based self-identifying format (SIF).

5.1.2 The Key Channels

The Key Channels, or KCs, form the first stage of the searching engine. The function of each KC is to match a reference value provided as part of the task specification against the value in the appropriate field of the current record, and to note whether the comparison yields a result of equality, record less than reference, or record greater than reference.

There are 16 KCs in the current CAFS engine. Each operates independently of all the others. Thus in one search two KCs might be examining separate fields, while in the next they may be comparing for different reference values in the same field.

These comparisons are made on the fly, as the data being transferred from disc passes along the internal highway of the CAFS engine. The data is not in any way buffered while the comparisons are taking place. As soon as a field has been compared the KC is ready to be re-activated for the next comparison.

Associated with each reference value the task specification may optionally contain a mask, whose function is to eliminate from the comparison any part of the target field whose inclusion might invalidate the result. A common duty of such a mask is to implement the use of an omnibus character (sometimes also called a joker, don't care, or wild card character). Thus an enquirer interested in SMITHs, but not wanting to distinguish between those with an 'I' and the others with a 'Y', might ask for 'SM*TH'. This would be translated into a mask which excluded the third character of the field from the comparison. Masks can also be used to enable searching to be successful on fields which do not conform to byte boundaries and, in text, by masking out the case-bits to make a search case-blind.

Associated with each KC is a Key Comparator, in which the stored result (= , < , or >) of a comparison may be retained. The Key Comparator may be considered as a store with a capacity of 3 bits, one for each possible result. In the case where a target field is repeated within a record several successive comparisons may be made and several results may be stored. For example in a personnel history file the salary of a long-serving employee could be correctly registered as successively less than, equal to, and greater than some reference value.

VME always uses 1 KC to distinguish target record-types from any others. Up to 3 other KCs may also be pre-empted for system software purposes. The number of KCs actually available for the user's enquiries therefore varies from 12 to 15.

5.1.3 The Search Evaluation Unit

The function of the SEU is to decide whether or not a record qualifies as a 'hit'. To achieve this it must obey a program which links together, with appropriate ANDs, ORs, and NOTs, the results of all the comparisons made by the Key Channels. The SEU is activated, at the end of the record, by a control signal from the LFU.

The main part of the SEU consists of 16 co-operating processors which initially read the contents of the Key Comparators and subsequently operate on their very compact internal stores. Each of these processors is capable, in a single program step, of reading its store, reading a signal from one neighbour, emitting a signal to the other neighbour, and writing to its store. The combination of these primitive functions permits even the most complex logical expression to be evaluated in less than 4 microseconds. The result of the evaluation is held in a 16-bit store called the task word.

Also within the SEU are a Select Processor, which assists in the management of the task word, and a Quorum Processor which allows the conventional boolean logic to be supplemented by quorum expressions. These may either be simple—of the form 'any 3 out of 5'—or may include a threshold value and weights to be associated with each term.

The SEU also increments counts. A count is always kept of the number of hit-records identified during a CAFS search. Up to 15 other counts may be specified; their purpose is to show how many records satisfied nominated sub-conditions within the overall selection expression.

5.1.4 The Retrieval Unit

This unit is instructed by the LFU to retrieve either the whole or specified subsections of a record. It does this while the record is passing along the internal highway, and stores the retrieved material in in a 32-kilobyte buffer. When the end of the record is reached, the SEU decides whether or not it is a hit, and passes a suitable control signal to the RU. A hit record is then prefixed by an identifier, a length, and its task-word, and becomes available for possible further processing before being returned to the host. Retrieved data from miss records is discarded.

The RU also notes in the store the end of each logical block of data for subsequent checkpointing purposes.

5.1.5 The Retrieval Processor

Up to this stage the CAFS engine works as a strictly synchronous pipeline processor. The RP, however, operates on data that has been stored for it by the RU, as described above. This allows the front-end of the CAFS engine, in an extreme case, to work thousands of records ahead of the RP: the search engine examines all records, whereas the RP only operates on hit records.

The RP has two roles. One is to calculate data derived from the file-search as a whole; the second is to be the final arbiter about passing data back to the host mainframe. These are separable functions: the RP, after returning a nominated number of records to the mainframe, can cease to retrieve but continue to analyse the whole file.

The RP may execute a number of simple processing functions in a sequence determined by the task specification. Whereas the KCs performed separate comparisons between each field and the appropriate reference value, the RP can perform simple comparisons between fields in the same record. This provides a further level of selection logic: a record may have qualified as a hit at the earlier stages but be rejected by the RP on the basis of inter-field comparison.

Also the RP may be used to determine the maximum and minimum values of specified fields, and can take totals of the values in specified fields. Finally, the RP maintains a count of hit-records which may be used in two ways to limit the number of records returned to the mainframe host. In the simple case, when the given limit is reached, the RP terminates all CAFS functions, and the search ends. Alternatively, when the limit is reached, the RP may stop returning hit records to the mainframe, but continue all other functions, so that complete counts, totals, maxima and minima may be generated across the whole of the target file.

While the functions of the RP are reasonably extensive, it should be noted that it is not a particularly powerful processor. The remainder of the CAFS engine is capable of handling records at the full transfer speed of the fastest current discs, but the RP is designed for cases where hit records constitute no more than 10% of the file. Normally the throughput of CAFS is determined by the disc transfer speed, but at high hit-rates the throughput of the RP becomes the limiting factor.

In relational database terms, the Key Channels and Search Evaluation Unit jointly perform the SELECT operation, and the Retrieval Unit performs the PROJECT operation.

The essence of the CAFS design is the use of special purpose hardware operating with a high degree of parallelism. It is this which gives CAFS its outstanding performance, and enables it to perform search and retrieve operations at speeds well in excess of anything that can be achieved by traditional serial search software running in any machine of conventional von Neumann serial architecture. Furthermore, by removing the search task from the mainframe to the disc controller, CAFS eliminates almost all of the mainframe processing previously required. The software running in the mainframe formulates the task specification and processes hit records, but is relieved from all the heavy work of identifying hit records.

5.2 The CAFS Search Option

The objective of this software, which is an optional feature of the VME operating system, is to enable the search power of CAFS to be applied to existing programs, but without requiring any change in their code. Such programs are those which the user either does not wish to change or is unable to change. The former category includes operational programs which have been well run-in, many of them regular reporting and analysis tasks, as well as the year-end undocumented antiques of apocryphal importance and complexity to be found in every installation. The latter category includes packages that were developed before CAFS became available, and parts of applications generated by QuickBuild.

CSO is addressed to the classic batch type of COBOL program which reads serially through the whole of a file, performs some selection process, and then processes the hit records. When such a program is enhanced by CSO it still acts as though it were reading the whole file, but in fact the file is searched by CAFS and only the hit records are passed to the program.

CSO exploits a limited subset of the CAFS selection functions. Boolean logic is permitted, but quorum logic and inter-field comparisons are beyond its scope. No subsetting of hit records by the Retrieval Unit can be invoked; it would in any case be inappropriate to use it, because it would necessitate changing the existing program. No Retrieval Processor functions are exploited. CSO can be used to search simple files, and is unable to search an IDMSX database; but the majority of the programs for which it is intended were written before IDMSX was implemented. Only those programs which use the COBOL run-time library can be accelerated by CSO.

The CSO interface takes the form of a single SCL command: SET_CAFS_CRITERIA. The parameters to this command define the layout of the target records and specify the simple selection criteria. The objective of allowing the program to run without change is achieved by invoking the CAFS search by SCL external to the program, rather than from within the program itself.

The reduction in elapsed time and use of system resources is governed entirely by the hit-rate on the file; the smaller the hit-rate, the greater the improvement. No complicated evaluation of the likely hit-rate is needed in order to establish when CSO should and should not be used. The interface is so simple that users are recommended to try it on all candidate programs, retaining it in subsequent operational use wherever it proves to offer substantial benefit.

5.3 The Direct CAFS Interface

DCI allows a programmer writing in any common high-level language to generate programs which exploit all the functionality of the CAFS engine. The 'Direct' in the title means that it is the programmer who decides all aspects of where and how CAFS is to be used; by contrast, RCI and QUERYMASTER contain an access optimiser which determines, independently of the user, when and to what extent CAFS searching is to be exploited.

DCI is provided as a series of procedures, all of which may be invoked by the standard VME procedure call mechanism. The details of a task are established by the parameters to these procedures. DCI expects such parameters to be passed as simple character strings, which minimises the programming effort by eliminating any need to codify the input or reduce it to rigid format; the form in which requirements are specified by an end user at an enquiry terminal are thus remarkably close to the form in which those requirements are passed to the DCI procedures.

DCI is currently the only CAFS interface through which the following features of CAFS can be exploited:

- quorum logic;
- full sub-condition counting;
- the full range of Retrieval Processor functions;
- the multiple variant of self-identifying format for text in documents.

The simplest form of DCI program includes the following basic group of procedures:

- open the target file for searching by CAFS;
- define the layout of fields within the target record-type, and specify which fields of hit records are to be retrieved;
- specify the search criteria which will identify a hit record;
- read a hit record;
- close the file when all searching has been completed.

The first time that a hit record is to be read, the codification of the CAFS task specification is completed, the CAFS search is initiated, hit records are returned to a mainframe buffer managed by VME, and the first such record is passed to the application for processing; subsequent calls extract further hit records from the buffer; if the buffer fills with hit records, VME suspends the CAFS search until they have been processed; if the application empties the hit buffer VME initiates a continuation CAFS search; to the application program the hit buffer appears as a simple serial file from which records can be successively read. The process of defining search criteria and reading hit records may be repeated as often as desired for successive enquiries during the same run of the program.

Other DCI procedures are provided to cover more advanced uses:

- distinguishing between initiation procedures and main processing;
- calling retrieval processor functions;
- specifying counts on chosen conditions and groups of conditions;
- saving and restoring, to support searches which may extend over several transaction processing message pairs;
- management of text—conversion in both directions between plain text and text in self-identifying format;
- management of trailers in textual applications—defining the contextual precision of enquiries: whether the required terms are to be sought within the same sentence, the same paragraph, the same section, etc.;
- run-time control—certain additional checks and diagnostics.

At the time of writing (September 1985) DCI is only available for use with serial or indexed sequential files; however, ICL has confirmed that DCI will be extended for additional access to IDMSX databases.

DCI is one of several interfaces available for use from COBOL, but is currently the only interface available to non-COBOL programmers. To date it has been used from COBOL, FORTRAN, PASCAL, RPG2, ALGOL68, and SCL, from Application Master, Reportmaster and QUERYMASTER, and from the NCC's FILETAB and Logica's RAPPORT.

The following examples from an estate agency application illustrate how a variety of search criteria can be established during a single run of the same program, using a mixture of pre-set sub-conditions and run-time variable conditions.

Three pre-set sub-conditions are defined:

```
C1 = FEATURE EQ 'dglazing'
C2 = FEATURE EQ 'garage'
C3 = FEATURE EQ 'carpets'
```

Four typical enquiries might be:

List all houses costing less than £30,000 in Congleton:

```
PRICE LT 30000 AND TOWN EQ 'CONGLETON'
```

List all houses costing less than £30,000 in Congleton with at least two of the desirable special features:

```
PRICE LT 30000 AND TOWN EQ 'CONGLETON' AND QUORUM
    THRESHOLD 1 C1 C2 C3
```

How many properties have double-glazing or a garage or carpets?

```
C1 OR C2 OR C3 AND FALSE COUNT C1 C2 C3
```

(The 'AND FALSE' clause prevents any hit records being returned to the host, which only receives the three completed counts.)

We need a property with a garage and double-glazing; we'd take anything near Crewe; anywhere else in Cheshire it would have to be a bungalow, but it mustn't be in Poynton:

```
C1 AND C2 AND (POSTCODE = CW! OR (COUNTY EQ 'CHESHIRE'
    AND TYPE EQ 1 AND TOWN NE 'POYNTON'))
```

5.4 QUERYMASTER

QUERYMASTER is a general purpose online enquiry program, and enables end users who have little or no technical training to conduct their own enquiry sessions and explore their own data without their having to rely continually on assistance from the dp department. QUERYMASTER is able to access data held in IDMSX databases, in indexed sequential files and in serial files. It takes full advantage of any available indexes, and can navigate through a database by following pointers, as well as being able to search data with CAFS. (CAFS access to serial files becomes available with version 250.) Although the data that is searched may be held in quite complex file and database structures, it is a principal feature of QUERYMASTER that the end user sees the data in a simple 'relational' form, and all the storage complexities are concealed.

The access optimiser at the heart of QUERYMASTER selects the most efficient combination of access methods for the resolution of each enquiry. This includes the choice of the best navigational route to follow in order to access associated records. This automatic navigation capability means that the user need only consider what data he requires, and not the means by which it is to be obtained.

The user makes his enquiries through a query view, created by the VME command CREATE_QUERY_VIEW from information defined in the DDS Data Dictionary. The interposition of a query view between the user and the data has the following advantages:

- it supports the security of the stored data, by ensuring that the user can only access those records and fields to which he is entitled;
- it is the mechanism which conceals the structural complexity of the stored data, and presents the user with the simple and readily intelligible relational view;
- the formal data names used by the database administrator to avoid ambiguity in the Data Dictionary can be replaced by shorter and more immediately meaningful names which assist the user's understanding and are more efficient to use in the formulation of enquiries.

As well as accessing data, QUERYMASTER is able to display it or print it in a simple tabular form. It can create temporary files, and perform further enquiry functions upon them. Results can be sorted before output, and summary information such as totals can be generated not only at the end of output but also at intermediate control breaks. Comprehensive help facilities are provided. Parameterised macros are available to simplify the generation of repeatable enquiries, which form a significant proportion of the work of an established enquiry service.

A typical enquiry consists of a command, a list of the data items to be retrieved and displayed or printed, and a selection expression which determines which records are to be retrieved. For example:

```
LIST CUST-NAME, ORDER-NO, ORDER-DATE, QUANTITY, PRODUCT-DESC
     WHERE COUNTY STARTSWITH 'LANC'
     AND ORDER-DATE = 1.10.85 TO 31.10.85
```

5.5 The Relational CAFS Interface

RCI is a programming extension to the COBOL language, intended for use by application programmers. It follows many of the same principles as QUERYMASTER, with which it shares important modules of code such as the access optimiser. Its principal objectives are:

- to give the application programmer the benefits of a simple structures;
- to enable application programmers to write programs capable of exploiting CAFS, without requiring them to become familiar with the details of how CAFS operates and is controlled.

The programmer accesses the data through a relational view, which performs many of the same functions as a query view within QUERYMASTER. Thus it conceals the complexities of the underlying structures, and frees the programmer from the need to choose between competing access strategies. It can also implement value-based privacy constraints, supplementing the standard Codasyl item-based privacy rules.

The programmer 'sees' the required data, as provided by the relational view, in the apparent form of a simple serial file. He manipulates it with four additional COBOL verbs, which closely mimic the file-handling conventions with which he is already familiar. These are:

- START — this creates an instance of the relational view, fixing selection parameters; it is analogous to the opening of a real serial file;
- READ — delivers the next record; behind this apparently simple process the access optimiser may exploit indexes, follow IDMSX pointers, and instigate CAFS searches;
- SAVE — preserve the state of the view at the end of a TP phase;
- RESTORE — re-create the previous state of the view at the start of the next TP phase.

It will be seen that the programmer has no direct interaction with CAFS; RCI exploits CAFS searching capability wherever it is appropriate for it to do so. As an instance of this, CAFS can be used to search text without the programmer needing to get involved in the niceties of the process.

RCI can be used in free-standing COBOL programs. It can also be used by COBOL modules incorporated in, for example, dialogues generated by Application Master. Whereas QUERYMASTER only provides CAFS access in a MAC environment, RCI enables the benefits of CAFS to be extended to TP and batch services.

RCI has important effects on programmer effectiveness: not only are the programs, through their use of CAFS, more efficient in operation, but also, because the interface undertakes the management of all the complexities of database access, programmer productivity is considerably enhanced. As an instance of this, one user quotes an enquiry application, developed using Application Master, COBOL, and RCI in 13 man days, which in unassisted COBOL it had been estimated would require 2 man years.

5.6 The Relational System

ICL's relational system was not designed exclusively for CAFS. Its objectives and methods were established in the knowledge of parallel work on data-models and CAFS 800, but before the appearance of the current CAFS engine. A common emphasis on data retrieval and the ability to search existing data provided the basis for the successful integration of the relational CAFS engine and the relational software.

The complete relational system is provided as three separate software products: the Personal Database System PDS, the end-user enquiry language QUERYMASTER, and the Relational CAFS Interface.

PDS is classified here as part of the CAFS context rather than as a member of the CAFS system: PDS data is searchable by CAFS, but the PDS software itself does not invoke CAFS searching functions.

The objectives of the relational system are:

- to widen the community of data users;
- to integrate with existing data management products and preferred application development methodologies;
- to retain the flexibility to exploit new techniques as they become available.

The aim of providing effective access to a wider community implies that the data should be presented to users in a simple and uniform way and that the technical demands on the user should be reduced to a minimum. Low-level details, such as the organisation of data storage, are of real interest only to the professional minority whose task it is to provide a relational data access platform to the majority.

The relational system data model presents to the users a tabular data structure without visible navigational links between tables. The tables accessed by the terminal user or the COBOL programmer are the results of using the relational operators SELECT, PROJECT, and JOIN on underlying physical structures.

The relational products are integrated with the Data Dictionary, which holds the formal definition of all data, documents the rules for data access by each distinguishable group of users, and avoids the need for repeated description of the format and storage organisation of data. The availability of CAFS has introduced further DDS definitions, to document the encoding of text in self-identifying format, and to facilitate CAFS searching of IDMSX areas.

The relational system addresses physical data in both IDMSX and indexed sequential and serial formats. By associating a high-level relational definition of a logical database with a CODASYL definition of physical structures, it combines the benefits of the simplicity of the relational approach and the efficiency of IDMSX.

5.7 The CAFS Context

The CAFS subsystem forms part of a wider system, comprising the following elements:

- the operating system—VME;
- the physical data management systems—PDS, RECMAN, and IDMSX;
- the man-machine interfaces—languages, forms, viewdata;
- the system development control and documentation mechanism—DDS.

VME provides support for the CAFS system by managing resources at the record (RECMAN and IDMSX) and physical magnetic media (MAMPHY) levels. VME also provides job control, operator control, and monitoring facilities associated with CAFS usage, and contains the CAFS code generator which codifies each CAFS task specification.

The range of CAFS-searchable standard file types includes all the most common ones, including serial, ordered-serial, indexed sequential, and hashed-random, but excludes files containing spanned records or non-embedded keys. The primary key of an indexed sequential file can be used to focus a CAFS search onto a small range within the file. This facility implies that the primary key should be chosen not only to distinguish one record from another but also to provide the most useful physical clustering of associated records within the file.

IDMSX databases are made CAFS-searchable on an area by area basis. The relevant areas are reformatted to a CAFS-searchable format by a single-pass process of resequencing the order of information within the page. The selective retrieval function of CAFS can be used to convert physical IDMSX records into subrecords as described in an IDMSX subschema; there is therefore a strong argument for tailoring IDMSX subschemas to anticipated CAFS searches.

The range of man-machine interfaces is wide. CAFS has hitherto been exploited from systems employing flexible online dialogues, screen-based form-filling, and viewdata. The underlying facilities make CAFS accessible through whatever medium is deemed appropriate for a particular application.

It is commonly recognised that the activities of system development need to be co-ordinated around a central model of the host organisation and its computer systems. The Data Dictionary system DDS continues to be the key ICL component for modelling and controlling all phases of system development.

5.8 Connectivity, co-existence and performance

The trend has been to increase mainframe to disc-channel connectivity. This increases the accessibility of the data and reduces the performance-interference between separate processes. Greater mainframe-CAFS connectivity also implies that a major search-task can be syndicated to a larger battery of CAFS engines working in concert; 10 engines can search about 25 Mbytes per second.

The CAFS engine attaches to the DCM on 296-family machines—2953, 2957, 2958, 2966 and 2988—and on Series 39. Single OCP 2900s can connect to 6 CAFS engines; dual and superdual configurations can connect to 8 engines. On Series 39 the connectivity is greater. Although there are some restrictions on early systems, in principle each node can support several Macrolans, and each Macrolan can support up to 15 controllers. Within the overall constraints of VME on the size of backing storage and the number of storage devices that can be supported, there is considerable freedom. Thus a single node Level 30 could support 36 CAFS engines, and a single node Level 80 could support 72. Series 39 systems therefore can be configured to handle whatever level of CAFS searching is required by the planned workload.

The Series 39 DCM is the High-Speed Disc Controller (HSDC) and connects to one CAFS engine. On 2900, the DCU/2 disc control unit and more commonly the DSC unit connect to two CAFS-compatable DCMS.

The maximum disc-drive string on the various controllers are:

HSDC	–	8 × FDS300 or 4 × FDS2500 or 16 MDSS 'retained' drives MDSS drives are EDS80s, FDS160s or FDS640s
DCU/2 DCM	–	16 MDSS drives
DSC DCM	–	32 MDSS drives

CAFS attaches to standard DCMS working with standard discs. It is also the case that CAFS searches most standard RECMAN files, PDS databases and IDMSX databases. In the case of IDMSX, a single-pass reformat procedure sets up areas of the database for CAFS searching.

Coexistence objectives require that a 'long' CAFS search should be interruptible by a 'short' transaction-processing task. It would probably be undesirable for a single-record fetch to queue behind a 40-track full-cylinder CAFS scan on an FDS640. VME therefore fragments CAFS searches, each fragment searching consecutive blocks on a disc-cylinder and being no longer than a system-parameter defined number of tracks.

On 2966s etc., the maximum-fragment parameter default of 10 tracks can be changed at system setup time to any value. The value must be chosen to balance the needs of the existing workload against the need to exploit CAFS searching.

On Series 39, multi-block fetches and CAFS searches travel second-class; single-block fetches travel first class. The HSDC exercises the right to pre-empt long tasks on the disc-channel when a short task arrives. Given this degree of HSDC intelligence, the maximum-fragment parameter is unnecessary.

CAFS can search data at some 3.6Mb/sec, outrunning the delivery-rate of the fastest FDS2500 drives. In practice, therefore, the following parameters always affect the data-delivery speed of a CAFS engine:

- *ds* – the maximum formatted-data search rate of the disc-drive
- *bf* – the blocking-factor; block-size choice effect on search-rate
- *ff* – the fragment-factor; governed by the temporal dissection of the CAFS-search into search-fragments
- *pf* – packing-factor; the proportion of the data-blocks occupied by the records relevant to the CAFS search

The basic upper limit on searching-speed is therefore:

File-search rate	$ds \times bf \times ff$ Mbytes/sec
Data-search rate	$ds \times bf \times ff \times pf$ Mbytes/sec

Other factors such as disc-head movement, rotational latency, file-fragmentation, buffer management and process-multiplexing all subtract from the data-rate as perceived by the application program or the end-user. Note however that these aspects of performance are standard and preceded the introduction of CAFS.

The disc-speeds *ds* of the drives are 1.17696 Mb/sec (MDSS), 2.22910 Mb/sec (FDS300) and 2.83277 Mb/sec (FDS2500).

The blocking-factor *bf* is also specific to the drive concerned. Below are listed *bf*s for the three drives where the block-size is chosen as Nk bytes or as a maximal value for that number of blocks/track:

MDSS Disc drives				**FDS300 Disc drives**				**FDS2500 Disc drives**			
Block size (bytes)	*Blocks per track*	*Bytes per track*	*Blocking factor*	*Block size (bytes)*	*Blocks per track*	*Bytes per track*	*Blocking factor*	*Block size (bytes)*	*Blocks per track*	*Bytes per track*	*Blocking factor*
2048	9	18432	0·939641	2048	15	30720	0·804525	2048	18	36864	0·785142
3072	6	18432	0·939641	3072	10	30720	0·804525	3072	13	39936	0·850570
4096	4	16384	0·835237	4096	8	32768	0·858160	4096	10	40960	0·872380
6144	3	18432	0·939641	6144	5	30720	0·804525	6144	7	43008	0·915999*
9216	2	18432	0·939641*	9216	4	36864	0·965431*	19216	4	36864	0·785142
18432	1	18432	0·939641	18432	2	36864	0·965431	18432	2	36864	0·785142
2057	9	18513	0·943770	2100	15	31500	0·824953	2164	18	38952	0·829163
3155	6	18930	0·965029	3412	10	34120	0·893568	3188	13	41444	0·882688
4801	4	19204	0·978997	4404	8	35232	0·922690	4276	10	42760	0·910717
6447	3	19341	0·985981	7316	5	36580	0·957993	6356	7	44492	0·947606
9739	2	19478	0·992965	9300	4	37200	0·974230	11476	4	45904	0·977679
19616	1	19616	1·000000	19092	2	38184	1·000000	23476	2	46952	1·000000

Blocking-factors are more significant on the faster drives and the best block-sizes vary from drive to drive.

The Fragment Factor (*ff*) reflects the fact that the CAFS search is fragmented by VME. The inter-fragment overhead is typically 2 disc-revolutions or some 33 msec:

$$ff = (\text{max fragment size}) \ / \ (\text{max fragment size} + 2)$$

Taking into account the parameters *ds*, *bf* and *ff* and adopting the asterisked block-sizes, we can calculate 'typical' effective CAFS file-search speeds as 0.922 Mb/sec (MDSS), 1.793 Mb/sec (FDS300) and 2.290 Mb/sec (FDS2500).

The last parameter to be discussed is the data packing-factor *pf*. Data is not usually 100%-packed in a file for many reasons. Varieties of red-tape accompany the object-data, several record-types may coexist in a file and dynamic data should be packed at lower densities to avoid overflow.

Low packing-densities have a proportionate effect on the data-delivery rate of CAFS but this does not mean that they subtract from the value of CAFS. Unused file-space and non-target data-types are the first examples of CAFS' effectiveness in filtering out disc-space which is irrelevant to the search and to any subsequent processing.

The Series 39 figures below indicate retrieval processor times for various tasks. In the worst case of short records and a high hit-rate, the RP component of CAFS will not keep pace with the search-rate at the CAFS front-end.

130	μsec	= overhead per hit-record (260 μsec on 2966s etc.)
(L+8) / 3.5	μsec	= output-transfer time for the L-byte record
87	μsec	= overhead per function call
200	μsec	= function adding totalling on an 8-byte field
250	μsec	= function comparing 2 2-byte fields
500	μsec	= function comparing 2 20-byte fields

When the high hit-rate is a local phenomenon, a cluster of records satisfying the LC condition, the buffering within CAFS helps to maintain the output performance of the engine.

6 Using CAFS

6.1 Introduction

CAFS can be made use of in a data processing installation with both existing and new systems and data files. In both cases the right environment has to be created in order to exploit the CAFS system.This section outlines how existing systems can be amended to use CAFS and the analysis and design implications of CAFS on new developments.

In designing new systems to utilise CAFS, a central problem to be addressed is the degree of data normalisation that should be introduced into the record structure. CAFS is best suited to large unnormalised records, where related data items are held in the same record. However traditional processing systems are most flexible when designed with fully normalised records. There is no simple answer to this design dichotomy, and each case must be resolved on its own merits.

6.2 Existing systems and data files

CAFS can be used with existing systems where there is a demand for on-line enquiries or extensive ad hoc reports from existing data files, which have not been designed to support the required access paths. All the conditions and observations in this section apply also in the case of new systems. The content of this section, apart from the mandatory changes, is determined by a judgement of what is worthwhile. This involves an assessment of the value and difficulty of the listed tasks. The judgement of this report is of necessity qualitative; individual sites can make a more quantified decision.

Some changes to data are not independent of existing valued code or may materially affect the performance of existing systems. It is likely that these changes would be regarded as too strategic, i.e. too difficult, for implementation on existing data.

6.2.1 The decision to CAFS search

The CAFS facility offers a realistic choice between searching the operational data in situ or searching an alternative, possibly subsetted, copy. In some cases it is necessary to search the 'live' data for reasons of objectivity or performance; in other cases, the use of the data calls for a 'snapshot' of the live data. In the third category of cases, a decision has to be made between 'in situ' and 'offline' searching.

The current use of the data and access requirements to it are the major factors affecting this decision.

Imagine a high volume TP system supported by object data and operating at a peak period. The introduction of CAFS searching on that data could overload disc channels and extend terminal response times unacceptably. This extreme example indicates the issues involved.

If it is not possible to CAFS search the data in situ, several alternatives are available.

6.2.2 Mandatory changes

Once the decision has been made to CAFS search a given file, that file must meet the following requirements.

All DME files must be converted to VME files. This can be achieved using an ADRAM conversion aid. An audit of the data on the VME side after conversion will be assisted by CAFS.

VME files may not be accessible by CAFS if they are on the wrong volume, disc drive or channel. They can be addressed by CAFS by attaching disc drives to a CAFS-capable disc channel or by mounting exchangeable disc volumes on a drive connected to a CAFS-capable channel. Files held on old discs, e.g. EDS200—must be copied to discs which are CAFS-searchable. CAFS is able to search IDMSX databases, and also ISAM and Serial files. Hashed random files are treated as a subset of ISAM files, and are therefore CAFS-searchable. Some relatively rare file structures, e.g. those where records do not contain their own keys, are not CAFS-searchable. Therefore, any such files will require a format conversion. IDMSX and ISAM data must be held in a specific CAFS-searchable form; hence, areas of existing databases may require conversion to the searchable page format. This is a one-off job achieved by an IDMSX dump/restore procedure. The software environment must be that defined in Chapter 5. The earliest versions of relevant software products which will support CAFS working are VME8.01, DDS.650, QUERYMASTER.210 and IDMSX.350.

6.2.3 Recommended considerations—general

These changes are not mandatory and are suggested in the cause of improving the performance or applicability of CAFS.

A file which is fragmented into a number of file extents should be mapped into as few extents as possible, as CAFS searches are interrupted by fragmentation. Disc volumes with faulty tracks may increase the number of file extents. Obviously, discs with no faulty tracks are preferred. If there are faulty tracks, these should be marked as used rather than replaced by alternative tracks.

Data not searched by CAFS should be allocated to non CAFS-capable disc channels as far as the requirements to balance the system load across channels allow. In particular, CAFS-searchable files and high hit-rate files should not be mixed on the same disc.

For dual access disc drives, with CAFS available only on one route, the first attempted CAFS read to a file will automatically cause an attempted change to the CAFS supported route. On super-dual systems this will not cause a problem, as CAFS requests will be routed via the node with the CAFS disc controller. All other requests will take the shortest available route from the processing node in which they originate.

On other systems, however, there could be a problem. If the disc load is balanced across controllers at the start of the day, any CAFS requests to files on a non-CAFS primary route will cause that route to be changed. In an extreme case, this could lead to all the discs being accessed through the CAFS controller, whilst the non-CAFS controller is idle.

6.2.4 Recommended considerations—ISAM systems

A CAFS search on the main body of an ISAM file will be supplemented by a separate search on each required block of the overflow area. This is less efficient than a single CAFS search on the same number of blocks as it increases the OCP resource and elapsed time required.

CAFS searching is therefore a reason for reducing the percentage of records in the overflow area. This can be done by reorganising the file more frequently and/or lowering the packing density.

6.2.5 Recommended considerations—IDMSX systems

CAFS complements the access methods defined by CODASYL. Physical database design is geared to the high volume use of the data. It is therefore possible that the lower volume uses of the data would be better implemented via CAFS access than via IDMSX navigation.

IDMSX records of a particular type are typically less densely packed in an area than records in an ISAM file. The CAFS data delivery rate will therefore be lower. However, to measure the usefulness of CAFS in terms of its delivery rate is incorrect. This should be measured in terms of the reduction in elapsed time for a query and in terms of the reduction in resource requirement on OCP and disc channels.

These points cover the decision to CAFS search the IDMSX database, duplexing, relocated/fragmented records, the AREA-to-channel mapping, database procedures and IDMSX-restructure.

In-situ enquiry

If CAFS is to be used to search an IDMSX area, the page of that area must be held in searchable format. Conversion of existing databases is a linear one-off process but there is a continuing run-time overhead for non-CAFS IDMSX processing.

Database pages are converted between the old 'non-CAFS' and the new 'SEARCHABLE' formats every time they move between disc and store. If there are 'R' records on the page, the costs in machine instructions are:

40 + 30R PLI for reading the page; 40 + 40R PLI for writing

To put these figures into context, the 'average' IDMSX DML statement uses some 1000 PLI in IDMSX.350.

The decision to allow CAFS searching is made for each IDMSX area.

A further consideration is that CAFS searching is done with area level locking or with no locks set. The former may be incompatible with other uses of the data and the latter may be unacceptable.

The alternatives to free in-situ enquiry include:

- snapshotting to ISAM files
- duplexing the database
- keeping a restored IDMSX dump online for CAFS and fast recovery
- scheduling queries in quiescent periods

Duplexing

The obvious way to handle an increased workload is to double the number of access channels. The needs of resilience and increased accessibility are simultaneously served by siting the AREA plexes on different disc channels.

The usefulness of duplexing increases as the percentage of read-only activity increases; CAFS searching is a read-only activity and uses only one plex. Both plexes are in the same page format, either SEARCHABLE or not. IDMSX normally flip-flops accesses to the two plexes but if a CAFS search is proceeding on one plex, IDMSX directs requests to the other plex.

The system does not prevent CAFS searching proceeding on both plexes simultaneously. The question does arise therefore as to whether both plexes should be on CAFS-capable drives. If CAFS is only available on one plex, CAFS requests are queued on that plex, leaving the channel open for non-CAFS TP access.

Duplexing is only available with IDMSX as are several other facilities useful in the CAFS/QUERYMASTER context. These include page-extend, secondary indexes and multiple areas for a single record-type.

Relocated and fragmented records

These cannot be recognised as hit records in a CAFS search. QUERYMASTER will either comment when it is told of the offending record or, if a standard modification is made to the software, will comment the first time it sees such a record.

IDMSX index records can be relocated at run-time but these records are not currently CAFS-searchable. Other relocated records can only be created by IDMSX-

RESTRUCTURE and are removed by IDMSX-REORGANISE. IDMSX.400 RESTRUCTURE available in SV211 will avoid creating and will remove relocated records.

Fragmented records can be either fixed-length or variable-length.

Fixed-length records can only be fragmented by STORE if there is a 'MINIMUM ROOT' or 'MINIMUM FRAGMENT' clause in the IDMSX STORAGE-SCHEMA. The fragmentation is removed by IDMSX-REORGANISE by omitting such clauses.

Variable-length records for which 'MINIMUM ROOT IS RECORD LENGTH' is defined in the STORAGE-SCHEMA will not be fragmented when first stored. They may be fragmented if lengthened. IDMSX-REORGANISE, and at IDMSX.400 IDMSX-RESTRUCTURE, will re-STORE them and therefore remove the fragmentation. IDMSX databases should therefore be monitored for the presence of fragmentation.

Fragmentation can usually be avoided by using ERASE and STORE commands, rather than a MODIFY command.

AREA-to-channel mapping

The distribution of IDMSX AREAS across disc channels may be affected by the new traffic pattern on the AREAS and by the distinction between CAFS-searchable AREAS and non CAFS-searchable AREAS.

Queries may usefully be syndicated to a number of CAFS engines. IDMSX enables records of one type to be stored in multiple AREAS which may be accessed by multiple CAFS units. Eight CAFS engines working in parallel give almost eight times the search speed because the lightly loaded OCP can take eight times the work without becoming a bottleneck.

Each AREA storing the target record-type should be on its own CAFS-capable disc channel. IDMSX subschemas must be tailored to each AREA to prevent QUERYMASTER searching other AREAS in sequence. A QUERYMASTER AVM must be set up to search each AREA.

The query must then be syndicated from a file using say the VME event mechanism; answers from each component-search will be consolidated using the event mechanism again. This requires some work on the part of the systems management and is not an intrinsic part of the current CAFS system. It also requires that the user file his enquiry rather than operate online and so is perhaps restricted to enquiries of a 'production' nature.

Syndication is a prime example of distributed processing. The query is distributed to a number of CAFS engines. Note that requests for AVERAGES do not distribute correctly as an AVERAGE of AVERAGES is unlikely to be correct.

Database procedures

QUERYMASTER will invoke 'after-GET' database-procedures in order to enforce the privacy requirement in a CAFS-independent way.

Compressed and encrypted data cannot be searched by CAFS. If 'after-GET' procedures are used to decompress and decrypt then the record types are no longer CAFS-searchable. Other 'after-GET' procedures which change data formats may or may not remain valid in a CAFS context.

Procedures for privacy, security and audit which do not change the data format are fully compatible with CAFS searching.

IDMSX-RESTRUCTURE

An IDMSX-RESTRUCTURE is a less complicated task than an IDMSX-REORGANISE. With the exception of recommending the removal of relocated/fragmented records, section 6.2 considers only changes brought about by IDMSX-RESTRUCTURE.

Data and database structure may be subtracted, added or changed.

The removal of features of an IDMSX database implies that existing affected programs should be changed or discarded. Superchief records, access infrastructure, especially record/set indexes, and derived data are all candidates for removal. This will particularly be the case if the database design was comprehensively enthusiastic rather than basic and prototype.

A skeletal database design serving only basic requirements can be enhanced to facilitate both QUERYMASTER and CAFS. QUERYMASTER is helped by the presence of record indexes and by OWNER-pointers. CAFS is helped if the number of record-types required for a query can be reduced by by duplicating the data. This last technique must be used with care and is discussed below (§6.6.2).

6.2.6 Recommended considerations—PDS Systems

PDS does not invoke CAFS itself. However, if the PDS database is defined in the Data Dictionary it can be accessed by QUERYMASTER, which will invoke CAFS. This technique can be used to create a Queryview consisting of data held in PDS and data held in conventional sytems. Currently the PDS data descriptions must be entered manually into DDS, using the output from a PDS RENEW session.

Once these definitions have been entered into the Data Dictionary, a Queryview may be defined for the PDS database in the usual way. This will enable the Queryviews to be set up for the database, and consequently CAFS usage directly on it. In the same way, RCI and CAFS may be used on the database for use with other systems, using COBOL.

If this approach is to succeed, a PDS Database Controller must be appointed to ensure the consistency of data on the Data Dictionary. PDS databases can be highly variable, with new tables defined and old tables deleted without reference to the DDS. There is clearly a need for software which will ensure consistency of DDS and PDS if the use of CAFS on PDS is to become widespread.

6.3 The evolution of existing systems

Existing data systems are developed by the addition of new data and new data subject areas. QUERYMASTER contributes to easier development by removing the distinction between IDMSX and ISAM held data. QUERYMASTER and RCI can access both IDMSX and ISAM files within a single query view. The value-based set mechanism can be used to link data from many different databases and files. Extensions to existing IDMSX-based systems may therefore be prototyped, initially by using ISAM.

The most significant new data-type is text. CAFS, potentially, offers an even greater performance boost in this context than it does for conventional data.

6.4 Future systems—general

Future systems and data files should be designed with CAFS usage in mind if CAFS is considered to be an effective method of implementing specific business requirements. Before evaluating the design implications of CAFS in future systems development it is useful to classify systems as follows in order to assess the likely usefulness of CAFS.

Batch Systems These require significant data searching for non-key items.

High Volume TP systems CAFS is unlikely to be useful as access is probably by pre-defined keys. The very high volume of operational transactions may well overwhelm a battery of CAFS engines if such keys are not implemented.

Low Volume Online Systems CAFS is very likely to be of significant benefit for these types of systems.

Office and Text Systems In the future CAFS is likely to be of use in these types of systems.

In addition two further systems implications of CAFS need to be considered when designing new systems. The first is that CAFS may move more systems from batch to TP environments. This is likely for some ad hoc management information systems. The second is that CAFS will give the opportunity to develop some new systems (e.g. text systems) that previously would not have been either cost effective or timely.

6.5 Future ISAM systems

CAFS should prove especially useful for developing systems around large volume ISAM files. Although these files should support large-volume, pre-defined, regular enquiries, identified during the analysis phase of a project, there will be many management information requests which would be suited to the use of CAFS processing. Obviously if search criteria can be satisfied via primary or secondary keys then CAFS should not be used.

Unlike IDMSX records, ISAM records cannot be fragmented, and files can be searched by primary or alternate keys. CAFS searches can be restricted to a part of the file by using the primary key.

6.5.1 Prime-key optimisation

As stated above, the prime-key serves two purposes. It traditionally provides fast access to specific records. Secondly, it is used if possible to limit the extent of a CAFS search. This second purpose may weigh in the design of the primary key format.

For efficient 'focussing' of CAFS searches it is advisable to use a primary key sequence which ensures that records which will be required together are as far as possible, stored together. Such a sequence makes use of a 'clustering' key. In such a case the 'record-identifying' key will probably have to be used as an alternate key.

6.5.2 Variable-length records

CAFS will always be able to perform searches on the fixed-length part of the data as these records are never fragmented in ISAM-files. However, several caveats still remain with respect to variable-length records.

Lengthening records may well have a severe effect on the amount of overflow in the file and slow down the CAFS search.

QUERYMASTER does not use the CAFS counting hardware when counting variable-length records.

QUERYMASTER is able to search variable length records containing fields of the following types only:

- a single variable length character field;
- a variable length array consisting of a variable number of repeats of a single, fixed length, field;
- a field containing text in self-identifying format.

6.6 Future IDMSX Systems

The IDMSX design process is affected by the availability of CAFS, by the availability of Application Master, Reportmaster, QUERYMASTER, RCI and also by the need to reorganise the database from time to time. As the software improves in its ability to handle IDMSX databases, its effect on the design process should diminish. Nevertheless, there will for some time presumably be some residual set of IDMSX features whose exploitation can reduce the applicability of other products.

There are certain restrictions to be noted when designing a database for use with

CAFS. Compressed and encrypted fields, and fragmented and re-located records cannot be handled by CAFS. Such records are ignored, with CAFS reporting that it has met the record.

The topical goals of information systems and the recent availability of new production tools are changing the objectives of the design process. Simpler database designs are likely to serve many of these new objectives.

6.6.1 Analysis and logical design implications

IDMSX supports pre-defined regular enquiries, identified during the analysis phase of a project. These are then expedited by the database design. One of the major limitations of this approach in the past has been the inability to predict enquiries of an unspecified and ad hoc nature. This aspect has traditionally either taken up much analysis time or has been ignored.

In anticipating the use of CAFS/QUERYMASTER, the analysis phase need only consider the main business processes in detail. The management information questions need not be determined in advance. However, in order to achieve this situation, there are several factors which must be considered during analysis and design.

In principle data analysis and data design should be simplified, since, although the same data is involved, fewer processes need to be considered. This should lead to the situation where the logical schema design will closely mirror the entity model produced during data analysis. This should result in fewer record and set-types.

In most database systems, the implemented schema contains record-types and sets that do not represent entities and relationships in the data model or in the real world. Typical reasons for this are:

- the need to access data by multiple keys;
- the need for data-subset totals often causing maintenance problems;
- the need to shorten search-paths through large unsorted sets;
- the need to hold superchief records as access points.

CAFS is likely to remove the need for some of these extra record-types as it effectively replaces IDMSX navigation by searching. CAFS returns subschema records rather than schema records, thus eliminating unwanted data and reducing OCP time.

In a traditional database design, several set-types are often implemented purely to satisfy low-volume, ad hoc enquiries. Again, CAFS may eliminate the need for them by substituting for IDMSX navigation.

By simplifying IDMSX data structures, the read-only CAFS unit thus curiously contributes to performance on the update side.

6.6.2 Physical-database design implications

Physical clustering

CAFS searches records of one record-type. These will be stored in one IDMSX area and possibly several IDMSX areas. Within an area, records will reside in a specific page range, if this is defined, and in the secondary overflow space.

CAFS searches can be limited to less than a full area if page ranges and overflow areas are consciously defined rather than by default. More thought should be given to the segregation of different types of data which need not be clustered together.

An archive policy separating ‘historical’ from ‘live’ data contributes to the reduction of area, page range and CAFS search sizes. Similarly, the splitting of one record-type into ‘heavily used’ and ‘lightly used’ data can reduce searching times.

Data-clustering

The traditional aim of clustering has been to minimise the number of physical page transfers during IDMSX database processing by, for example, ensuring that order detail records are held on the same page as their order header record.

If CAFS is used to search the order header records in such a database the efficiency of the search will be low because so much time is spent scanning the more numerous, interspersed order detail records which are held on the same pages.

There may be a case, therefore, for changing the attitude to clustering. If the order header records are clustered in an area of their own, CAFS searches of them will be much more efficient. Such a gain must be weighed against the loss of efficiency in IDMSX processing of order information.

Data-duplication

Databases were adopted among other reasons to enable consistent, topical and trustworthy data to be held on file. The typical physical duplication of data in many 'conventional' files had previously contributed in large measure towards the removal of these three key data properties.

It is thought, therefore, that the objective of database storage is to eliminate the physical duplication of data. It is not. The objective was, and is, to guarantee, or at least expedite, the integrity of the data.

Third normal form analysis of the data removes any unnecessarily duplicated data. Curiously though, the relational model relies on the duplication of key data to relate data in one table with data in another.

Databases are not, in practice, implemented at the physical level in strict third normal form. Performance considerations dictate otherwise and CAFS may suggest further departure from TNF.

CAFS performs selection on physical rather than virtual records. There is therefore some advantage to be gained by propagating identifying indexes, alternative keys and other important data down IDMSX hierarchies.

This data is likely to be static or slow-moving but clearly the integrity of the data is threatened. Strict control must be exercised over programs which change physically duplicated data as there is no built-in IDMSX mechanism for 'simultaneously' and automatically changing physical copies of a data item in a manner transparent to the programmer.

Controls provided by DDS and the subschema mechanism must be used fully.

Derived Data

Derived data, such as counts, totals and averages, can be derived either in line or on demand. CAFS may well enable the latter method to replace the former.

CAFS can also be used effectively for integrity checking and auditing the database, again removing the number of functions which need to be designed and programmed into the core system.

6.6.3 Implications of CAFS on locking

CAFS not only does no locking, but actually ignores any existing page locks. This can lead to the situation whereby data may be out-of-date or logically inconsistent, if the database is in the process of being updated by, say, a TP system. This obviously does not apply in a read-only service. Other methods of overcoming this are either to use extract files for TP scans or to duplex the data.

The problem may be made less acute if the CAFS search fragment size, i.e. the number of tracks per search, is reduced from its default of 10. Each case must be looked at individually, but the problems must be borne in mind. The situation is slightly different on the Series 39 machines. Here, CAFS searches are interruptable, by assigning priorities; the fragment size therefore becomes irrelevant.

6.7 Text systems

Many DP applications are characterised by the presence of simple text, though this is not always apparent. Such information as product descriptions, names and addresses, delivery instructions, incident reports and so forth are often simply tagged onto a conventional fixed-format record. Because of the limitations of COBOL and similar languages, such data remains essentially passive: it is carried around with the record during processing, and may be displayed or retrieved with it, but it cannot actually be processed itself in any detail.

With CAFS-ISP, however, every word within such fragments of text can be distinguished and searched for, so that the content of the text becomes meaningful to the application as well as to the end user.

To use this facility, the text must be stored in a special format, known as Self Identifying Format (SIF) or 'signalled text'. Data to be stored as text must therefore be converted before it is written to disc, and converted back to legible format when it is displayed or printed out. The conversion process is, however, usually invisible to the end user. There are two commonly-used forms of SIF, known as 'single' and 'multiple', of which QUERYMASTER and RCI can process only the former.

6.7.1 QUERYMASTER and RCI text facilities

QUERYMASTER can process text items written in 'single' SIF form only. Each such item contains a variable number of tokens (strings such as words or punctuation sequences), each of which is prefixed by a pair of identifier bytes and a single length byte. It is conventional to use odd-numbered identifiers for punctuation sequences. Tokens with several different identifiers may be used within one SIF item.

In a comprehensively coded name and address record, one might use the following identifiers: 24 for Initials, 28 for Title, 32 for Surname, 40 for Address and 48 for Post code. Thus the address

Prof. J. W. Branestawm,
Excelsior House,
292 – 298 Anglepoise Way,
Chorley CH1 4NQ

would then be stored as a byte string of the following form:

28 4 Prof 24 1 J 24 1 W 32 10 Branestawm 33 1 ,
40 9 Excelsior 40 5 House 41 1 ,
40 7 292 – 298 40 10 Anglepoise 40 3 Way 33 1 ,
40 7 Chorley 48 3 CH1 48 3 4NQ

Text stored in this format evidently requires some extra storage space; the overhead is not however as great as it may appear, given that punctuation sequences consisting of single blanks (by far the most common case in running text) are removed entirely from single SIF records. When text is not stored in SIF form, it is usually held in fixed format fields with many trailing blanks, the storage overheads of which can be equally significant.

Two routines are available with QUERYMASTER to convert records into single SIF format. The first, MOVE TO CAFS operates one token at a time, while the second (TEXT LEGIBLE TO CAFS) operates on a string of text, which is divided into tokens using a user-supplied list of separating characters. In both cases, the user must specify the identifier to be used for the resulting SIF. The routines require the ability to pass parameters 'By Value' which makes them difficult to use other than from within a COBOL program. When SIF records are recovered by QUERYMASTER, conversion back into legible format is carried out automatically.

A record such as the above example would be documented in DDS as follows:

```
RECORD PERSON
*STRUCTURE  ITEM PERS-KEY
            ITEM SIF-LENGTH
            ITEM SIF-TEXT OCCURS 1 TO 512
                DEPENDING ON SIF-LENGTH
*SECTION FREE-TEXT
            REDEFINES SIF-TEXT

SECTION FREE-TEXT
*TEXT-STRUCTURE   ITEM NAME-TITLE         IDENTIFIER 28
                  ITEM NAME-INITIALS      IDENTIFIER 24
                  ITEM NAME-SURNAME       IDENTIFIER 32
                  ITEM ADDRESS            IDENTIFIER 40
                  ITEM POST-CODE          IDENTIFIER 48

ITEM SIF-TEXT
*PICTURE X

ITEM NAME-TITLE
*TYPE TEXT
*PICTURE X(4)
```

Similar declarations are needed for all the items in the TEXT-STRUCTURE property of the FREE-TEXT section: the length specified by their *PICTURE property is of course arbitrary, but the *TYPE property is mandatory. In this example, an OCCURS clause is used to define the maximum size of the free text portion of each record. Alternatively, if the records contained only fixed-length items for ease of processing then the SIF-TEXT should be filled out to the maximum with LOW-VALUES (X00). In either case only the used portion will be displayed by QUERYMASTER. Once this definition has been completed and compiled into a query view, the user may use the power of CAFS to search for individual words of a specific category anywhere within the text fields of these records. For example, to find full details of people with initials 'J', whose address contains the word 'House' anywhere, the user types:

```
LIST FREE-TEXT-BUFFER WHERE ADDRESS EQ 'HOUSE'
    AND INITIALS EQ 'J'
```

As indicated here, 'EQ' means 'contains any word equal to'. Similarly, 'STARTSWITH' when applied to a SIF item means 'contains any word starting with'. 'CONTAINS' when applied to SIF items means 'contains any word which contains' and is not CAFS-supported.

6.7.2 Advanced text facilities

A number of more sophisticated text processing facilities are available only by means of the DCI product, which provides programmers with all the facilities of the hardware. Of importance in text processing are the support offered by DCI for multiple format SIF, masked SIF identifiers and trailers.

Multiple Format SIF

The second type of SIF mentioned above, 'multiple' SIF, is more appropriate to applications in which large amounts of free text, mostly of the same category, are to be searched for individual words. One obvious application area is that of office automation and document retrieval.

In multiple format SIF, two levels of structural description are possible. At the higher level, a body of text may be divided into large items such as paragraphs or sentences, tagged with an identifier byte pair and a length byte pair, in the same way as 'single' SIF items. At the lower level, the individual tokens within 'multiple' SIF items are tagged only with their lengths. Thus, the sentence 'The cat sat on the mat.' might be stored in multiple SIF as '24 25 3 The 3 cat 3 sat 2 on 3 the 3 mat 1.'

Since the length prefix for each token effectively replaces the space character which would otherwise be there, the storage overheads of storing text in multiple SIF format are much less than for single format; in this case, the overhead is four bytes per sentence. (Unfortunately, the only routine currently provided for conversion into multiple format SIF does not optimise its use of storage space to quite this extent.)

The DCI package includes routines for conversion between legible text and either single or multiple format SIF. These routines are easier to use than those provided with QUERYMASTER, and are usable from any high level language, including SCL, but do not allow the user to re-define the characters which are to be treated as punctuation within text.

Masked Identifiers

The choice of identifier values can be of great importance in determining the range of possible enquiries. It is often the case that records contain items which may sometimes be searched separately and sometimes as members of a group of associated items. For example, a file of addresses might hold a mixture of current addresses and previous or associated addresses. If these are given identifiers differing only in their least significant bits, it becomes possible to perform searches for addresses either of a specified type or of any type, by applying a suitable bit mask to the values of the different identifiers found.

The mask to be applied, and a name for the resulting group of identifiers, are defined by the DCI data layout definition routines. This an example of the flexibility and power of the facilities offered by DCI for exploitation of the capabilities of the CAFS engine.

Trailers

Another advanced facility available only through DCI is support for that variant of SIF conventions known as 'trailers'. A trailer is a free-standing identifier, with no associated length or type bytes. Its function is to mark the end of a 'retrieval unit' within the text, that is a logical unit such as a paragraph, chapter or section. DCI routines can be used to program the CAFS engine to evaluate a search condition through the whole of any unit identified by a particular type of trailer, rather than simply within one record, as is normally the case. This is of great importance in document retrieval applications, in which the concept of a 'record' is often of little use. A search may be defined for target words co-occurring within the same sentence, or within the same paragraph, and evaluation of the search is deferred across record boundaries until a trailer of the appropriate type has been encountered.

The only restrictions on the use of trailers to delimit search units larger than the individual record are, first, that a CAFS search cannot be continued beyond a disc block boundary; and secondly, that the trailer structure must always be a hierarchic one. For example, if trailer 6 is used for chapter, and trailer 4 for sentence, every sentence must start and finish within one chapter. The first restriction implies that there must be a trailer at the end of each record, or that records must not span blocks. A third complication is that it is not advisable to mix fixed-field type items together with SIF items containing trailers in the same record.

6.7.3 Text facilities not supported

The ease of construction and sheer power of simple CAFS-based text searching applications does not mean that the system is capable of everything. There are still some basic text-searching operations with which CAFS alone cannot help. For example, CAFS offers no way of extracting a list of all the different words in a text superior to the traditional software techniques of hashing or sorting. Perhaps more surprisingly, CAFS cannot directly support fuzzy matches in which it is the left-hand end of a word which is to be ignored instead of the right, although the possible introduction of super-elastic characters would help to overcome this problem. Text prepared using complex mark-up symbols or accents which interrupt words can also cause problems when search terms are being specified.

Another example is the inability of CAFS to perform context-sensitive searches: any occurrence of a searched-for word will count as a hit, no matter where. This means not only that searches for, say, 'blind' and 'Venetian' will recover both 'Venetian blind' and 'blind Venetian', but also that CAFS searches for phrases such as 'that that' (as distinct from 'that' on its own) are impossible. However, if the context of each word is pre-defined, using a different identifier byte for the same word, this may not be a problem.

Despite these shortcomings, all of which can be overcome with sufficient software, there can be no doubt that the text-searching capabilities of the CAFS-ISP system offer DP staff radically new and dramatically efficient tools to make the best use of much data which has hitherto been of limited significance in the information processing activities of an organisation.

6.8 Future batch systems

The experience of one early CAFS user in designing a large batch system based on a 15 Gigabyte IDMSX database which is also used by TP gives some indications of the effect of CAFS on such systems.

Most of the data is duplexed and the TP service supports a high number of mainly simple transactions. Extensive use is made of RCI and QUERYMASTER leading to a simplification of the database structure and the data maintenance processes. The system has been designed to minimise the number of I/Os in the common TP transactions, and to keep all the data updated by our transaction on the same page. This leads to a packing density of 50 – 60%.

The batch system consists of some 60 processes. Each process has been designed to run against a 'hit' file which always contains the database key of the records to be processed and sometimes contains a small amount of other data. The hit files for a given day (not all processes are run daily) are produced by CAFS scans of the database. The selection of records is achieved via RCI and the program checks each hit record to see which selection criterion was met before writing to the appropriate hit file. To reduce elapsed time multiple concurrent CAFS scans can be run.

Early indications are that RCI is easy to use. Relational views can be quickly generated by renaming anything in the end-user view. Programming becomes simple with the DDS doing all the data definitions and relational views. Both RCI, suitable for high throughput TP or batch systems, and QUERYMASTER, suitably for ad hoc queries, access the same data definitions.

6.9 Using RCI

RCI gives program access to CAFS capabilities and has already been exploited by COBOL programs, by Application Master and by Reportmaster. More than one site envisages incorporating set query services in existing TP systems and providing Viewdata services using BULLETIN and CAFS. No doubt, RCI will be accessed via COBOL by programs written in other languages. RCI will enable search-tasks to be performed online which were previously not thought feasible. However, it does not allow any sorting.

RCI eases system implementation by providing additional data/program independence and by removing much of the drudgery of file-handling. This implies that existing staff will code more quickly and less experienced staff will be able to write programs more effectively. Systems will be more resilient to future changes.

RCI can be used for several different reasons. It can be used to combine the output from more than one file for direct input to Reportmaster. It is also useful as a data protection tool, in that the recipient of the information receives only the required subset of data, not all the data, as in QUERYMASTER.

RCI is a retrieval system only, but it can, in fact, be used to update a subset of data; the database keys can be returned, using DML as well as RCI in the same program; these keys can then be used in a COBOL program for updating.

The implications of RCI for database design seem at first sight to be exactly the same as those of QUERYMASTER and CAFS described above. In fact, QUERYMASTER can be used to prototype the RCI system, to check, for example, the number of hit records, and the access path. This will enable the most efficient method of logical record derivation. QUERYMASTER should be used as a standard checking procedure for RCI prototype for these reasons.

There are some initial pointers to good RCI/COBOL program design. First, the program must be able to monitor and react appropriately to various status and response-codes returned by RCI. The program is to RCI what the end-user is to QUERYMASTER.

The programmer should also note that CAFS-searches are done with no locks applied. It is therefore possible for a concurrent process to alter or even erase records which have just been passed back from RCI to the program. The program should not therefore assume that RCI returned data still exists unchanged.

In the TP context, a design balance must be struck between buffer sizes search limits and time-out times. An RCI search cannot currently be restarted after the search limit has been reached, but the default search limit is 'ALL'.

In the multiphase TP context, a performance comparison can be made between two approaches to searching:

- perform the search in one phase and store the results in a hit file;
- spread the search across phases, linking with SAVE/RESTORE commands.

The first technique will spend less time RCI-searching and is recommended if the hit file overhead is not considerable.

6.10 Using DCI

The Direct CAFS Interface (DCI) provides an interface which can be called from any high level language supporting the standard call interface. It supports all the CAFS facilities available through the VME Restricted System Interface (RSI). DCI may be used on both data and text (see § 6.8).

At present DCI will only access free standing files, subject to the following restrictions:

- the file must not contain spanned records;
- the file must not contain non-embedded keys;
- IDMSX databases cannot be accessed (this restriction will be removed in the next release—due early in 1986).

The interface consists of about 20 routines with simple integer or character string parameters. The information passed to any DCI routine is easy to read and define.

Any search of a file can be split up into the following four distinct stages:

(i) open the file;
(ii) prepare the search;
(iii) retrieve the hit records;
(iv) close the file.

The following example (written in SCL) shows how the basic set of DCI routines can be used to search a file for all the records containing the value 'RICHARD' in a particular field. It uses the five basic DCI routines.

(i) Open the file

This stage consists of simply assigning the file and calling the DCI routine CAFS—OPEN which is used to declare a file to DCI and to pass certain information which remains constant during a search. CAFS_OPEN returns a currency which is used to identify the file on subsequent calls of DCI routines.

```
INT       CURRENCY, RECORDSIZE, DCIRESULT
(15)INT COUNTERS
STRING BUFFER := FILL(40); AF(CAFSFILE,FILE1,LEVEL=C)
CAFS_OPEN(FILE1,BUFFER,RECORDSIZE,COUNTERS,CURRENCY,DCIRESULT)
```

(ii) Prepare the search

The data layout of the file is defined in terms of binary, character and Self-Identifying Format (SIF) fields. In this example each record within the file consists of a 6-byte character field followed by a 4-byte binary field and a 30-byte character field. A simple search is set up.

```
STRING DEFINITION          := 'FIELD1 6;FIELD2 4 BINARY;FIELD3 30'
CAFS_DEFINE_DATA_LAYOUT(CURRENCY,VAL DEFINITION)
STRING CRITERIA            := 'FIELD1 EQ 'RICHARD' '
STRING NOCOUNTS            := ' '
STRING NOFUNCTIONS         := ' '
INT       NODURATION       := -1
CAFS_FREPARE_SEARCH(CURRENCY,VAL CRITERIA,NOFUNCTIONS,NOCOUNTS,
NODURATION,DCIRESULT)
```

(iii) Retrieve the hit records

The DCI routine CAFS_READ is used to read the next hit record into the buffer defined in the call of CAFS_OPEN. A result code of 9034 returned in DCIRESULT indicates that there are no more hit records.

```
CAFS_READ(CURRENCY,DCIRESULT)
```

(iv) Close the file

The file is closed and all information about it is removed from DCI's tables.

```
CAFS_CLOSE(CURRENCY,DCIRESULT)
```

The five basic DCI routines used in the above example can be used to perform more complicated searches. The remaining DCI routines can be used to exploit all the facilities available from the CAFS hardware. They provide the ability to:

- define subconditions;
- define functions to be run on retrieved records;
- convert text to and from SIF;
- checkpoint and restart searches across TP phases;
- switch on diagnostic tracing facilities.

Further examples of the use of DCI are given in §5.3 and §4.6.4. Some points of good programming practice are summarised below.

Standard function calls and subconditions can be included as initialisation actions once the data layout is defined. For this, CAFS_END_INITIALISATION and CAFS_END_PHASE are provided so that the file is opened, the data layout, standard functions and conditions are defined, and the initialisation ended.

If more than one search at a time is required, then the file must be opened twice, with two currencies, as if it were two separate files. The file must be assigned two local names in SCL.

Currencies from DCI should not be SAVEd and RESTOREd across multi-phase TP transactions. An AVM must know for which file it is restoring current position and use its own currency for that file accordingly.

Saving search positions in TP over phases can be complicated. Anything done in initialisation can be assumed permanent in each AVM, so this should include as much as possible. For each subsequent phase careful saving and restoring will be required.

A limit of 15 elementary comparisons can be made at one time using CAFS. If logical ANDS, ORS or NOTS are used, their combined comparisons can quickly use up the 15 CAFS channels. The number of elementary comparisons can be maximised by using the CAFS QUORUM Processor. It gives the cheapest way of using extra key channels to combine conditions. Using counts as well as key channels uses up key channels.

If search subsetting is required on an ISAM file when the prime key is used in the condition, ANDTHEN must be used.

6.11 CAFS search option (CSO)

CAFS Search Option (CSO) is an option available with VME from SV211 onwards. Its objective is to enable some existing programs to get the benefit of CAFS without the need to re-write or re-compile the programs concerned.

CSO works by initiating a CAFS search from the SCL statements that run the program, rather than from within the program itself. The program thus continues to operate as though it was reading the whole file, when in fact it only sees hit records.

For example, suppose a program ISSUEDEMAND processes a file MYMASTER containing records of several different types, each starting with a four-character field, TYPE. The program is only interested in records where:

TYPE = 4 and AMOUNT_OWING ⩾300,000.00

The following SCL boosts the performance of ISSUEDEMAND:

```
ASSIGN_FILE (MYMASTER, MAINFILE, LEVEL = C)
STCC (LNAME = MAINFILE,
      ITEMS = TYPE (1:4) & AMOUNT_OWING (9:9),
      CONDITION = '(TYPE EQ 4) AND (AMOUNT_OWING GE 300000)')
ISSUEDEMAND
```

ISSUEDEMAND continues to evaluate returned records as normally because it has not been changed in any way. It finds that each record is a hit and that the file appears to be only 0.001% or whatever of its previous size. The program completes its task a great deal sooner than before.

To be able to take advantage of CSO a program must conform to the following rules:

- It must read a file serially and process selectively. A program which processes every record will not work faster because of CSO.
- The input file must be either serial or ISAM.
- As issued, CSO can only be used with programs using the COBOL runtime library to access a file.
- The program may write to other files, but must not try to write processed records back to the file from which they have been extracted by CAFS.
- The selection of records must be achievable by comparing data fields with external parameter values, not on the comparison of different data fields within the same record. For example, in a sales ledger application CSO could extract all accounts which have a balance of more than £10,000 outstanding for more than six weeks, but it could not extract accounts where the balance outstanding exceeds the credit limit.

Not all the data types which can be searched by CAFS may be used as CSO parameters. The restriction is to simple integers and character strings.

The extent of the benefit of CSO depends on the hit rate, or ratio of records processed to records read. The lower the hit rate the greater the benefit. Experiments suggest that up to about 30% hits a program will run faster in elapsed time and will use less OCP power. From 30 – 70% it will still use less OCP but may actually take longer in elapsed time. Above 70% hit rate there is no benefit in either OCP load or elapsed time.

CSO is such a simple mechanism that there is no need to conduct detailed theoretical evaluations to decide which programs can be improved. It is easier to try it on candidate programs, retaining it wherever significant benefit accrues.

7 CAFS and the End User

The introduction of CAFS to a site can have considerable effects on its end users. On the surface the benefit is the faster access of data. However, the implications of this can be far reaching and can result in the end user becoming not only less dependent on the DP department but also one of its most valuable additional resources. The end user's own ad hoc enquiries often highlight data inconsistencies and bring more constructive ideas to the system design process. The users most direct interface to CAFS is via QUERYMASTER. However, benefits will also be derived from the use of RCI, CSO and DCI.

7.1 CAFS/QUERYMASTER

QUERYMASTER is the user's interface to CAFS. Its language is powerful yet easy to use. QUERYMASTER interrogates data held on mainframe ISAM files or IDMSX databases. Without CAFS, QUERYMASTER can be slow to search through large volumes of data. CAFS makes QUERYMASTER a more practical proposition for this type of enquiry. The raw power of CAFS drastically reduces the time taken to enquire on the data. The end user can then browse through large volumes of data refining his enquiries according to the results he gets from earlier enquiries.

This immediacy means the data in the mainframe becomes more accessible to the user. He thinks of the data as being his, instead of being 'locked away' in the DP department and inaccessible to him. He has more facts at his fingertips to enable him to do his job. Consequently, the end user feels he has more control over his data. He no longer has to ask the DP department whenever he wants information. This independence gives him more responsibility and job satisfaction.

The accessibility of the data makes inconsistencies more apparent. As a general rule in conventional systems the source data was often converted, summarised and printed before the user saw it. The ability to search large volumes of data in an acceptable length of time means that it is no longer necessary to summarise the data. Any data inconsistencies are therefore highlighted when it is subjected to close scrutiny by a QUERYMASTER end user.

The user finds himself capable of making enquiries that he would never have considered making before. These are typically the type of enquiries which would have involved the DP department in writing an ad hoc program and therefore would probably not have been cost justifiable within the required timescale.

The user can become more involved at the system development stage. By using QUERYMASTER on test data ahead of the system implementation stage, enquiries can be planned in advance. Users often find it difficult to appreciate the potential use of data until they can see it physically on a screen. Therefore, the earlier the user can get this thought process going, the greater the chance of a successfully implemented system. Because QUERYMASTER is easy to use, automatically navigating the database or ISAM file relationships, the user need not be involved in the technical detail of how or where his data is stored. Users cover a broad spectrum of jobs and levels of responsibility, varying from professional staff such as accountants, engineers, office managers librarians to clerical staff, storemen etc.

The users who gain most benefit from QUERYMASTER/CAFS will be those who can best understand the data and can therefore make the most useful enquiries. It is most important that staff are not only trained in the QUERYMASTER language, but also in the contents and meaning of their data.

At its simplest a Queryview may contain a single ISAM record type. This is a relatively simple concept for a user to understand. Obviously the more complicated the data structure in the Queryview the more help the user will need. Structuring an enquiry to make the fastest search of an IDMSX database made up of many sets can be an uphill task for some end users, and it is probably best to design simple IDMSX structures and duplicate data within sets in order to make the most of CAFS/QUERYMASTER.

It is also important to ensure that the user gets adequate training in basic computer concepts in order to get the most out of the software and hardware.

Most DP departments seem to have set up their own training schemes for non-computer users. Some opt for day long seminars for several users while other sites use two or three short training sessions of about an hour on a one to one basis.

7.2 CAFS/QUERYMASTER training

Once the user has understood the basics of QUERYMASTER he will need to know how to phrase his enquiry to make the best use of CAFS. Some typical pitfalls are:

- 'CONTAINS' in an enquiry will not use CAFS;
- complicated enquiries which use up more than 16 CAFS key channels will not invoke CAFS
- QUERYMASTER will always try to use ISAM file indexed searches or IDMSX CALC keys before using CAFS;
- the user also needs to be aware that different navigation can be obtained by differently phrased enquiries—thus giving different results;
- use of 'STARTING' forces QUERYMASTER to use a certain access path and hence may or may not make use of CAFS.

7.3 Data training

It is most important that the user is provided with enough training and documentation to interpret results correctly. For example,

- Does a certain amount field include an oncost amount or is this held separately in the record?
- To what period does the data relate—this current financial year?

The user who does not fully understand the data is in no position to make decisions based on his enquiries.

7.4 Computer concepts training

Once the end user has experience of this powerful tool, there is sometimes a tendency to assume that CAFS should solve every problem, which may lead him to misuse or misunderstand the computer system, to his own detriment. He may, for example, assume that because a serial search through an ISAM file is reduced to 1/60th of the time, the same will apply to the sort phase and the printing phase.

Basic computer concepts such as disc contention should also be pointed out: if the staff of one department have access to several videos, speed of access will suffer if they are all logged into the same query view making enquiries on the same files concurrently.

7.5 User support

Even when the user is fully trained he will need adequate support. The CAFS/QUERYMASTER systems will become just as critical as a TP system and therefore without training and support the user will be no better off than with an old batch system where the data is 'locked away' within the DP department.

7.6 CAFS/CSO, RCI, DCI

Whereas QUERYMASTER is the users interface to CAFS, CSO, RCI and DCI are the DP departments interface.

CSO will invoke CAFS on ISAM files. No program amendments are required therefore existing systems can be modified easily to make use of CAFS.

Typical situations where the end user may see a benefit are old TP systems with a large volume of end-of-day processing. The time for this end-of-day processing may be reduced leading to extended hours for the TP service.

Similarly, Reportmaster programs which previously ran as a batch job could be initiated by the user as a MAC job with the output coming back to his screen.

RCI will invoke CAFS on ISAM or IDMSX files.

RCI also has the benefit of reducing the complexity of programming—thus reducing development timescales. By interfacing RCI to a TP service, it is feasible to make inverted file look ups in acceptable TP response times.

DCI enables programers to invoke the most advanced CAFS features enabling the power of CAFS to be exploited directly within a high level language program. The design of really powerful and user-friendly systems becomes much simpler as the programmer can concentrate on the end user's requirements and leave the donkey work to CAFS.

8 CAFS—The Cost Case

8.1 Introduction

In recent product announcements, for example Series 39, ICL's 'CAFS with everything' policy has been re-affirmed. The question, therefore, of whether there is a cost case for (or against!) purchasing CAFS-ISP hardware is now somewhat academic. Those sites considering upgrading on 2900 series will now receive the Decision Support Controller (DSC) which, of course, includes a single CAFS-ISP module as standard. However, exploiting CAFS will cost money, and many sites who do not have CAFS at present may still be considering whether to take the product or not. This chapter reviews likely costs and benefits of following the CAFS path, in the following areas:

Hardware requirements

Software requirements

Installation and service development

The end-user area

Charging for CAFS services

The chapter concludes with a discussion as to whether a cost/benefit analysis is relevant to CAFS exploitation.

8.2 Hardware requirements

Any new 2900 series or series 39 machines delivered will include CAFS as standard. However, as user upgrades may be being made from old hardware bases, it would be prudent to list the current hardware position:

2900 Series ('S' machines only)

(i) CAFS may be fitted as a field upgrade to a Device Control Module (DCM) of a Device Control Unit Model 2 (DCU2), on an existing 2900. Two CAFS may be fitted in DCU2, one on each DCM. Total (list) hardware prices for these upgrades are about £35,000 (single CAFS) and £60,000 (dual CAFS).

(ii) A single CAFS-ISP module is fitted as standard to a Decision Support Controller (DSC). The unit cost, DSC + CAFS, is about £59,500.
A CAFS extension module may be fitted to a DSC, as a field upgrade, to provide a dual CAFS capability. This upgrade costs between £24,000 and £33,000 depending on the type of disc in use.

CAFS can only search discs belonging to the ICL MDSS family.

Series 39

CAFS is fitted as standard to the first controller on any Series 39 configuration, and is optional on subsequent controllers. ICL will no doubt quote prices for any configuration, but an entry level system, comprising a single node 4Mb. store, an HSDC with CAFS and two FDS300 discs is currently priced at around £150,000.

The above gives a guide to hardware costs for sites considering a direct upgrade from no CAFS capability. Many sites will, by now, have CAFS hardware. The next question, in both cases, is what software is required, and how much will it cost.

8.3 Software requirements

The first, and most important point, is that CAFS can only be exploited under the VME operating system (including VME/B or VME 2900 variants). The minimum release level is SV201, but to exploit fully the list of products below, and the latest CAFS enhancements, you will need version SV211. Sites which are primarily DME, or DME under CME may already be moving to a VME system. In this case it may be more cost-effective to consider provision of CAFS services as part of the software conversion.

The products which can exploit CAFS are listed below, together with their current quarterly licence fees. The fees quoted relate to a 2966 single machine licence without any discount.

QUERYMASTER	– interactive enquiry for conventional/IDMSX applications ... £1,011
RELATIONAL CAFS INTERFACE	– high level language CAFS interface ... £2,130
DIRECT CAFS INTERFACE	– interface to CAFS at the functional level ... £1,415
CAFS SEARCH OPTION	– VME option to provide limited CAFS facilities from SCL ... £820

CAFS SEARCH OPTION is a base option on VME; the other products are all VME superstructure products.

In addition to this CAFS exploiting software, the Data Dictionary System (DDS) is required to drive QUERYMASTER and RCI, and there are various product dependencies for release of COBOL compilers, IDMSX etc.

It might be concluded that going from a 'no CAFS' position to provision of a full CAFS service would be expensive and involve a long learning process. However, the true position is neither as clear-cut nor as dire. Many installations will be in the position of having some of the components, and the decision to exploit CAFS will, more than likely, be dependent on external factors, such as development resources and end-user requirements. Irrespective of such external factors, it is important for installations considering CAFS services to be aware of the investment commitment required. CAFS is not a small step, and may cause a radical re-think with regards to service provision at both the DP and end-user formations.

8.4 Installation and service development

To make the cost argument more coherent this section will consider the position of a 'typical' ICL site, albeit one that has been chosen to show both good and less good implications for the introduction of CAFS. This hypothetical site consists of:

- a large single or dual 2966 running a mixed workload (batch/MAC/TP) under VME, and possibly just beginning to show signs of strain;
- an end-user base with an amount of enquiry requirements to be satisfied;
- an application mix involving conventional and IDMSX systems, some old, some new.

It is likely that the potential need to exploit CAFS will come either as a result of end user pressure, due to increasing response times on enquiry services and demand for more applications, or be identified by installation management as a means of getting more out of the existing investment without major upgrade.

8.4.1 Costs

If this mythical site were to pursue a CAFS path, where would costs be incurred? Obviously, the decision to develop CAFS services will cause work in the DP department. At least the following will be involved:

- training/learning for DP staff;
- installation of the chosen software products, and possibly some upgrades to later releases of existing software;

– putting systems on DDS if QUERYMASTER and/or RCI are to be used;
– education of end-users.

Many of the above can be open-ended, and may require more than a given installation is willing to commit. Some CAFS services, say using QUERYMASTER, can be developed cheaply, while others are more expensive. The only rule is that there are no hard and fast rules. Should an installation perform a quantitative assessment of likely costs, costs of providing a CAFS service may appear high. It is probably wise to consider provision of CAFS services within the context of application upgrades, where the 'first cost' of introduction may be offset by obviating the need for some development work. The 'first cost' can then be spread out over future CAFS applications; many CAFS using sites have reported that the costs incurred in extending CAFS use are significantly less than those for its first use. This will, to some extent, depend on whether a given site can take a medium-term view of the CAFS investment.

8.4.2 Benefits

What likely benefits will there be for the DP department? Section 8.5 considers the end-user, but CAFS is available to any user of the machine, including the DP section.

Trials by several sites have shown that using CAFS reduces processor workload by up to 95% (or more) for the services exploiting CAFS. Depending on the scale of implementation , this could enable deferment of other hardware upgrades with consequent savings in interest charges on the capital investment. Additionally, it may be possible to use CAFS to remove application bottlenecks, thereby smoothing machine load and allowing more effective use of the existing hardware base.

Products such as QUERYMASTER and RCI can lead to significant improvements in applications development 'productivity'. Although not easy to measure, some installations are reporting significant (i.e. >50%) reductions in resource commitment when CAFS and CAFS-related software are employed.

Another significant benefit is the reduction of documentation necessary for enquiry systems, particularly where QUERYMASTER is used.

The DP section can use CAFS services itself. It is surprising how useful QUERYMASTER can be as a testing aid, and the potentiality for rapid prototyping of applications with QUERYMASTER and RCI can be exploited.

There may well be benefits in other areas. Some typical comments from sites include:

– the end-user sees DP in a better light;
– the DP department keeps up with the latest ICL offerings;
– having got CAFS, it does not cost very much to keep it and exploit it further. Every new application shares the start-up costs over a wider area.

To counter this air of optimism, some sites have reported that development of CAFS services has cost them more than they thought it would, some reasons being:

Lack of documentation/training from ICL. This is now being addressed, and a specific CAFS manual and training course will emerge in 1985.

Lack of expertise in the first line support provided by ICL. This is again being addressed, but, in some cases, enterprising users of CAFS are outstripping ICL's ability to keep their own staff up to date.

A slightly more contentious, but often heard, argument against CAFS is that it does tie a site or system to ICL equipment. ICL will, of course, exploit this for sound business reasons. Change of hardware supplier is a big, and expensive, investment decision. For some users the decision as to whether to exploit CAFS or not may become part of a wider argument about the hardware base. Although it is doubtful whether any site has tried to take away a CAFS service once provided, end-user resistance may be considerable.

8.5 The end-user area

It is in the end-user area that many of the benefits of CAFS become apparent. These fall into two main areas:

- the provision of acceptable enquiry interfaces on applications, conjoined with ...
- the real, and significant, reductions in response times for data searches.

Both of these can be translated into real cost savings. The reduction in unproductive waiting time at terminals enables end-user departments to be more productive, and the ability of CAFS to handle searches not possible without it can reduce the pressure from end-users for system modifications.

A cautionary note. Once CAFS services have been provided for end-users on some applications, they themselves develop a 'CAFS with everything' syndrome. Careful planning and control of end-user requirements cannot be foregone.

8.6 Charging for CAFS services

Many sites either charge end-user departments directly for services or have an internal cost-recovery mechanism. Many charging systems are based on algorithms using a combination of elapsed and OCP time. CAFS, unfortunately, negates these for two main reasons:

- the reduction in OCP time for CAFS searches;
- the possible reduction in elapsed time as a consequence of reduced response time.

Charging algorithms will need to be re-examined. Some sites have got round the problem by making a fixed charge for provision of CAFS services, and using only elapsed time as a measure thereafter. This has the advantage of directly equating the costs of providing a CAFS service with the revenue gained from it.

An additional problem can arise if disk transfers are included in the charging algorithm. At present there is very little quantitative monitoring of CAFS access within VME, especially if application is to IDMSX systems; its provision is one of the enhancements requested by this Working Party. Unfortunately, even if such measuring facilities were provided they would not tell the complete story. For example, a CAFS search on a database may involve high disk access rates, yet with the CAFS search speed and reduced OCP load may be more efficient in terms of machine resource usage than a similar search on conventional lines.

Significant use of CAFS techniques can thus render charging algorithms unreliable. Very few of the installations surveyed have come to a satisfactory solution to the problem, with the possible exception of the 'fixed charge' concept mentioned above. This area of CAFS business is therefore as yet unfinished.

8.7 Conclusion

Is it possible to define cost/benefit criteria for CAFS? The answer is, 'Yes', provided that the analysis is done on the basis of one application or a group of applications which can be considered in isolation. However, such an application may need CAFS anyway, irrespective of costs.

For example, in some applications involving heavy search loads CAFS can be considered as part of the fundamental system, and simply rolled together as part of the total development investment.

One worthwhile approach could be to perform a cost/benefit analysis on a pilot scheme project which appears as though it has a potential for CAFS exploitation. Once this has been done the return on further exploitation could be evaluated. On the other hand, some sites have already pointed out that it can cost more to appraise CAFS than to go ahead and implement. The 'suck it and see' approach may be most cost-effective of all!

This chapter is perhaps misnamed. Only an individual site can appraise a cost case for their needs. However, the contents have been based on the real experiences of CAFS users, and indicate the areas to consider. A final conclusion could be that CAFS has as significant an effect on investment decisions as on the applications which may exploit it.

9 The Future of CAFS

9.1 The working party view

9.1.1 Introduction

Over the past few months, the CAFS Working Party has spent some time in producing a list of enhancements for CAFS. This has been achieved through close liaison with a panel of experts from various areas within ICL, who were able to advise on the feasibility of incorporating our suggestions.

The initial basis for this list was the 'Recommendations for the Future' chapter of the first Working Party Report. Some of the recommendations there have already been actioned, others have been changed to reflect more recent thinking in certain areas. In addition, in the light of experiences gained by users over the last eighteen months in the live use of CAFS, various additional requirements have been included in the current list.

The list may seem long. This is not because there is great dissatisfaction with CAFS within the Working Party, but because we all feel that despite the product's merits, continued effort from ICL is needed to make it still better. The list should be viewed, therefore, as an attempt to make constructive comments which will assist ICL in their strategic development of CAFS, and in their allocation of priorities to the various requirements.

For the purposes of this report, the list has been broken down into the following categories:

Items on which ICL have already taken some action.

Items which ICL intend to action within the life of the current CAFS engine.

Items which ICL have asked the Working Party to put into priority order, implementation of which is still under consideration.

Items which ICL have indicated as being not feasible within the life of the current CAFS engine.

9.1.2 Items on which ICL have already taken some action

The following specific problems are all addressed by enhancements in QUERYMASTER 250.

- CAFS/QUERYMASTER should support Serial RECMAN files.
- QUERYMASTER should use CAFS searching on QUERYMASTER-produced hit files (that is, files created by MAKE or EXTEND commands).
- QUERYMASTER should use CAFS hardware facilities for counting variable-length records. This will be incorporated into QUERYMASTER.250 for RECMAN files, but not for IDMSX.
- The current QUERYMASTER record length restriction of 4Kbytes is too small. This will be raised to 32Kbytes in QUERYMASTER.250.
- A lot of frustration and wasted machine time would be saved if it were possible to insist that particular QUERYMASTER accesses could only be made when CAFS is available. Also, QUERYMASTER should warn the end-user if CAFS is not being used on an enquiry, so that the end-user can abort the query if he so wishes. QUERYMASTER.250 will give a warning to users if a serial search is not going to use CAFS, and to allow the DP department to prevent access to a query view if CAFS is not going to be available.
- QUERYMASTER should allow the end-user to confirm easily that CAFS hardware is operational.
- The QUERYMASTER log-in procedures are too lengthy.
- Division is not permitted in QUERYMASTER macros.
- Restrictions on the amount of data that can be sorted should be raised.
- QUERYMASTER namespace is insufficient for CAFS versions of some Queryviews.

Additional points raised by the working party which ICL have already begun to address are:

ICL should develop specific documentation on CAFS as part of their manual range. ICL are currently developing a new Technical Publication entitled 'CAFS Exploitation: a User Guide'.

CAFS training needs to be provided by ICL. A one-week general course on CAFS is now available.

There is insufficient CAFS experience amongst ICL Support Staff. ICL is addressing this problem.

Information should be made available on facilities likely to be offered in future product releases of CAFS and related software. ICL are addressing this problem.

As disc drives become faster and faster, it is important that CAFS keeps one step ahead, so that it is always able to make best use of the disc speeds available.

9.1.3 Items which ICL intend to action within the life of the current CAFS engine.

- CAFS/QUERYMASTER can currently read PDS files, but only if dictionary entries for the PDS files have been set up, and Queryviews created from these. The Working Party believe that ICL should produce a simpler method of achieving this.
- The ability to abort a query ought to be made more flexible within QUERYMASTER, so that end users can break out of a query at virtually any time they wish.
- QuickBuild components such as Application Master and Reportmaster should be able to use CAFS transparently. This can be achieved now only by explicitly incorporating DCI modules or RCI/COBOL modules in an AM/RM program via their COBOL interface.
- Text and office products should use CAFS for searching their data. ICL made reference to this in their Statement of Direction on CAFS in August 1984. It is not clear at the moment how or when steps will be taken in this area.
- CAFS should be made available anywhere within a distributed network. The ICL Statement of Direction also referred to this. Again, we are awaiting further information.
- QUERYMASTER HELP facilities should be improved, and include the ability to customise HELP screens.

9.1.4 Items which the Working Party have put into priority order

Priorities were arrived at by rating each item as Essential, Desirable, Nice, Irrelevant or Bad Idea. Weightings were then given to each of these five categories and the results were collated. The greater the points score the higher the priority. As a rough guide to how important we considered these items, the first seven were considered essential by more than a third of the Working Party.

1. Zero dates are handled differently by CAFS and non-CAFS QUERYMASTER enquiries.
2. QUERYMASTER should be able to perform arithmetic on DATE and CENTURYDATE fields.
3. IDMSX should use CAFS for such DML commands as FIND NEXT WITHIN REALM.
4. QUERYMASTER should use CAFS to search all data items which are described as packed-decimal format.
5. QUERYMASTER should use CAFS hardware facilities for counting IDMSX records.
6. The QUERYMASTER PRINT command should have parameters to ask for multiple copies (including none) of a print and to say whether or not the session output file should be deleted. The PRINT command should also show on the screen how many records have been spooled for printing.

7. If QUERYMASTER macros need to be changed, they have to be re-input from scratch. A macro editing facility within QUERYMASTER is needed to resolve this untidy situation.
8. There should be a single routine to convert all formats to and from Self-identifying text, including the use of identifier and mask to convert groups of Self-identifying text. There should be an automated SIF conversion within VME.
9. Sorts within QUERYMASTER take too long compared with the time taken to select the data.
10. QUERYMASTER should allow fuzzy or stem matching on value-based sets. Limited use of OR conditions in value-based sets would also increase relational capabilities.
11. QUERYMASTER and RCI should be able to handle logical NOT operations.
12. VME should produce performance statistics on CAFS usage by terminal, job, AVM, etc.
13. At the current release of QUERYMASTER, the end-user is often left not knowing what is happening to his query. It would be better if QUERYMASTER were to give comforting signals to the end-user that his query is being serviced.
14. It should be possible to change the size of the maximum CAFS search fragment more dynamically than at system set-up time.
15. QUERYMASTER is at present unable to advise the end-user how long a query will take. If QUERYMASTER had access to volumetric data, through the Data Dictionary for example, an estimate of likely CAFS search time could be given.
16. QUERYMASTER would be more consistent with Reportmaster if it had an ENDSWITH operator as well as STARTSWITH.
17. The Data Dictionary would be a much more usable tool if it were made CAFS-searchable.
18. It is essential that CAFS keep in step with developments in document storage and international character set standards.
19. In IDMSX data structures it should be possible to use CAFS to search record and set index blocks. It should also be possible to use these indexes to limit a CAFS search to specific pages.
20. CAFS should not be affected by relocated and fragmented IDMSX records. ICL suggest that this would have to be dealt with by inhibition and control of such circumstances.
21. CAFS ought to be able to search data items which are described as single-precision floating-point format.
22. In order to achieve greater search efficiency, it would be better if CAFS could be told to limit its search to specific pages of a database.
23. CAFS will need to be integrated with non-VME products like Unix.
24. When using duplex data it should be possible to constrain CAFS searches to one plex while TP reads use the other plex.
25. CAFS/QUERYMASTER should support Hashed Random RECMAN files.
26. It should be possible for QUERYMASTER to create an ISAM file to hold its output records on an indexed file. In this way, it would be possible to perform an outer join operation.
27. VME should allow DCI field definitions to be held within the file node, or within the file header, to enable this information to be picked up automatically.
28. QUERYMASTER should support records containing Multiple string SIF fields as well as single string.
29. String operators should be provided for invisible characters in Self-identifying fields.
30. Access to the length element of a Self-identifying field is required so that queries which rely on the lengths of words can be carried out.

The following items are still to be given a priority:

The CAFS Search Option (CSO) currently supports only COBOL-based software. It would be of much greater benefit if it were extended to other software.

VME BUS utilities that use selection criteria would be enhanced if they used CAFS.

As the manipulation of data through QUERYMASTER becomes more sophisticated, there is a need for CAFS to incorporate a facility for performing join operations by specialised hardware. Ideas on this have already been published by ICL.

9.1.5 Items which ICL have indicated as being not feasible within the life of the current CAFS engine

The fact that these are identified as longer-term enhancements does not render them unimportant. In fact, we believe that the enhancement of the CAFS engine is in itself of high priority.

- QUERYMASTER should use CAFS hardware facilities for totalling.
- CAFS cannot evaluate a 'CONTAINS' condition.
- QUERYMASTER/CAFS should inform the end-user of how long a search is likely to take. QUERYMASTER should warn of a possible mismatch of question and answer if it has a choice of access-paths.
- Searching text fields should be made possible without having to change the field to TEXT format.
- Better ENDSWITH operation is required for Self-identifying fields.
- There is a need to know where in a record a hit has occurred when using Self-identifying fields.
- A search should be able to continue within a record after a hit has occurred within a Self-identifying field, in order to count all such occurrences.
- There is a need for Self-identifying format fields to be able to be searched for 'string A' followed by 'string B' within the same field.
- CAFS should support text searching within a fixed length field of Self-identifying text.
- It would be less restricting on the way data is stored in files if CAFS were able to search across physical block boundaries of a file.
- Emulation routines to perform DCI functions without using CAFS would be useful in case of CAFS unit failure.
- It is often necessary to set up an index for a document, based on all different occurrences of words within the text. To facilitate this, CAFS should be able to identify all different values occurring within a text field.

9.1.6 Conclusion

In summary, the Working Party have put their minds to a number of proposed enhancements to CAFS, and have attempted to help ICL by giving priorities to them. Certainly there have been lengthy discussions on many of the proposals, and it is to be hoped that ICL will devote their attention to dealing with many of the points raised.

The enhancements fall broadly into three areas of improvement. First, there are those which CAFS already does now in some form, but which we feel it could do better. For example:

- QUERYMASTER should make better use of CAFS, including better information to the end-user about when and how CAFS is being used for an enquiry.
- There is a need for better information from ICL on CAFS in general: manuals, training courses and the like.
- The usage of CAFS should be extended, particularly for accessing IDMSX databases.
- Self-identifying format (TEXT) fields need to be more easily manageable if they are to be used significantly in the future.

Secondly, there are items where the range of CAFS applicability could be extended. For example:

- There should be CAFS support for all field types and file organisations allowable within VME.
- CAFS hardware should be extended to do more: CONTAINS, totalling, counting, joins and other functions.
- CAFS should be extended as much as possible to take advantage of new advances such as faster disc drives.
- There needs to be greater scope in the way in which Self-identifying format fields can be interrogated.
- QUERYMASTER and RCI would both benefit from better join facilities between record types.
- Performance statistics on CAFS usage are needed.

Thirdly, there are items where CAFS could be extended to application areas which are not currently feasible. For example:

- CAFS needs to be better interfaced with ICL products such as PDS, Application Master, Reportmaster and Data Dictionary.
- CAFS should be made available to text and office products, anywhere within the distributed network.
- CAFS will need to be extended to non-VME products like Unix.

9.2 The future of CAFS—an ICL view

This section is in some ways the equivalent of an ICL Statement of Direction: it is of necessity confined to general statements rather than containing a list of commitments to particular developments and dates.

One continuing feature of the CAFS development story has been the close collaboration between ICL and potential or actual users. During the initial research stages, great care was taken to test the capabilities, both of the hardware itself and of the software which exploited it, against real user requirements and preferences. These interests have continued to be expressed through the ICLCUA CAFS User Group and its several Working Parties, and the current Report is an example of the continuation of this mutually beneficial process.

Without specifying what stage of progress or planning has currently been reached, it can be stated that the the following general lines of development will be followed:

CAFS hardware

In all elements of computer hardware, each successive generation is smaller, faster, and cheaper than its predecessor. This has already been proved for CAFS: CAFS 800 filled two tall cabinets; CAFS-ISP occupies a mere nine boards; the next version will certainly be even more compact, in line with general industry trends. This will permit CAFS capability to be implemented across a wider spread of ICL's hardware product range.

Extensions will be made to the capabilities of the CAFS engine itself. Some of these will be in direct response to requests input from the Working Party responsible for the Report, and will cover some of the requirements categorised in the preceding section as 'not feasible to implement within the life of CAFS-ISP'. Others will be derived from the independent assessment, by ICL's research and development authorities, of the best opportunities for increasing the power and functionality of CAFS. Aspects of these developments have been discussed in confidence during the deliberations of the Working Party, and general agreement has been reached.

A particular area in which developments can be confidently predicted is the ability of the CAFS hardware to search and evaluate text.

The role of CAFS in aspects of relational processing is already important, and will certainly become more so. There will therefore be extensions to the list of functions in this field which can be directly implemented by the CAFS hardware.

CAFS software

The fundamental CAFS capabilities of VME will of course be extended to permit the fullest exploitation of all new hardware functions.

These will also be progressively incorporated into enhanced versions of ICL software products in, for example, the Data Management and QuickBuild families.

CAFS will be used where appropriate by an increasing number of application package products developed and marketed by ICL Business Centres. The fact that from now on CAFS is an automatic constituent of ICL mainframe systems will be reflected in decisions to simplify and accelerate development by deliberately not building non-CAFS versions of certain packages.

CAFS has already aroused considerable interest in external software houses. So far as possible ICL will encourage them both to add CAFS capabilities to existing packages and to develop new packages which make optimum use of CAFS.

The amount of development work associated with CAFS that is in progress within ICL, much of which cannot be described in this report, is a clear indication that the strategic importance of CAFS is fully recognised by the Company. At the same time, the pressure for enhancements demonstrates with equal force the importance which CAFS is assuming in the information services of members of the CAFS User Group. However, the value of the capabilities which are already available has been proved by the awards in 1985 to the Southern Water Authority of the Office Automation Award for the best information storage and retrieval system, and to ICL of the Queen's Award for Technological Innovation for the existing CAFS system.

Select CAFS bibliography

Articles etc. of major significance in the history of CAFS and its applications

T.R. Addis, 'A relation-based language interpreter for a content addressable filestore.' *ACM Transactions on Database Systems* 7 (2), June 1982.

E. Babb, 'Implementing a relational database by means of specialised hardware.' *ACM Transactions on Database Systems* 4 (1), 1979.

'The CAFS File Correlation Unit.' *ICL Technical Journal* 4 (4), November 1985.

L.D. Burnard, 'CAFS and Text: the View from Academia.' *ICL Technical Journal* 4 (4), November 1985.

J.W.S. Carmichael, 'Implementation of a Content Addressable Filestore.' *SICOB Convention Informatique*, Paris, September 1980.

'IDMS and CAFS.' *BCS '81—Information Technology for the Eighties*, London, July 1981.

'Personnel on CAFS.' *ICL Technical Journal* 2 (3), November 1981.

'The contribution of a computer specialising in data management.' *SICOB Convention Informatique*, Paris, September 1981.

'CAFS: an updated overview.' *Yankee Group Seminar—The next generation of database*, London, November 1982.

'Content Addressing Technology and Cartographic data.' *Euro-Carto I*, Oxford, December 1982.

'Use of CAFS with Large Databases in the Public Service.' *VLDB*, Singapore, May 1983.

'CAFS: Database Made Easy.' *Pergamon Infotech: ICL The Way Ahead*, London, December 1984.

'The application of ICL's CAFS to Text Storage and Retrieval.' *PROTEXT I*, Dublin, November 1984.

'History of the ICL Content Addressable File Store.' *ICL Technical Journal* 4 (4), November 1985.

G.F. Coulouris, J.M. Evans, R.W. Mitchell, 'Towards Content Addressing in Data Bases.' *BCS Computer Journal*, February 1972.

C.H.E. Corbin, 'Creating an end-user CAFS service.' *ICL Technical Journal* 4 (4), November 1985.

L.E. Crockford, 'An associative data management system.' *ICL Technical Journal* 3 (1), May 1982

L. Harding, 'Content Addressable Filestore as a memory resource.' *ICCC 6—Pathways to the Information Society*, London, Sept 1982,

G. McC. Haworth, 'The CAFS System Today and Tomorrow.' *ICL Technical Journal* 4 (4), November 1985.

P.B. Hawthorn, D.J. Dewitt, 'Performance analysis of alternative database machine architectures.' *IEEE Trans. Software Eng.* vol. SE-8 no. 1, January 1982.

A.T.F. Hutt, 'History of the CAFS Relational Software.' *ICL Technical Journal* 4 (4), November 1985.

ICL Computer Users Association, 'CAFS Exploitation.' CAFS *User Group Working Party Report* July 1984.

M.H. Kay, 'Textmaster: a document retrieval system using CAFS.' *ICL Technical Journal* 4 (4), November 1985.

M.J.R. Keen 'DRAGON: The development of an expert sizing system.' *Database and Network Journal* 14(1), 1984.

V.A.J. Maller, 'The Content Addressable Filestore.' *ICL Technical Journal* 1 (3), November 1979.

'A Content Addressable Filestore.' *IEEE Spring Conference, San Francisco*, 1979.

'Information retrieval using the Content Addressable Filestore.' *IFIP Congress, Tokyo* October, 1980.

'The content addressable filestore: a technical overview.' *IUCC Bulletin*, 3 (1), April 1981.

V.A.J. Maller, J.W.S. Carmichael, 'The content addressable filestore.' *IFIP—File structures and databases for Computer Aided Design*, Frankfurt, September 1981.

N. McPhail, 'Development of the CAFS-ISP Controller product for series 29 and series 39 Systems.' *ICL Technical Journal* 4 (4), November 1985.

R.W. Mitchell, 'The Content Addressable Filestore.' *Proc. Online Conference on Database Technology*, April 1976.

J. Myszko, 'A corporate human resources information system.' *BCS '81—Information Technology for the Eighties*, London, July 1981.

G.G. Scarrot, 'The Wind of Change.' *ICL Technical Journal* 1 (1), November 1978.

R.M. Tagg, 'CAFS-ISP: Issues for the Application Designer.' *ICL Technical Journal* 4 (4), November 1985.

A.J. Walker, 'Content Addressing: the key to simple flexible design.' *Database Design Update: Proc. BCS Database Specialist Group*, April 1984.

D. Walker, 'Secrets of the sky: the IRAS data at QMC.' *ICL Technical Journal* 4 (4), November 1985.

A.G. Ward, 'Name-tracing using the ICL Content Addressable Filestore.' *VLDB*, Singapore, August 1984.

P.R. Wiles, 'Using secondary indexes for large CAFS databases.' *ICL Technical Journal* 4 (4), November 1985.

F. Wolferman, 'Der Context Addressable Processor—ein Datenretrievalsystem von ICL.' *BIFOA*, Cologne, January 1981.

ICL Technical Publications

CAFS R30053, 'CAFS Exploitation: a practical guide.' 1986.

DCI R00421, 'Direct CAFS Interface Programming Guide (DCI.100).' 1985.

——R00431, 'Direct CAFS Interface Reference Card (DCI.100).' 1985.

DDS R00408, 'Data Dictionary System: The DDS Model (DDS.700).' 1984.

IDMSX R00154, 'IDMS Part 2: Database Establishment (third edition).' 1984.

—— R00155, 'IDMS Part 3: Using a Database (third edition).' 1984.

—— R00156, 'IDMS Part 4: Database Programming (third edition).' 1984.

—— R00153, 'IDMS Part 5: Database Design (third edition).' 1984.

QUERYMASTER R00433, 'Using QUERYMASTER (QM.250).' 1985.

—— R00434, 'Using DDS to prepare a Query View (QM.250).' 1985.

—— R00435, 'Running QUERYMASTER in VME (QM.250).' 1985.

—— R00436, 'QUERYMASTER User's Reference Card (QM.250).' 1985.

RCI R00251, 'The Relational CAFS Interface User Guide (RCI.100).' 1985.

VME R00475, 'Programmer's Guide.' 1985.

Document Reference: ICLCUA.002

First Edition: November 1985

Published by:
ICLCUA (UK)
PO Box 42
Bracknell, Berks.
RG12 2LQ, UK.