DEVELOPMENT OF A TOOL TO CURL OUT BUSINESS LOGIC OUT OF LEGACY CODE

PROJECT WORK DONE AT
INFOSYS TECHNOLOGIES LTD.,
BANGALORE

PROJECT REPORT P-1220

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR AWARD OF THE DEGREE OF
M.Sc.(APPLIED SCIENCE) SOFTWARE ENGINEERING
OF BHARATHIAR UNIVERSITY, COIMBATORE

SUBMITTED BY
ANUSHA DAMODHARAN
REG NO. 0137S0023

UNDER THE GUIDANCE OF

External Guide
Mr.Sudheer H.R
Infosys Technologies Ltd.,
Bangalore

Internal Guide
Mr.R.Rajasehar –MCA
Lecturer-Dept of CSE-KCT
Coimbatore

Estd. 1984

Department of Computer Science And Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY,
COIMBATORE-641006.
(JUNE-2004 TO OCTOBER-2004)

Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
(Affiliated to Bharathiar University)
Coimbatore-641006.

(JUNE-2004 TO OCTOBER -2004)

CERTIFICATE

This is to certify that the project entitled

DEVELOPMENT OF A TOOL TO CURL OUT BUSINESS LOGIC OUT OF LEGACY CODE

Done By

ANUSHA DAMODHARAN
0137S0023

Submitted in partial fulfillment of the requirements for the award of the degree of M.Sc. (Applied Science) Software Engineering of Bharathiar University.

[illegible]
Professor and HOD

[illegible]
Internal Guide 27/9/04

Submitted to University examination held on

[illegible]
Internal Examiner

[illegible]
External Examiner

Infosys

Infosys Technologies Limited
Regd. Office : Electronics City, Hosur Road
Bangalore - 560 100, India.
Tel : 91 80 2852 0261 Fax : 91 80 2852 0362

10 September 2004

TO WHOM SO EVER IT MAY CONCERN

This is to certify that Ms. Anusha Damodharan, student of Bharathiyar University, in the M.Sc(Software Engineering) has successfully completed the assignment with our Banking Business Unit, Infosys Technologies Ltd, Bangalore.

She has worked on the project "Development of a Tool to curl out Business Logic out of Legacy Code" from 22nd June 2004 – 10th Sep 2004. Her performance was satisfactory.

We wish her all the best in her future endeavors.

T.S. Venkata Ramanan

T S Venkataramanan
Head – Engineering
Banking Business Unit.

DECLARATION

I hereby declare that the project entitled "**DEVELOPMENT OF A TOOL TO CURL OUT BUSINESS LOGIC OUT OF LEGACY CODE**" submitted to Bharathiar University, Coimbatore as the project work of Master of Science Degree in Software Engineering, is a record of original work done by me under the supervision and guidance of Mr. Raghavan Muthukrishnan Infosys Technologies Ltd., Prof. K.R.Baskaran – Asst. professor & course coordinator (Software Eng.neering) , Mr. R.Rajasehar -- Lecturer. Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore and this project work has not found the basis of the award of any Degree/ Diploma / Fellowship or similar title to any candidate of any University .

Anusha

Place : COIMBATORE

Date : 23-09-04 .

Anusha Damodharan

Reg.No. 0137s0023

M.S.c (Software Engineering)

Kumaraguru College of Technology

ACKNOWLEDGEMENT

An endeavor over long period can be successful only with the advice and support of many well – wishers. I take this opportunity to express my gratitude and appreciation to all of them.

I am bound to express my gratitude to our beloved Principal **Dr.K.K.Padmanabhan B.Sc.(Engg.,), M.Tech., Ph.D.,** Kumaraguru College of Technology , Coimbatore, for constant encouragement throughout my project.

I wish to thank **Dr.S.Thangaswamy B.E (Hons), Ph.D.,** Head, Department Of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, for his invaluable guidance and suggestions that encouraged me to complete this project successfully.

I admit my heart full thanks to my internal project guides **Prof. K.R.Baskaran – Assistant Professor and course coordinator [Software Engineering] and Mr.R.Rajasehar M.C.A., Lecturer,** Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, for being supportive throughout the tenure of my project.

My heartfelt thanks and gratitude to **Mr. Venkataramanan T. S., Mr. Sudheer H.R., Mr. Raghavan Muthukrishnan., Business Banking Unit, Infosys Technologies Ltd,** for extending valuable suggestions, support and motivation throughout the project work.

I also take this opportunity to extend my sense of gratitude to all faculty members, non-teaching staffs of the Computer Science and Engineering department, Kumaraguru College of Technology, Coimbatore, for their guidance and cooperation rendered throughout my project.

SYNOPSIS

The information is the back bone of any organization. Therefore, it has to be made available at all times to ensure proper decision-making and towards this end; information has to be accurate, current, timely and usable.

This project is developed for banking purposes and it involves the creation of a tool to curl out business logic from legacy code. The project was undertaken at Infosys Technologies LTD, Bangalore. Infosys is at the forefront of driving and managing IT strategies in the financial services industry for our various clients. We provide a range of consulting and IT services for the world' leading financial services firms - including banks and insurance companies.

The aim of the team is to deliver the product "FINACLE" Which provides end to end solutions to banking transactions. The expected end users are the bank tellers and the administrative personnel.

The Banking Business Unit has developed Finacle - Infosys suite of solutions comprising Finacle Core Banking, Finacle eChannels, Finacle eCorporate, Finacle CRM and Finacle Treasury. Since their rollout, the banking solutions have gone from strength to strength and today these solutions are deployed at several banks spanning the globe, providing them the benefits of cutting edge technology and rich functionality.

The Business Banking Unit (BBU) of the company has undertaken the reengineering process of the UI screens from an indigenous language (modified SQL*forms language) called as bas files to JSP architecture, which reduces the Complexity of coding.

The output generated by the bas files are a set of forms with customer data. The task is to generate a tool which reads each field and provides details related to the corresponding field. This assist engineers in reengineering process to verify fields with valid values for converting the bas files into JSP architecture.

The system is implemented in the UNIX environment using C and is developed to de used in UNIX and windows platforms.

The tools LEX and YACC provide a complete translation of bas files to a simpler and more understandable format. This will provide the engineers get a clear view of UI screens that is developed using bas files, which makes it easier for them to develop the same in the JSP architecture.

CONTENTS

1. INTRODUCTION

1.1 PROJECT OVERVIEW

This project is developed for banking purposes and it involves the creation of tools to curl out business logic from legacy code. This project comprises of three tools.

- Bas File Generator
- Bas Field Illustrator

The Banking Business Unit has developed Finacle - Infosys suite of solutions comprising Finacle Core Banking, Finacle eChannels, Finacle eCorporate, Finacle CRM and Finacle Treasury.

The TBA Forms are data input platform to FINACLE. These forms are similar to SQL forms in functionality but give more control to developer. Evolved as a replacement for ORACLE SQL*Forms. Facilitates easier design of screens, faster than SQL*Forms. TBA forms have direct interaction with DB reduced.

Forms Manager is the controlling program, which uses a binary input file (with extension **.trm**) and produces the desired TBA form at run time. In order to create this input file, a developer has to write an ASCII text file (with extension **.bas**).

UI screens are written in an indigenous language (modified SQL*forms language) and are called as bas files. Bas files are compiled into binary code as part of application build. Compiled bas files are input files to TBA Forms manager.

Bas File Generator

The Bas File Generator is a tool generated to document the bas files. Compiled bas files are input files to Bas File Generator. This tool will parse the entire bas file and generate an output file. This output file will contain the detail description of each line in the bas file. This tool is generated to assists the engineers who are involved in the reengineering process.

Bas Field Illustrator

Bas Field Illustrator tool is generated to give detailed information of each field that is available in the bas file.

The bas file contains different types of fields. All the fields in the bas file will fall in one of the two types

- Protected fields – Defines the protected fields.
- Unprotected fields – Defines an ordinary unprotected field.

The arguments for each field will differ. Bas Field Illustrator will parse the field definition level and displays a window describing the details of all the arguments of the field. This will help reengineering team in developing UI screens in JSP architecture without spending their time in understanding the bas files.

1.2 ORGANIZATION PROFILE

Infosys Technologies Ltd. (NASDAQ:INFY) is a leading provider of IT consulting and implementation services to the world's finest organizations. Infosys provides complete end-to-end solutions for technology driven business transformation initiatives. Today, global sourcing is an accepted and thriving business model, and organizations are increasingly capitalizing on the advantages it offers.

The Global Delivery Model (GDM) pioneered and perfected by Infosys Over the last 20 years has made global strategic sourcing a reality today and forms the very core of Infosys' ability to respond to client needs on-time and on-budget. By doing work where it makes
The most business sense, Infosys' Global Delivery Model brings faster, superior quality solutions at minimum risk and optimum cost.

Infosys combines world-class business and IT consulting services with domain and technical expertise. Our ability to bundle services brings integrated sourcing to our clients, ensuring that you get the right mix of technology and services appropriate to your requirements.

Services include:

- IT and business consulting
- Systems integration and applications management
- Infrastructure management
- Business process outsourcing
- Product development

Keeping pace with technology

Infosys invests extensively in technology and domain research to understand how technology impacts your business as a whole.

Apart from this, strategic relationships with the world's leading companies and partnerships with industry leaders allows us to stay ahead of the technology curve.

This enables us to provide you the precise mix of technology and services for your business requirements. Expertise and experience brings you the latest advantages technology has to offer.

Superior quality. Excellence in delivery. Optimum cost.

Infosys understands that offering low cost without high quality is meaningless, and that providing even the best quality is futile without on-time delivery. To that end, Infosys was the first organization in the world to attain the now globally recognized CMM Level 5 IT services quality standard. Assignments are managed to comply with this standard, bringing the level of projects that are delivered on time and on budget to 96%, 70% better than the industry average.

World-class, secure infrastructure

Infosys works through a backbone of resilient, secure infrastructure spread across multiple locations. Communication links through a number of service providers via several different paths ensures that your business is safe in any eventuality.

2. SYSTEM STUDY AND ANALYSIS

2.1EXISTING SYSTEM

The product "FINACLE" is a banking software, which provides end to end solutions of the banking transaction.

The UI Screens were developed using bas files now there is a reengineering process taking place to convert the same to JSP based Architecture.

Bas Files

TBA Forms are data input platform to Finacle and evolved as a replacement for ORACLE SQL* FORMS. This facilitates easier design of screens. The user interface screens in Finacle are written in an indigenous language and are called as bas files.

Forms Manager is the controlling program, which uses a binary input file (with extension **.trm**) and produces the desired TBA form at run time. In order to create this input file, a developer has to write an ASCII text file (with extension **.bas**). A .bas file is an ASCII text file, which contains the definition of an interface form or screen.

A .bas file is an ASCII text file, which contains the definition of an interface form or screen. The definition involves the following:

1. Physical layout - the external appearance of the form; literal, borders, graphic attributes etc., and
2. Navigation and operational logic - the manner in which the form behaves during its execution; control flow, validations, display of information etc.

The .bas file should be converted to a binary format to make it readable to the forms manager TBA*Forms. This conversion involves two steps:

1. Generation of .**trd**, an intermediate file. This is done by the TBA*Forms utility **gtrd**.
2. Conversion of **.trd** file to **.trm** format, done by the TBA*Forms utility **trdm**.

JSP BASED ARCHITECTURE

A part of the Finacle suite of products, use JSP (Java Server Pages) for the user interface. The ARJSP utilities library was developed to assist developers in JSP based development. ARJSP is a part of the Archie SOAR libraries. This document describes the ARJSP libraries and facilities available therein.

Java Server Pages (JSP) is a Sun Microsystems specification for combining Java with HTML to provide dynamic content for Web pages.

2.2 PROPOSED SYSTEM

Currently the UI Screens are created using the JSP architecture. There is a reengineering team formed to convert the UI screens developed in bas files to JSP. The engineers in the team do not have knowledge of bas files.

We have developed a toolkit which will curl out business logic from the legacy code. This tool kit consists of a Bas File Generator and a Bas Field Illustrator. The Bas file Generator creates a text document which will give a detailed description of the selected bas file.

The Bas Field Illustrator generates a UI Screen and gives detailed information of a particular field in the UI when it is selected.

2.3 USER CHARACTERISTICS

As far as the toolkit to curl out business logic from legacy code is concerned, the users are classified into two categories.

- The developer's team who creates the UI (User Interface) screen uses the Bas File Generator tool.

- The backend team uses the Bas Field Illustrator tool.

3. PROGRAMMING ENVIRONMENT

3.1 HARDWARE REQUIREMENT

● Processor	- 366MHz Pentium III
● Hard disk	- 40 GB
● RAM	- 256 MB
● Keyboard	- 104 keys Keyboard
● Mouse	- A scroll mouse
● Monitor	- A 14" Color Monitor

3.2 SOFTWARE REQUIREMENT

● Tools Used	- LEX , YACC
● Language Used	- C
● Operating System	- UNIX

3.3 ABOUT THE SOFTWARE

LEX AND YACC

INTRODUCTION

Programming in UNIX environment consumes a length of time, we have encountered the mystical programs Lex and YACC, or as they are known to GNU/Linux users worldwide. These programs are massively useful, but as with your C compiler, their manpage does not explain the language they understand, nor how to use them. YACC is really amazing when used in combination with Lex.

When properly used these programs allow you to parse complex languages with ease . This is a great boon when you want to read a configuration file, or want to write a compiler for any language you might have invented.

LEX

The first phase in a compiler reads the input source and converts strings in the source to tokens. Using regular expressions, we can specify patterns to lex that allow it to scan and match strings in the input. Each pattern in Lex has an associated action. Typically an action returns a token, representing the matched string rather than return a token value.

- First phase in a compiler
- Reads the input source and converts strings in the source to tokens
- Tokens are numeric representations of strings
- As lex finds identifiers in the input string, it enters them in a symbol table
- All subsequent references to identifiers refer to the appropriate symbol table index.
- Lex generates C code for Lexical Analyzer.

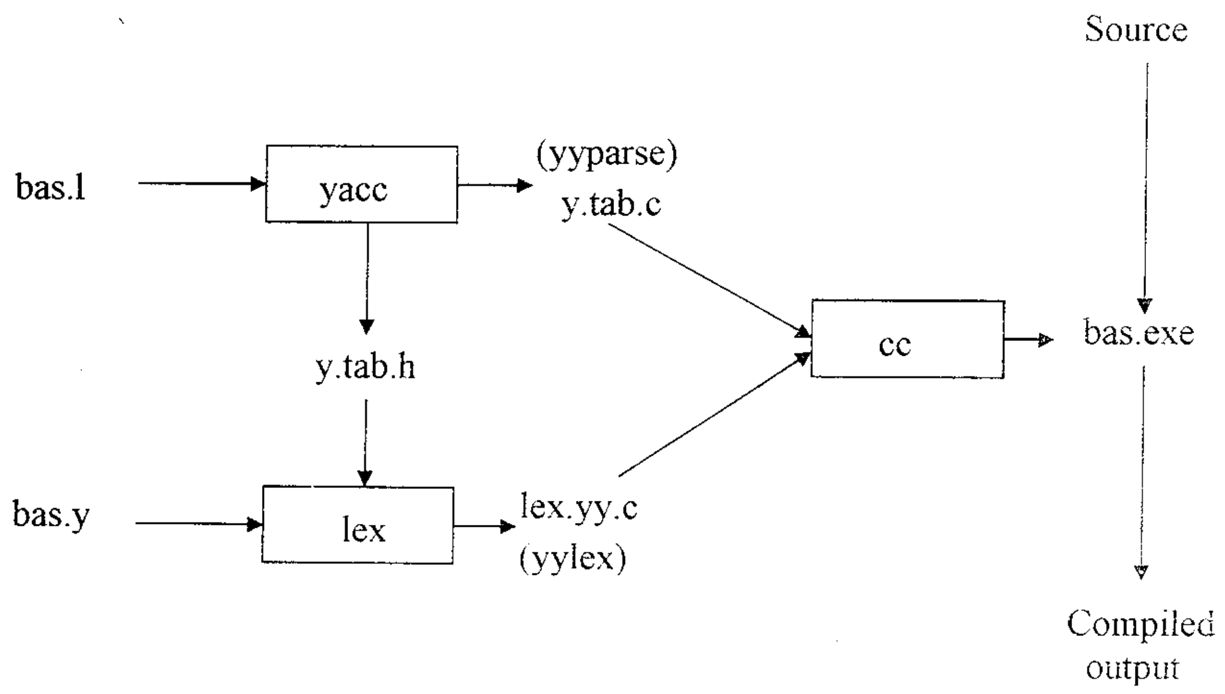
YACC

Grammars for yacc are described using a variant of Backus Naur Form (BNF). This technique was pioneered by John Backus and Peter Naur. A BNF grammar can be used to express context-free languages. Most constructs in modern programming languages can be represented in BNF.

- Uses grammar rules to analyze tokens from lex and create a syntax tree.
- A syntax tree imposes a hierarchical structure on tokens.

- The next step, code generation, does a depth-first walk of the syntax tree to generate code.
- Yacc generates C code from syntax analyzer.

BUILDING COMPILER USING LEX AND YACC



- Yacc reads the grammar descriptions from bas.y and generates a parser, function yyparse, in file y.tab.c
- Token declarations are also included in file bas.y
- Yacc convert token declarations into constant definitions and put them in file y.tab.h
- Lex reads the pattern declarations in bas.l, includes file y.tab.h and generates a lexical analyzer, function yylex, in file lex.yy.c

- Finally, the lexer and parser are compiled and linked together to form the executable, bas.exe
- From main, yyparse is called to run the compiler and yyparse calls yylex to obtain each token

C LANGUAGE

C is a general-purpose programming language. It has been closely associated with the UNIX system, since it was developed on that system, and since UNIX and its software are written in C.

The language, however, is not tied to any one operating system or machine, although it has been called a "system programming language" because it is useful for writing operating systems, it has been used equally well to write major numerical, text-processing, and database programs.

C is a relatively "low level" language. This characterization is not pejorative, it simply means C deals with the same sort of objects that most computers do, namely characters, numbers, and addresses. These may be combined and moved about with the usual arithmetic and logical operators implemented by actual machines.

Finally, C itself provides no input-output facilities; there are no READ or WRITE statements, and no wired-in file access methods. All of these higher-level mechanisms must be provided by explicitly-called functions.

A compiler for C can be simple and compact. Compilers are also easily written, using current technology, and expect to prepare a compiler for a new machine in a couple of months, and to find that 80 percent of the code of a new compiler is common with existing ones.

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 INPUT DESIGN

Input design is the process of converting user oriented inputs to computer based format. Erroneous data entered by the user can be controlled by the input design. If the input data given to the system is wrong, then the processing may lead to incorrect output.

GUIDELINES

- Only register data is collected and similar data are grouped.
- Exception handling is properly provided.
- Screen design should be clear.
- Input through keyword should be minimal.
- Proper validations must be done.
- User help must be provided wherever necessary.

The user can make desired changes before the data is sent for processing. Screens have been designed in C in the UNIX platform.

The following are some of the constraints used in the input design.

- Specifying maximum length of each field.
- Specify the format for the data field, which are entered.
- Listing the values, where necessary.

4.2 OUTPUT DESIGN

Output requirements have been designed during system analysis. A good starting point for output design is the data flow diagram (DFD). Human factors or end-users issues for design involve addressing internal controls to ensure readability and distribution of outputs generated by the computer.

The output design principles are

- Computer outputs should be straightforward.
- Every report or Screen should have a title.
- Not all the users are allowed to view the output
- The distribution of all outputs must be specified.
- The timing and volumes of each output must be specified.

OUTPUT DATA FORMAT CONSIDERATION

Output Labeling

The output should be clearly and correctly labeled to ensure that the user has understood what is being reported.

Report Separation

Different categories of reports must be properly separated from each other to enhance the readability and recognize the ability of the report.

Header and Footers

The starting of the report should be identified by a proper header, which should be highlighted. It may appear on a page by itself, which may form the cover page of the report.

Output Tabulation

Data is more appropriate when presented in a tabular form. This allows the user to analyze all or at least some number of related data at the same time.

Highlighting

The practice of focusing on important or exceptional piece of data in an output and including techniques like underlining, capitalizing, shading, etc., enhances the report.

Report Summary

A report summary should always be terminated with a summary page. Often, only the top management reviews the summary page. The content of this page is naturally based on environment.

Bas File Generator

This tool is used to generate a text document, which explains the detailed information of
the following:

- The name of the bas file.
- The Block level definitions.
- The Field level definitions.
- The literal section.

BLOCK LEVEL DEFINITIONS

Block level definitions describe the Block name, logical group name and block level triggers.

FIELD LEVEL DEFINITIONS

Field level definition describes the field name, field attributes, field level triggers and user exit functions.

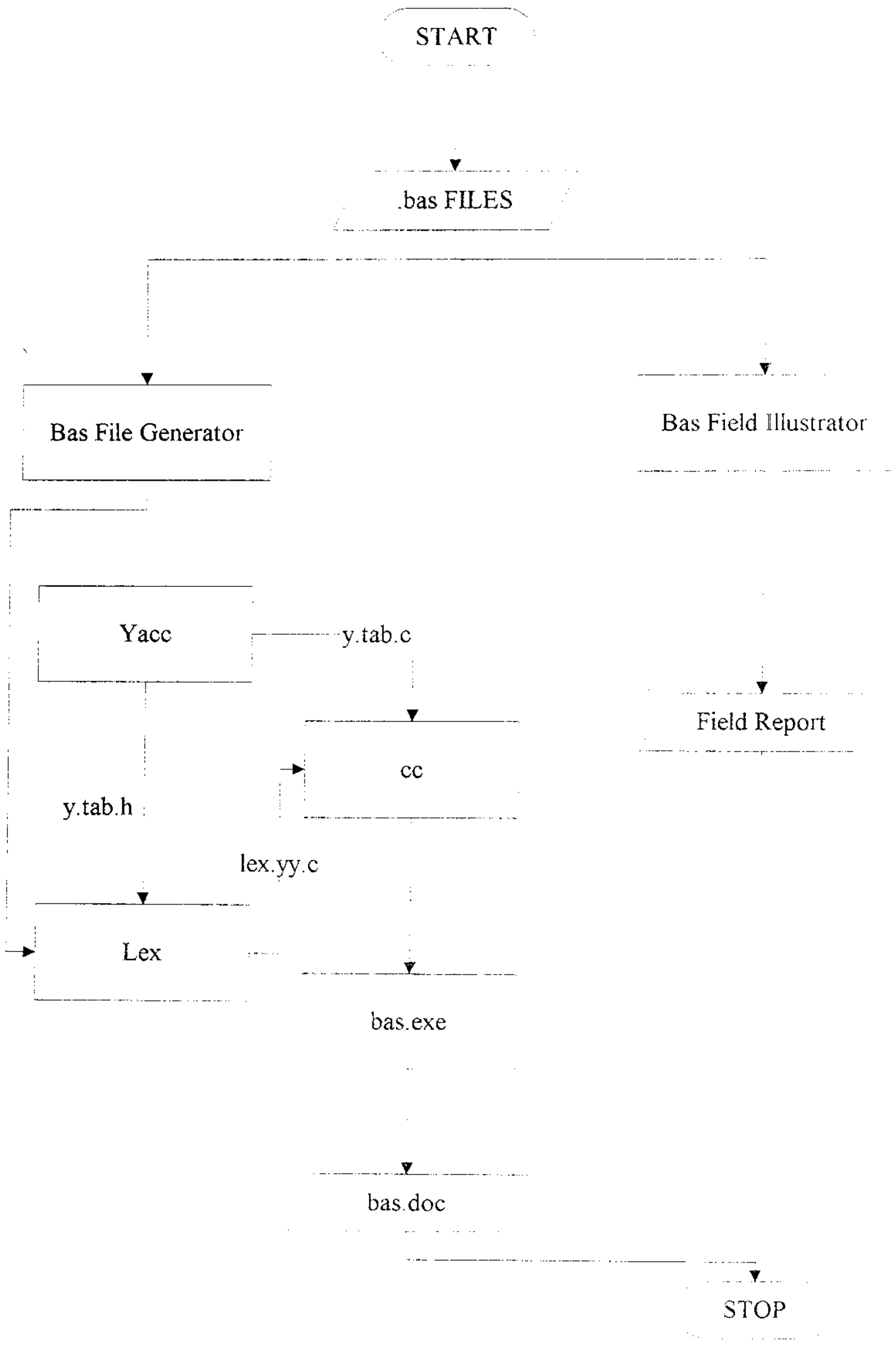
LITERAL SECTION

Literal section uses a set of syntax codes to design the output screens of the bas files.

Bas Field Illustrator

This is a tool which generates detailed information of any selected field in the field definition level of a bas file. The reengineering team will be provided with a consolidated description of the selected field in the UI screen.

4.3 FLOW CHART



5. SYSTEM TESTING & IMPLEMENTATION

5.1 SYSTEM TESTING

Software testing is an important element of software quality assurance and represents the ultimate review of specification, design and coding. System testing makes a logical assumption that if all parts of the system are correct, the goal will be achieved easily. The logical design and the physical design should be thoroughly and continually examined on paper to ensure that they will work when implemented.

When the programmers have tested each program with the test data designed by them, and have verified that these programs link together in the way specified in the computer run chart to produce the output specified in the program suite specification, the complete system and its environment must be tested to the satisfaction of the system analyst and the user.

Objectives of testing

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has high probability of finding a yet undiscovered error. Testing demonstrates that software functions work according to specifications. In addition, data collected from testing provides a full indication of software reliability and some indication of software quality. Testing results in the deduction of a number of errors. Critical modules are tested as early as possible.

Software testing for the tools generated has been done during the Pre-implementation stage using various software testing strategies and they are discussed below.

TESTING METHODS

Unit Testing

Unit testing focuses verification effort on the smallest unit of software design. The tests that occur as part of unit testing are illustrated below. The module 'interface' is tested to ensure that information properly flows into and out of the program unit under test. The local data structures are examined to ensure that data stored temporarily maintains integrity during all steps in an algorithms execution.

Boundary conditions are tested to ensure that all modules operate properly at boundaries established to limit to restrict processing. All 'independent paths' through control structures are exercised to ensure that all statements in a module have been executed at least once. Finally all error-handling are tested.

Integrating Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to unit-test modules and builds a program structure that has been dictated by design. Top-Down Integration is tested finally.

System Testing

System testing is series of different tests whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all the system elements have been properly integrated and perform allocated functions.

5.2 SYSTEM IMPLEMENTATION

A crucial phase in the system's lifecycle is the successful implementation of the new system design. Implementation is the stage of project when the theoretical design is turned into a working system. Implementation involves creating computer compatible files, training the operating employees, installing hardware before the system is up and running. A critical factor in conversion is not disrupting the functioning of the organization.

In system implementation, user training is crucial for minimizing resistance to change and giving the new system a chance to prove its worth. The training aids include user manuals, help screens, data dictionary, job aids, etc.

There are three types of implementation:

- Implementation of a computer system to replace a manual system.
- Implementation of a new computer system to replace an existing one.
- Implementation of a modified application to replace an existing one, using the same computer.

Software development is incomplete without any documentation. Documentation for the newly developed system is provided to satisfy the following needs.

- Protect the system when personnel are promoted, transferred or leave.
- Represents long-term money saving because it reduces the cost of training.
- Eases system maintenance by centralizing materials describing the system
- Provides a permanent reference of the system.

System Maintenance

The process of changing a system after it has been delivered and is in use is called software maintenance. There are three types of system maintenance.

- Corrective Maintenance

  It is concerned with fixing reported errors in the software.
- Adaptive Maintenance

  It means changing the software to some new environment such as a hardware platform for use with a different operating system
- Perfective Maintenance

  It involves implementing new functional or non-functional system requirements.

6. CONCLUSION AND SUGGESTIONS

We are delighted to tell you that, a toolkit to curl out business logic from a legacy code was a successful venture. The development of the toolkit has made us dwell into the intricacies of software engineering and software development.

We have tried to implement the latest practices in the software engineering and have undertaken various testing paraphernalia's. The toolkit has lead us to take deeper insights into Analysis, Design, Coding, testing, implementation and maintenance.

The toolkit comprises of two tools, Bas File Generator and Bas Field Illustrator. These tools are beneficial for the reengineering team in understanding bas files and converting them to the JSP Architecture.

- It simplifies the operation.
- It reduces the processing time.
- Help messages to operate the system
- Portable and flexible for further enhancement

7. SCOPE FOR FURTHER DEVELOPMENT

The toolkit assists in generating a quick and detailed description of bas files. This will provide the reengineering team with adequate information of the bas files to convert them to JSP Architecture.

Proper documentation has been made. The coding with enough comment statements makes the program self explanatory. This helps in adding or removing new modules to the system. In future we have plans of adding the parsing techniques for other indigenous languages of the Business Banking unit which will reduce the burden of the members in the reengineering team.

8. BIBLIOGRAPHY

1. R.Nigel Horspool , “ The BERKELY UNIX Environment”, Prentice-hall Canada Inc , 1992.

2.Brain W.Kernighan , Rob Pike, “ The UNIX Programming Environment”, Prentice-hall India Inc , 1998.

3. Brain W.Kernighan, Dennis M.Ritchie, “ The C Programming Language” , Bell Telephone Inc ,1978.

4. John Valley , “ C Programming for UNIX” , Prentice-hall India Inc , 1995.

5. John R.Levine , Tony Mason , Doug Brown , “Lex & Yacc” O’REILLY &Associates Inc , 1990.

6. Marc J.Rochkind , “Advanced UNIX Programming” , Prentice-hall Inc , 1985.

9. APPENDIX

SAMPLE INPUT FILE

```
;          BANCS2K FORMS
;*****************************************************************
; Name      :  BAFL2010.BAS
; Function  :  Inquiry on Discretionary Advances
; Remarks   :
;*****************************************************************
; Application Title:
bafl2010

;ORACLE workspace size :
; *********** Form Level Triggers ******************8
;
$$$include pre_form.h
$$$include form_keys.h
$$$include val_triggers.h
;
;
@@@fnkey_hlp_type   "'FUNCBLK',  'FUNCBLK',  'CRTBLK',  'CRTBLK',  'LISTBLK',
'LISTBLK'"
;
; ****** PRE-FORM trigger *********
;
; This trigger extracts the key fields from global.list_code
; into dummyblk key fields, which will be used in the query
;
@@@form_trg_hdr KEY-STARTUP
@@@trg_user_exit baol2010 0
$$tbaf_setup_solid_fields crtblk.set_id
; This will populate the user's home sol id into the set Id field.
$$val_sol_id "crtblk.set_id crtblk.set_id_desc"
; This is only to get the desc of the sol set id.
@@@copy_fld "'N'" pg0_virtual_flg
$exemacro case pg0_bafl2010_flg is
    when 'N' then goblk crtblk  ; goblk crtblk ;
    when 'Y' then goblk listblk ; exeqry ;
    end case ;
$$$include  int_stmt.h
$$$include trg_null.h
;
@@@form_trg_hdr TRG_INIT
@@@trg_user_exit baol2010 1
$$$include trg_null.h
;
@@@form_trg_hdr TRG_GETDATA
@@@trg_user_exit baol2010 2
$$$include trg_null.h
;
@@@form_trg_hdr TRG_PRE_BLOCK
@@@trg_user_exit baol2010 6
$$$include trg_null.h
;
@@@form_trg_hdr PRE-QUERY
```

```
@@@trg_user_exit baol2010 E
$$$include trg_null.h
;
@@@form_trg_hdr POST-QUERY
@@@trg_user_exit baol2010 G
$$$include trg_null.h
;
@@@form_trg_hdr TRG_EXPLODE
@@@invoke_form     "acct_tod_explode   listblk.acid   crtblk.status   crtblk.event_type
listblk.discret_advn_srl_num"
$$$include trg_null.h
;
; ************ Block Level Definitions *************
;
;
@@@blk_desc FUNCBLK func
;
@@@fld_pg0 pg0_bafl2010_flg CHAR 1
;
@@@fld_pg0 pg0_user_name CHAR 40
;
@@@fld_pg0 pg0_rec_type CHAR 2
;
@@@fld_pg0 pg0_crncy_desc CHAR 25
;
@@@fld_pg0  pg0_curr_sol_id_desc CHAR 25
;
@@@fld_protect funcblk header_date DATE 10 1 2 70
;
```

```
@@@fld_protect funcblk header_date_2 DATE 10 2 2 70
;
@@@fld_protect funcblk header_title CHAR 40 1 2 21
;
@@@fld_protect funcblk header_title_2 CHAR 40 2 2 21
;
@@@fld_protect funcblk header_subtitle CHAR 40 1 3 21
;************************************
; FIELD DESCRIPTION
;************************************
;
$$$include pg0_flds.h
$$$include pg0_std_flds.h
$$$include pg0_acct_flds.h
@@@fld_pg0 pg0_option CHAR 1
@@@fld_pg0 pg0_virtual_flg CHAR 1
;
@@@fld_unprot1 crt crtblk set_id "CHAR 8 1 5 24 N" Y
@@@val_pchg val_set_id "crtblk.set_id crtblk.set_id_desc"
@@@fld_trg_hdr KEY-F0
@@@gen_lst_choice bafl7296 "crtblk.set_id 8" N crtblk.set_id_desc
$$$include trg_null.h
@@@fld_unprot2 Y T75
;
@@@fld_protect crtblk set_id_desc CHAR 25 1 5 33
;
@@@fld_unprot1 crt crtblk acct_num "ACCOUNT 0 1 6 14 N" N
@@@val_pchg val_acct_id "Y crtblk.acct_sol_id crtblk.acct_num crtblk.crncy_code
```

```
crtblk.acct_name B N"
$$$include  trg_null.h
@@@fld_trg_hdr KEY-F0
@@@copy_fld "'1'" crtblk.pg0_option
@@@gen_lst_choice bafl0096 "crtblk.acct_num 16 crtblk.crncy_code 3 crtblk.acct_sol_id 8
crtblk.pg0_option 1" N crtblk.acct_na
me
$$$include    trg_null.h
@@@fld_unprot2 Y D12
;
```

SAMPLE CODE

```
/*
-------------------------------------------------------------------
Program header: se_ fldol.c
-------------------------------------------------------------------
Program: se fldOl.c
System: BANCS2000
Sub-system: TBAFORMS - FORMS MANAGEMENT LAYER
Description: contains main of se_fld
-------------------------------------------------------------------
*/
#include "se fld.h"
extern int fld_InitMainIOStruct (MAIN_IOSTATUS_PTR MainIOSPtr);
extern int fld_CleanupAndExit( int iCode, MAIN_IOSTATUS_PTR MainIOSPtr);
extern int fld- GetAndValInput (int argc, char *argv[], MAIN_IOSTATUS_PTR
MainIOSPtr);
extern int fld_MainProcessLoop (MAIN_IOSTATUS_PTR IOStatusPtr);
extern int fld_InitializeCurses( void) ;
extern int fld_EndCurses (void);
extern int fld_PrintErrorStr(FILE *ERRfp, char *errStr) ;
extern int fld_PrintError(FILE *ERRfp, char *errCode) ;
extern int fld_DeletePages (MAIN_IOSTATUS_P"R IOStatusPtr);
externint fld- CloseTheLL(LL- PTR_TYPEinit_list-ptr, char * llname);
/***************************/
/*
/
Function: main
```

```
Input paras       :
Output paras      :
Globals affected :
Returns : SUCCESS/FAILURE
----------------------------------------------------------/
*/
int main ( int argc, char * argv[] )
{
int iRetVal = 0 ;
MAIN IOSTATUS MainIOStatus;
MAIN_IOSTATUS_PTR MainIOSPtr = &MainIOStatus ;

/* initialize the main io struct with proper values */
memset (MainIOSPtr, '\0' , sizeof(MAIN_IOSTATUS));
iRetVal= fld- InitMainIOStruct( MainIOSPtr);
if( SUCCESS != iRetVal)
{
fld- CleanupAndExit ( FAILURE, MainIOSPtr );
}
--------------------------------------------------------------/*
get and validate the input */
iRetVal = fld_GetAndValinput (argc, argv, MainIOSPtr);
if( SUCCESS != iRetVal)
{
fld- CleanupAndExit ( FAILURE, MainIOSPtr );
}
/* process forms one by one */
fld- InitializeCurses();
```

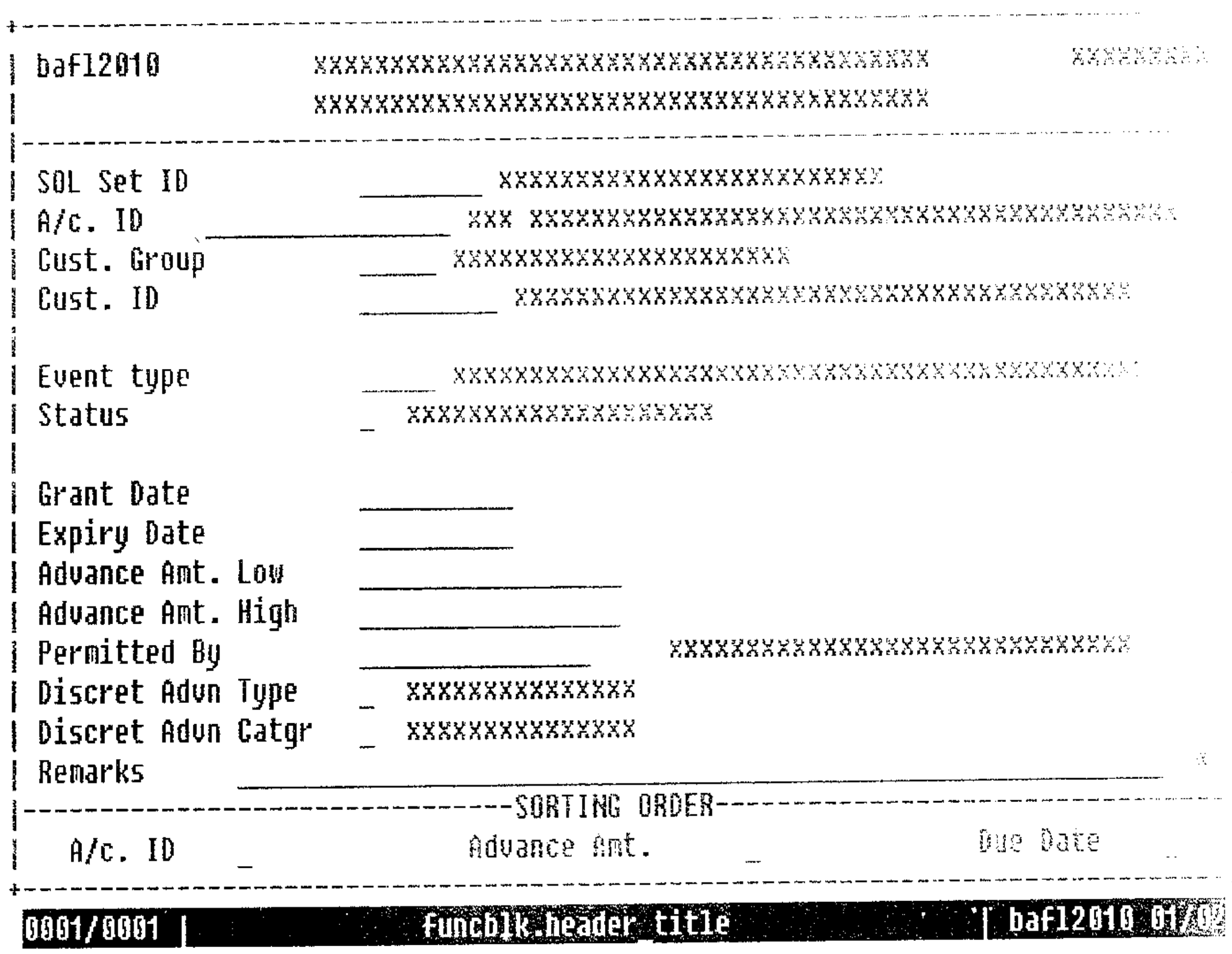
```
fld_ EndCurses();
if( SUCCESS != iRetVal)
{
fld- PrintErrorStr(MainIOSPtr-> ERRfp,MainIOSPtr->szErrBuf);
fld- CleanupAndExit ( FAILURE, MainIOSPtr );
}
/* success, clean up and exit */
fld_CleanupAndExit (SUCCESS, MainIOSPtr);
-return
(SUCCESS);
}/* main */
/*
---------------------------------------------------------------------------/
Function: fld InitMainIOStruct
Description. initializes main io status struct
Input paras . MainIOSPtr - pointer to main io status struct
Output paras:-
Globals affected: none
---------------------------------------------------------------------------/
*/
int fld_InitMainIOStruct (MAIN_IOSTATUS_PTR MainIOSPtr)
{
int iRetVal = 0 ;

/* clear the structure */
memset (MainIOSPtr, '\0', MAIN_IOS_SIZE);
/* initialize the main io struct with proper values */
```

```
MainIOSPtr->init_list~tr = NULL;
MainIOSPtr->ERRfp = NULL;
MainIOSPtr->iNumPages = 0 ;
MainIOSPtr->AllForms = 'N';
return (SUCCESS);
}/* fld_InitMainIOStruct */
/*

---------------------------------------------------------------------------/
Function : fld_ CleanupAndExit
Description: closes errfile, link
                lists etc, and then exits with code passed
               as parameter
Input paras: iCode - Code to return when exit
                        MainIOSPtr -pointer to main io status struct
Output paras:-
Globals affected: -
Returns: exits with code
---------------------------------------------------------------------------/
*/
int fld_CleanupAndExit( int iCode, MAIN_IOSTATUS_PTR MainIOSPtr)
{
/* close the err file, if it is opened */
if ( MainIOSPtr->ERRfp != (FILE *)NULL )
{
fclose(Mainl OSPtr ->ERRfp );
MainIOSPtr->ERRfp = (FILE *)NULL;
}
```

```
/* close the file list all */
fld- CloseTheLL (MainIOSPtr->init_listytr, FLD_FILE_LIST- LL);

/* close the crt list all */
CrtLLFree (MainIOSPtr->init_listytr);

/* clean up pages */
fld_DeletePages (MainIOSPtr);

exit (iCode);
}/* fld_CleanupAndExit */

/*
-----------------------------------------------------------------------/

Function: fld CloseTheLL
Description: if the 1 lexists in the given init_listytr, closes it
Input paras: init_listytr - pointerto init list ptr
             llname - link list name
Output paras:-
Globals affected: NONE
Returns : SUCCCESS
-----------------------------------------------------------------------/

*/
int fld_CloseTheLL (LL_PTR_TYPE init_listytr, char * llname)
{
if ( init_listytr !=NULL )
{
```

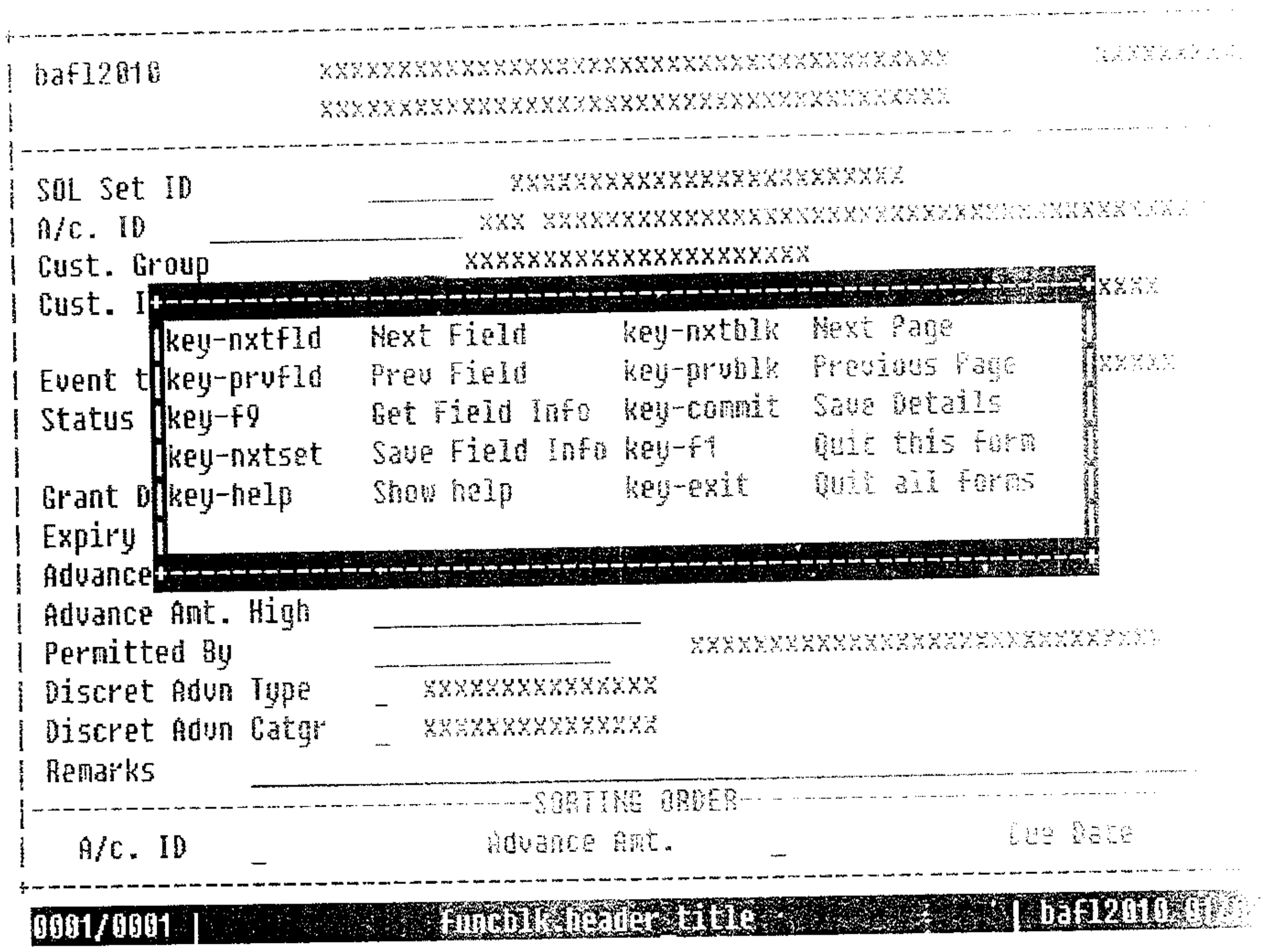
```
if( SUCCESS == shared_ll_exists( init_list-ptr,llname))
{
shared_ll_rtn(init_list_ptr, llname,
CLOSE,(PTR - TYPE)0, (PTR - TYPE)0 );
}
}
return(SUCCESS) ;
}/*CloseTheLL */
```

SAMPLE OUTPUT UI SCREENS



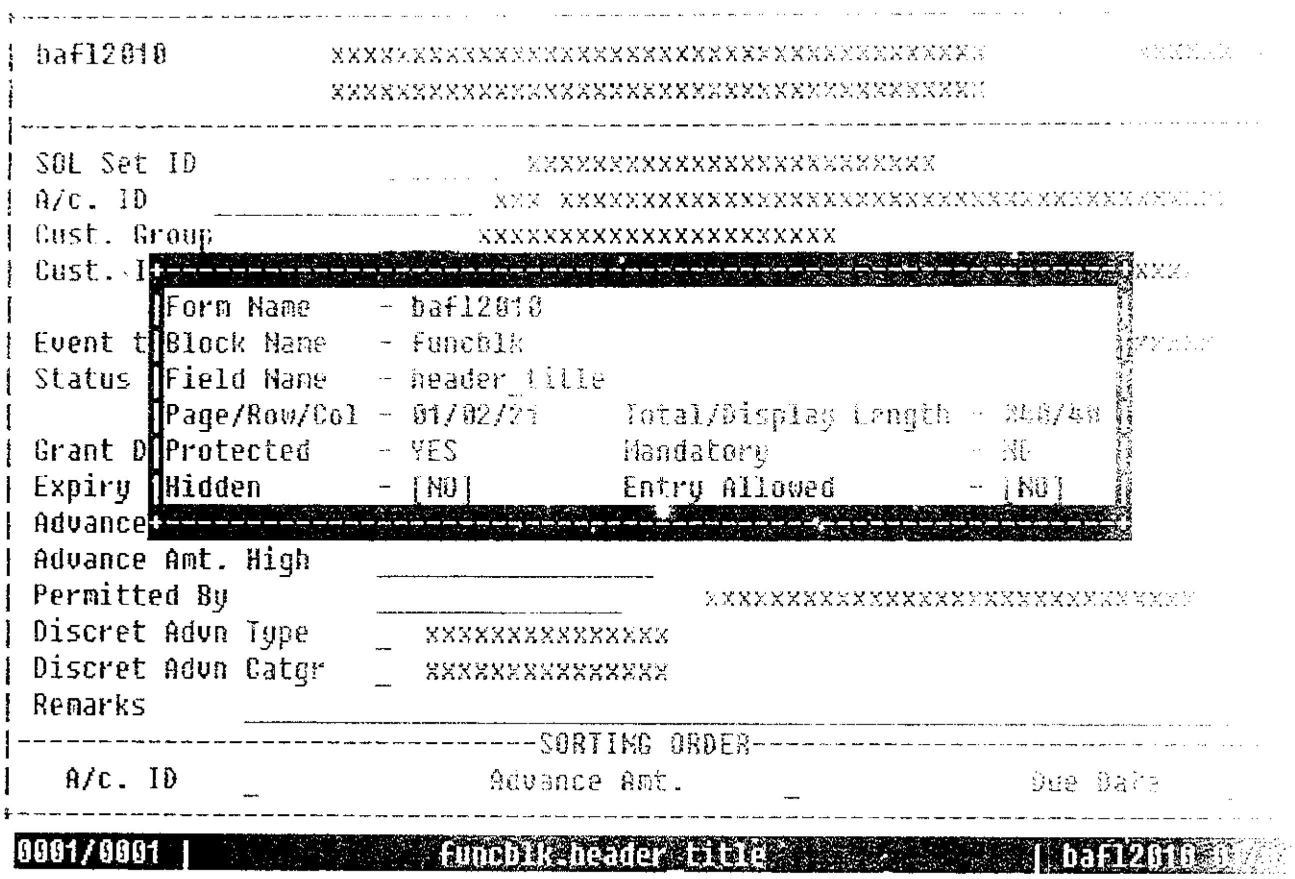
```
+------------------------------------------------------------------------------
| baf12010          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX       XXXXXXXXX
|                   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
|------------------------------------------------------------------------------
| SOL Set ID               ________  XXXXXXXXXXXXXXXXXXXXXXXXX
| A/c. ID        ________________  XXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
| Cust. Group              _____  XXXXXXXXXXXXXXXXXXXXXX
| Cust. ID                 _________  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
|
| Event type               _____  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
| Status                   _  XXXXXXXXXXXXXXXXXXXX
|
| Grant Date               __________
| Expiry Date              __________
| Advance Amt. Low         _________________
| Advance Amt. High        _________________
| Permitted By             _______________     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
| Discret Advn Type        _  XXXXXXXXXXXXXXX
| Discret Advn Catgr       _  XXXXXXXXXXXXXXX
| Remarks          ____________________________________________________________
|--------------------------------SORTING ORDER---------------------------------
|   A/c. ID    _                 Advance Amt.      _             Due Date     _
+------------------------------------------------------------------------------
0001/0001 |              Funcblk.header title              | baf12010 01/02
```

HELP MENU



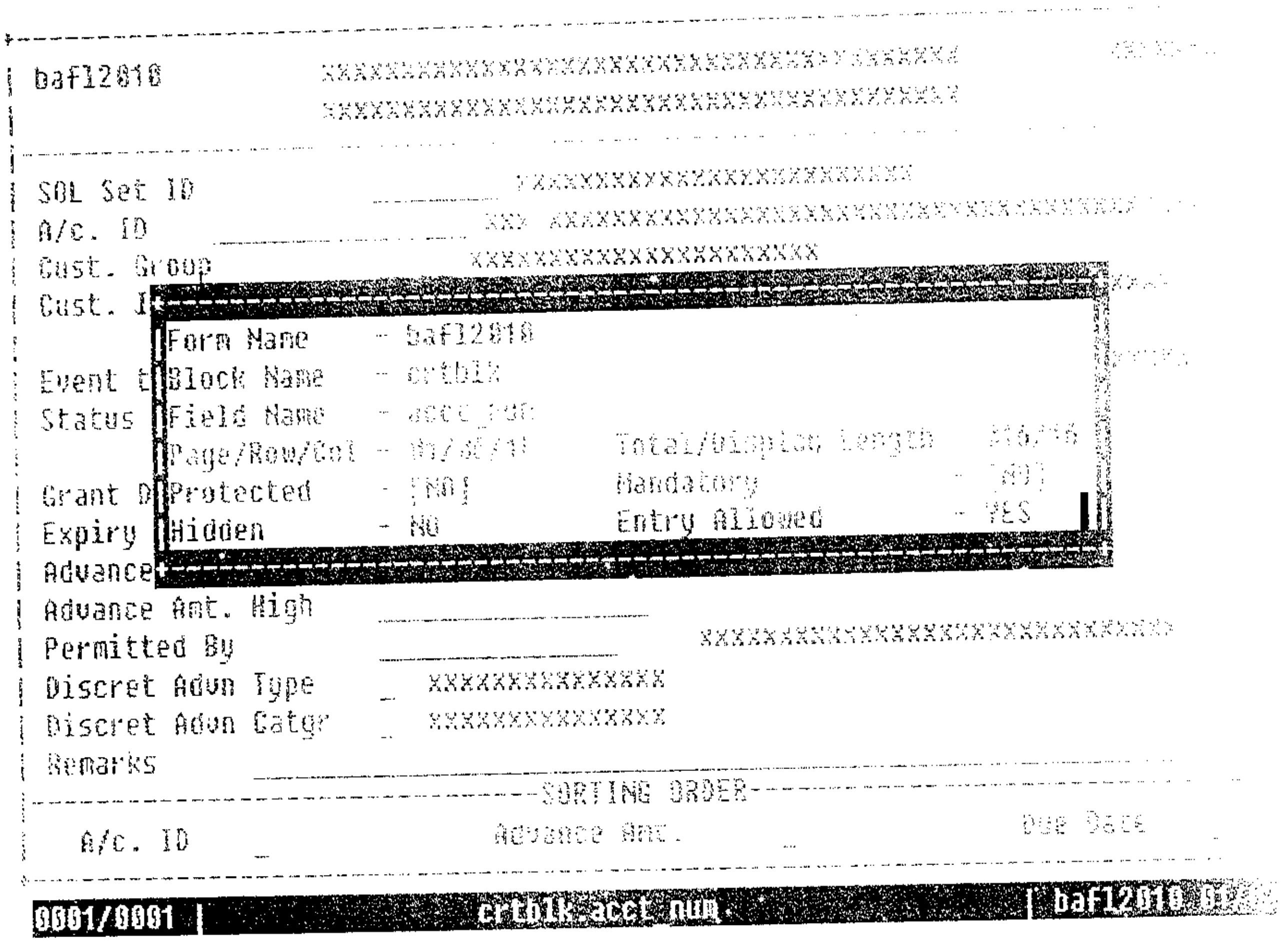
```
+------------------------------------------------------------------------------
| baf12010          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
|                   xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
|------------------------------------------------------------------------------
| SOL Set ID            ________  xxxxxxxxxxxxxxxxxxxxxxxxx
| A/c. ID         ________________  xxx  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
| Cust. Group                    xxxxxxxxxxxxxxxxxxxxxxxx
| Cust. I+-----------------------------------------------------------------+xxxx
|        |key-nxtfld    Next Field       key-nxtblk   Next Page          |
| Event t|key-prvfld    Prev Field       key-prvblk   Previous Page      |xxxxx
| Status |key-f9        Get Field Info   key-commit   Save Details       |
|        |key-nxtset    Save Field Info  key-f1       Quit this Form     |
| Grant D|key-help      Show help        key-exit     Quit all Forms     |
| Expiry |                                                               |
| Advance+-----------------------------------------------------------------+
| Advance Amt. High     _________________
| Permitted By          _______________          xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
| Discret Advn Type     _  xxxxxxxxxxxxxxx
| Discret Advn Catgr    _  xxxxxxxxxxxxxxx
| Remarks           _____________________________________________________________
|-------------------------------SORTING ORDER-----------------------------------
|   A/c. ID   _                Advance Amt.     _                   Due Date
+------------------------------------------------------------------------------
0001/0001 |            Funcblk header title            | baf12010
```

FIELD DESCRIPTION



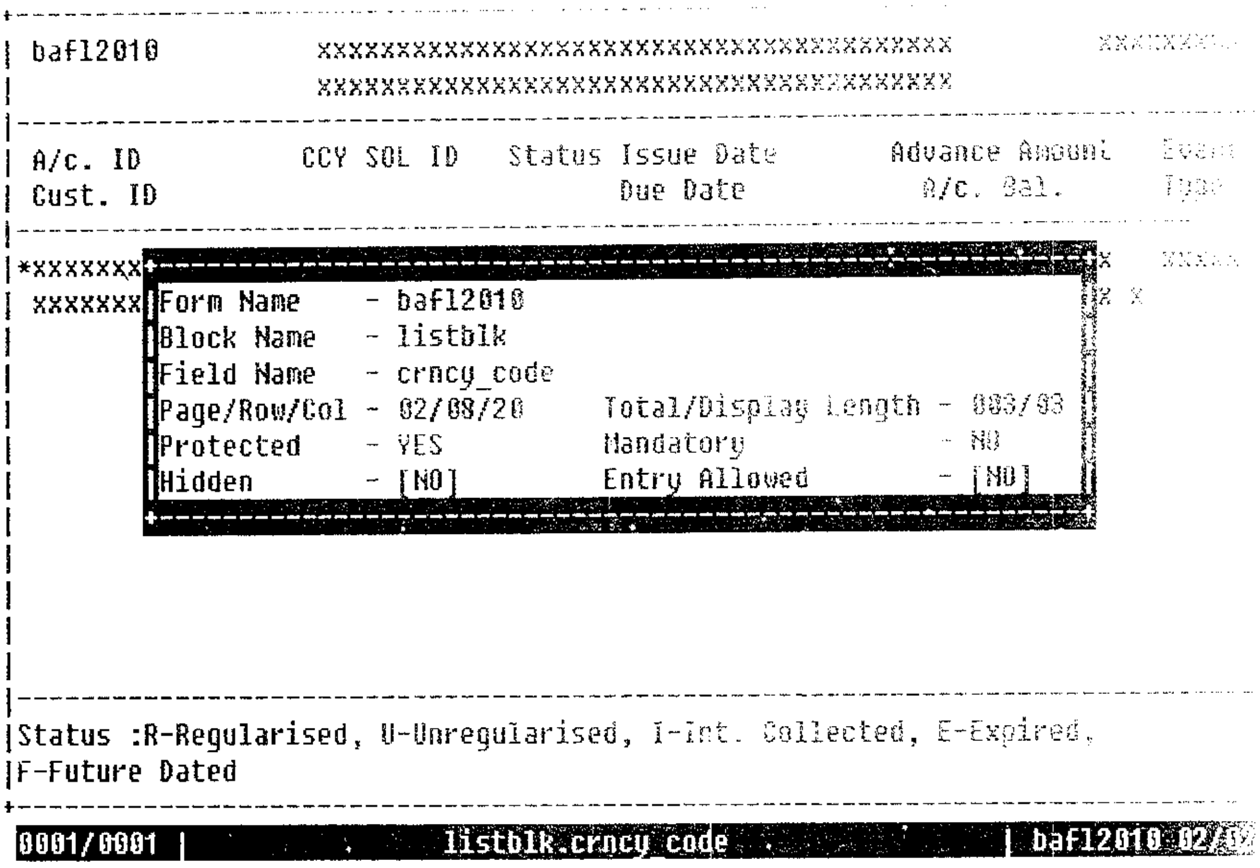
```
+------------------------------------------------------------------------------
| baf12010           xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
|                    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
|------------------------------------------------------------------------------
| SOL Set ID                   xxxxxxxxxxxxxxxxxxxxxxxx
| A/c. ID     ________________ xxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
| Cust. Group                 xxxxxxxxxxxxxxxxxxxxxx
| Cust. I+-----------------------------------------------------------------+xxx
|        |Form Name     - baf12010                                          |
| Event t|Block Name    - Funcblk                                           |
| Status |Field Name    - header_title                                      |
|        |Page/Row/Col  - 01/02/21       Total/Display Length  - 240/40     |
| Grant D|Protected     - YES            Mandatory             - NO         |
| Expiry |Hidden        - [NO]           Entry Allowed         - [NO]       |
| Advance+-----------------------------------------------------------------+
| Advance Amt. High       ______________
| Permitted By            ______________     xxxxxxxxxxxxxxxxxxxxxxxxxx
| Discret Advn Type     _   xxxxxxxxxxxxxxx
| Discret Advn Catgr    _   xxxxxxxxxxxxxxx
| Remarks           _______________________________________________________
|------------------------------SORTING ORDER-----------------------------------
|   A/c. ID  _                Advance Amt.   _                Due Date
+------------------------------------------------------------------------------
0001/0001 |                  funcblk.header_title                   | baf12010
```

FIELD DESCRIPTION – acct_num



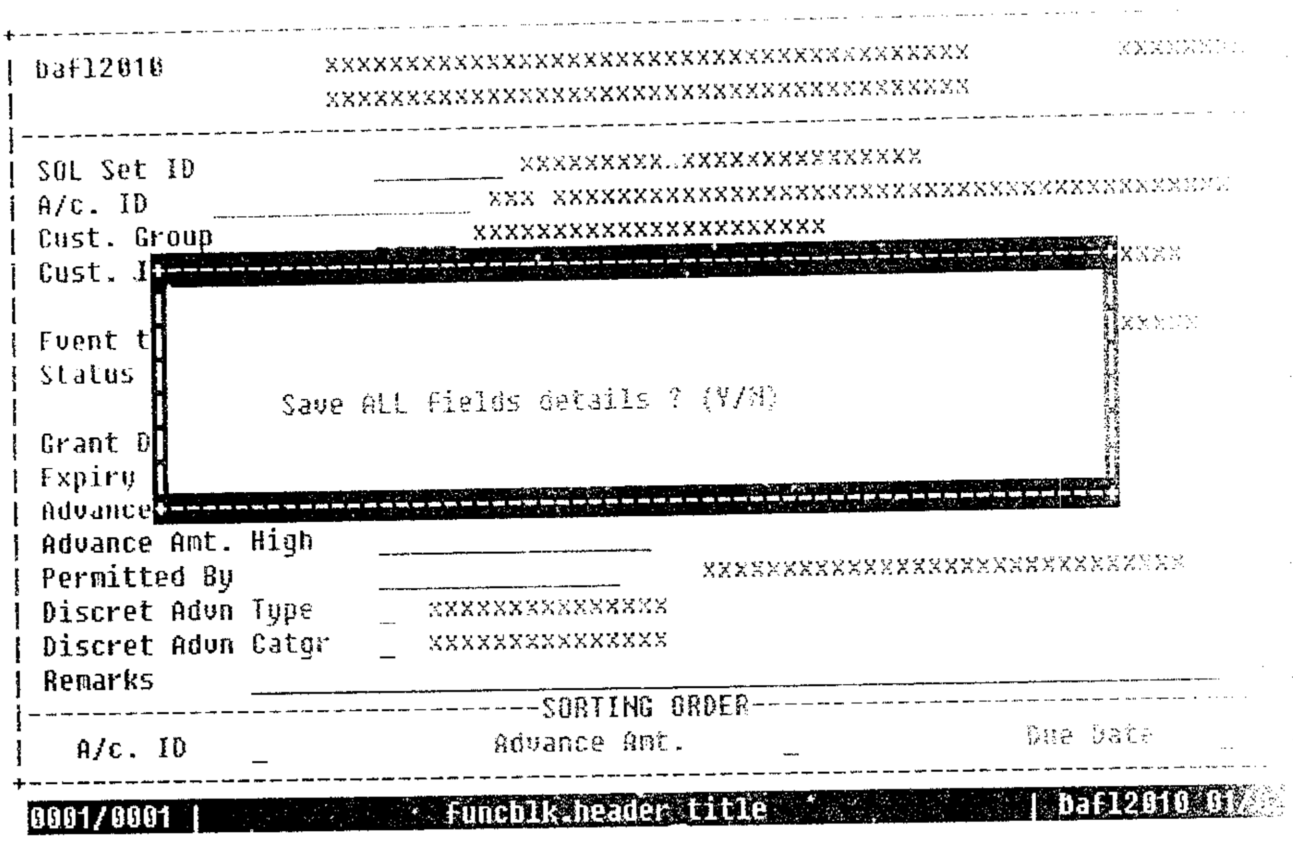
```
baf12010

SOL Set ID
A/c. ID
Cust. Group
Cust. I
+------------------------------------------------------------------+
|Form Name      - baf12010                                         |
|Block Name     - crtblk                                           |
|Field Name     - acct_num                                         |
|Page/Row/Col   - [illegible]    Total/Display Length  [illegible] |
|Protected      - [NO]           Mandatory           - [NO]        |
|Hidden         - NO             Entry Allowed       - YES         |
+------------------------------------------------------------------+
Event t
Status
Grant D
Expiry
Advance
Advance Amt. High
Permitted By
Discret Advn Type
Discret Advn Catgr
Remarks
-------------------------------SORTING ORDER--------------------------
  A/c. ID                    Advance Amt.                  Due Date

0001/0001 |            crtblk.acct_num.            | baf12010
```

FIELD DESCRIPTION – crncy_code



```
| bafl2010        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        xxxxxxxx
|                 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
|-------------------------------------------------------------------------------
| A/c. ID            CCY SOL ID   Status Issue Date        Advance Amount   Event
| Cust. ID                               Due Date           A/c. Bal.       Type
|-------------------------------------------------------------------------------
|*xxxxxxx+----------------------------------------------------------+x    xxxxx
| xxxxxxx|Form Name     - bafl2010                                  |x x
|        |Block Name    - listblk                                   |
|        |Field Name    - crncy_code                                |
|        |Page/Row/Col  - 02/08/20     Total/Display Length - 003/03 |
|        |Protected     - YES          Mandatory            - NO     |
|        |Hidden        - [NO]         Entry Allowed        - [NO]   |
|        +----------------------------------------------------------+
|
|
|
|
|-------------------------------------------------------------------------------
|Status :R-Regularised, U-Unregularised, I-Int. Collected, E-Expired,
|F-Future Dated
+-------------------------------------------------------------------------------
0001/0001 |                 listblk.crncy_code                | bafl2010 02/02
```

SAVE OPTION



```
+---------------------------------------------------------------------------------
| baf12010          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        xxxxxxx
|                   xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
|---------------------------------------------------------------------------------
| SOL Set ID             ________  xxxxxxxxx..xxxxxxxxxxxxxxx
| A/c. ID      _________________ xxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
| Cust. Group                     xxxxxxxxxxxxxxxxxxxxxxx
| Cust. I+-------------------------------------------------------------+xxxx
|        |                                                             |
| Fvent t|                                                             |xxxxx
| Status |                                                             |
|        |        Save ALL Fields details ? (Y/N)                      |
| Grant D|                                                             |
| Expiry |                                                             |
| Advance+-------------------------------------------------------------+
| Advance Amt. High      ________________
| Permitted By           _______________       xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
| Discret Advn Type     _  xxxxxxxxxxxxxxx
| Discret Advn Catgr    _  xxxxxxxxxxxxxxx
| Remarks           ______________________________________________________________
|----------------------------------SORTING ORDER-----------------------------------
|   A/c. ID    _                  Advance Amt.      _               Due Date
+---------------------------------------------------------------------------------
0001/0001 |            Funcblk.header title                    | baf12010 01
```