NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

DEVELOPMENT AND IMPLEMENTATION OF AN INFORMATION SYSTEM FOR THE NAVAL POSTGRADUATE SCHOOL INTERNATIONAL PROGRAMS OFFICE

by

Onder Celebi

September, 1996

Thesis Advisor:	James C. Emery
Associate Advisor:	Magdi N. Kamel

Approved for public release; distribution is unlimited.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1996	Master's Thesis

4. TITLE AND SUBTITLE **DEVELOPMENT AND IMPLEMENTATION OF AN INFORMATION SYSTEM FOR THE NAVAL POSTGRADUATE SCHOOL INTERNATIONAL PROGRAMS OFFICE**	5. FUNDING NUMBERS
6. AUTHOR(S) Celebi, Onder	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING /MONITORING ORGANIZATION REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The Naval Postgraduate School (NPS) International Programs Office's (IPO) mission requires timely, accurate, and intensive information exchange with the outside military and civilian agencies to accomplish the goals of the Security Assistance and the Information Programs. Therefore its information infrastructure is becoming a vital key to the organization success or failure in performing its mission-critical tasks. Currently the office achieves its goals to a great extent without taking advantage of an information system to support its administrative activities more efficiently.

This thesis conducts a thorough analysis and documentation of the information requirements of the Naval Postgraduate School International Programs Office. Based on the requirements identified, the thesis develops and implements a series of functional computer applications which supports the NPS IPO administrative activities.

14. SUBJECT TERMS Information System, IPOMIS, International Programs Office, Visual Development Methodology, Visual Development Tool.	15. NUMBER OF PAGES 260
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

DEVELOPMENT AND IMPLEMENTATION OF AN INFORMATION SYSTEM FOR THE NAVAL POSTGRADUATE SCHOOL INTERNATIONAL PROGRAMS OFFICE

Onder Celebi
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1990

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE
IN
INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 1996

Author: Onder Celebi
Onder Celebi

Approved by: James C. Emery, Thesis Advisor

Magdi N. Kamel, Associate Advisor

Reuben T. Harris, Chairman
Department of Systems Management

ABSTRACT

The Naval Postgraduate School (NPS) International Programs Office's (IPO) mission requires timely, accurate, and intensive information exchange with the outside military and civilian agencies to accomplish the goals of the Security Assistance and the Information Programs. Therefore its information infrastructure is becoming a vital key to the organization success or failure in performing its mission-critical tasks. Currently the office achieves its goals to a great extent without taking advantage of an information system to support its administrative activities more efficiently.

This thesis conducts a thorough analysis and documentation of the information requirements of the Naval Postgraduate School International Programs Office. Based on the requirements identified, the thesis develops and implements a series of functional computer applications which supports the NPS IPO administrative activities.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENT

Special acknowledgment must be given to my wife, Gonca, for her support and forbearance as I spent long hours at the computer working on this thesis. I am truly in her debt.

I. INTRODUCTION

A. OBJECTIVES

The objective of this research is to conduct a thorough analysis and documentation of the information requirements of the Naval Postgraduate School (NPS) International Programs Office (IPO). Based on the requirements identified, the thesis will develop a series of functional computer applications which shall support the NPS IPO administrative activities.

B. BACKGROUND

The Naval Postgraduate School International Programs Office is charged with interacting with outside military and civilian agencies to accomplish the goals of the Security Assistance Program (SAP) and the Information Program (IP). Additionally, it is responsible for the International Sponsor Program and acts as the Command Sponsor to the International Committee.

Since the NPS IPO's mission requires timely, accurate, and intensive information exchange, its information infrastructure is becoming a vital key to the organization success or failure in performing its mission-critical processes.

Although the NPS IPO currently achieves its goals to a great extent, it lacks an information system to support its administrative activities more efficiently. The information requirements that those are to be identified during the research phase of this thesis will bring out the issues associated with the new information system design for the NPS IPO. As a result of this thesis, a series of functional computer applications, to support the NPS IPO administrative activities effectively will be developed and implemented.

C. RESEARCH QUESTIONS

To attain the objectives of this thesis and gain insights into the information requirements of the IPO, the following research questions are defined.

- What are the information requirements of the Naval Postgraduate School (NPS) International Programs Office (IPO) to support its administrative activities?
    - How is the current information flow and data processing at the NPS IPO?
    - What are the concerns over the information system issues that motivate the new system development?
    - What are the feasibility issues regarding the new system development?
- How can the data and processes associated with the IPO's administrative activities be modeled with respect to the results of the system analysis?
- What will be the appropriate system design to meet the requirements identified during the requirement analysis?
- What development tools and methodology is appropriate for applications of the type developed for the IPO?

D. METHODOLOGY

A mixture of both Structured Development Methodology (SDM) and Visual Development Methodology (VDM) were followed for developing the information system required by the IPO.

Since there was no budget or time constraints for the system development and the first objective of this thesis was to conduct a thorough analysis of the information requirements of the IPO, preliminary investigation and requirements analysis were conducted by studying the discipline of the SDM. The discipline of this methodology is based on the idea that if developers can get perfect specifications up front, the end result will be a perfect application (Whitten et al. 1994). It puts a large emphasis on the initial requirements specification. But after the design specifications are complete, the

requirements are "frozen," and changes to the application's design are no longer allowed. This usually causes the SDM to fail to produce applications that meet users current needs. Therefore, in order to achieve the other objectives of this thesis, the VDM was employed to address the shortcomings of the structured approach.

The VDM is outlined in Figure 1. This development methodology encourages the developers to provide a basic application, which is then followed by incremental improvements and enhancements as users exercise the system, finding flaws, areas for improvement, and ideas for enhancement. It allows for changes to a system in a relatively responsive manner and it provides a means of allowing the business rules to drive the system, and not have the system drive the rules (Hodges 1995).

The VDM begins traditionally with the gathering of user requirements. As discussed before, requirements were determined and analyzed by employing the SDM in order to achieve thesis's objectives. However, based on the preliminary findings of the requirements analysis, basic data and process models were developed in such a way to allow for extension as emphasized in VDM.

Once the preliminary requirements were analyzed, prototypes of the system applications were developed in order to get feedback from end users and get them involved with the actual system design. The feedback of the end users were analyzed to refine the results of the analysis and to improve the prototypes to approach the final product.

E. CHAPTER OUTLINE

This thesis is organized as follows. Chapter II discusses the results of the preliminary investigation. It discusses the current information flow and data processing at the organization, the concerns over the information system issues that motivate the new system development, and the preliminary feasibility issues regarding the new system development.

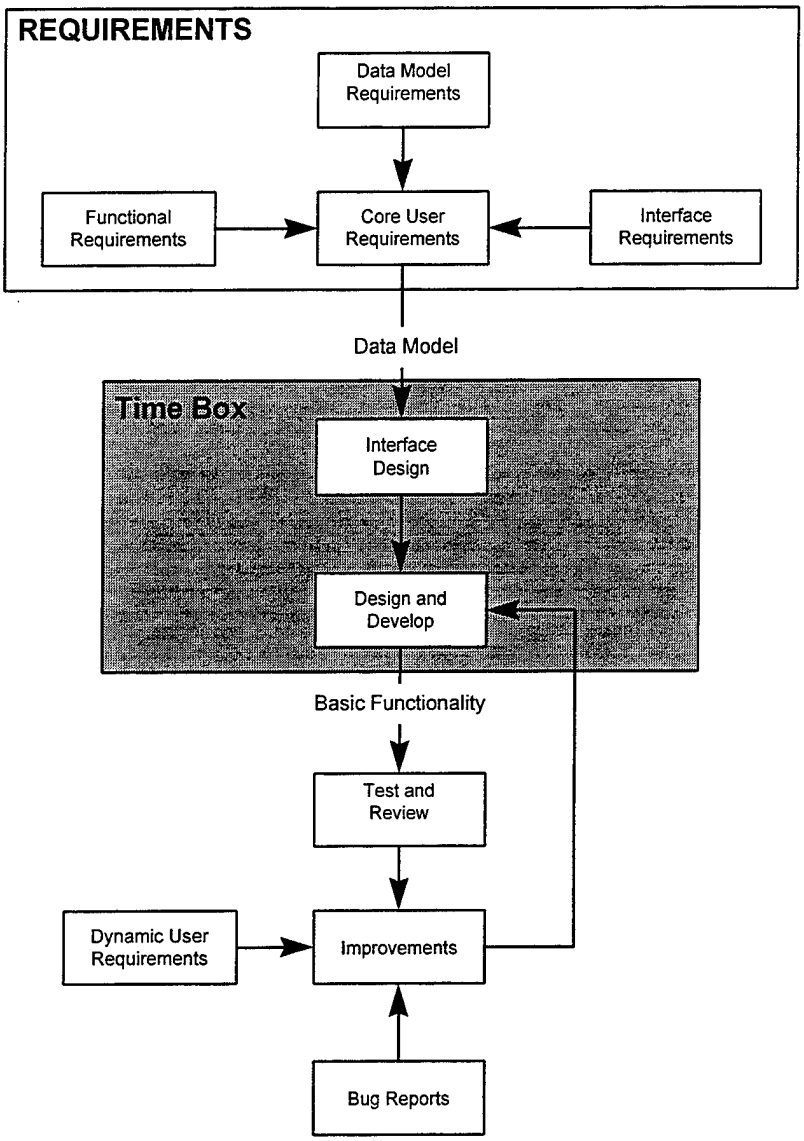


Figure 1. Visual Development Methodology

Chapter III describes the requirements analysis of the system. The analysis provides the conceptual data model which reflects the specific data requirements (objects) that must be represented in the new information system's database and the the process model which represents the application or functional requirements that support the new system.

Chapter IV discusses system design phase of the system development. Two components of the system design, logical database design and application design, are discussed. Issues associated with designing an appropriate system and selecting appropriate application development tool are addressed.

Chapter V discusses the implementation and maintenance phases of the system development.

Chapter VI draws conclusions and makes recommendations.

Appendices supplement the previously described text.

II. PRELIMINARY INVESTIGATION

This chapter presents the results of the preliminary investigation. It discusses the current information flow and data processing at the organization, the concerns over the information system issues that motivate the new system development, and the preliminary feasibility issues regarding the new system development.

A. OVERVIEW OF THE INTERNATIONAL PROGRAMS OFFICE

In order to gain insights of the IPO, its mission and the current organization are provided in the following paragraphs.

1. Mission

The NPS IPO is responsible for the cultural, social, and academic integration of the international community. The office is charged with interacting with the outside military and civilian agencies to accomplish the goals of the Security Assistance Program (SAP) and the Information Program (IP), described in the following paragraphs. Additionally, the office is responsible for the International Sponsor Program and acts as the Command Sponsor to International Committee.

a. Security Assistance Program (SAP)

The SAP is primarily involved with introducing American Technology to foreign countries (Callaghan et al. 1984) An important aspect of the program is the training and education of International Military Student (IMS) under Foreign Military Sales (FMS) and/or International Military Education and Training (IMET).

b. Information Program (IP)

The IP provides IMSs and their families with a basic understanding of the American way of life. The objective of the program is to provide them a balanced understanding of U.S. society, institutions, and goals in addition to the IMS's educational experience at NPS (SECNAVINST 4950.4).

c. International Sponsor Program

The International Sponsor Program is involved with assigning American sponsor(s) to each IMS to ensure academic, social, and cultural adjustments, well being, and success of IMSs and their families in the official and informal life at NPS.

d. Sponsor to the International Committee (IC)

The IPO acts as the official NPS sponsor between the International Committee and NPS. The IPO assists the IC in its actions and provides guidance, assistance, and approval for the conduct of activities of mutual interest and advantage to the IC and NPS.

The International Committee's purpose is to promote good relations between the U.S. and international officers and their families, to assist with adaptation to American life, and to support IMS and their families when needed and where possible (International Committee 1994).

2. Organization

The IPO is organized to manage the overall NPS's International Program.

a. Director

The Director of International Programs directs and coordinates academic and social activities and the physical adjustment of IMS and their families to the official and informal life at NPS. In addition, the Director provides policy guidance and implementation in planning and administering the SAP and IP.

b. Assistant Director

The Assistant Director functions as an assistant to the Director of International Programs. He/she manages the IPO staff and provides counseling to students and their families, prepares powers of attorney, and acts as an advisor to the IC.

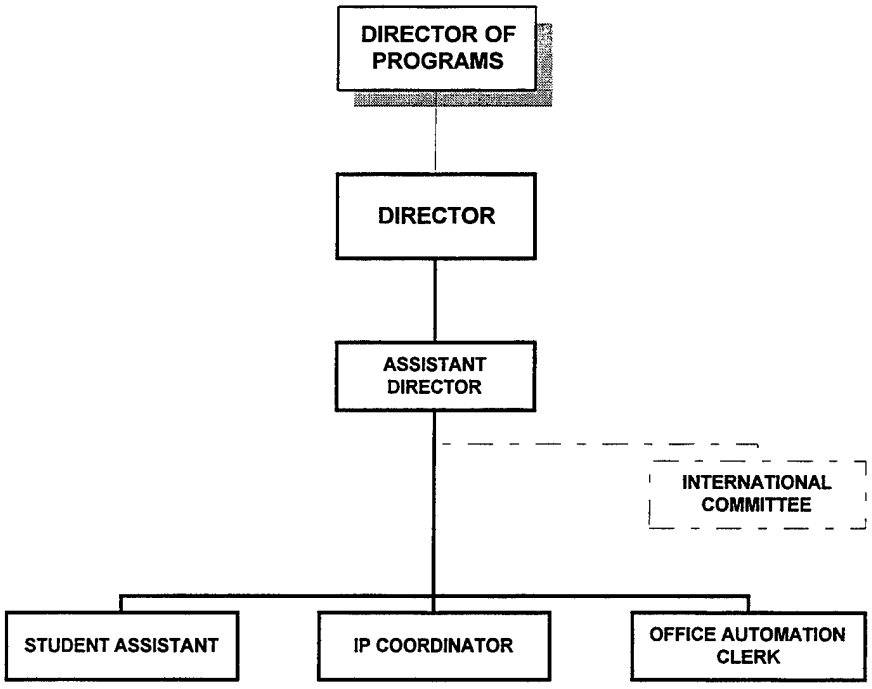


Figure 2. IPO Organization

c. International Student Assistant

The International Student Assistant is responsible for setting up and maintaining military personnel records for the IMS assigned to NPS. In addition, the assistant manages the International Sponsor Program and assists with the IP.

d. IP Coordinator

The IP Coordinator plans IP activities and maintains records of the IP to include financial records and accounting for IP budget, FMT travel, and FMT OPTAR accounts.

e. Office Automation Clerk

The Office Automation Clerk provides general clerical office support. He/she prepares student letters, provides basic information for students, purchases office supplies, prepares travel orders and arrangements, and maintains NPS OPTAR account.

3. Information Infrastructure

As mentioned earlier, the NPS IPO is charged with interacting with the outside military and civilian agencies to accomplish its goals. The nature of its mission requires timely, accurate, and intensive information exchange. Consequently, its information infrastructure is becoming a vital key to the organization's success in performing its mission-critical processes.

Two of the core functions performed by the IPO are maintaining information about IMSs in NPS as well as organizing IP activities. The following two sections discuss how these functions are performed and present the problems associated with these function areas.

a. Maintaining Information About IMSs

Currently the IPO relies on the use of a single database file to maintain information about the IMS population. This file consists of detailed information about IMS, IMS's dependent(s), and IMS's sponsor(s). The international student assistant sets up and maintains this database file based on the information provided by NETSAFA and IMS.

The existing database file and applications were created on an ad-hoc basis without the benefit of detailed analysis of the data structure and information processing requirements. The existing data and applications have been found to be inadequate for information processing requirements of the IPO. Problems with the existing database include incomplete data, inability to track changes in data, insufficient details in some data fields, and dysfunctional applications. For example, a "single" IMS could be legally single (i.e., not married) or he/she could be a geographical-single (i.e., his/her spouse is not together with him/her in the U.S.). This means that additional effort is required to verify details in some data fields to obtain supplementary data whenever such information is required.

Organization of the data stored into a flat file means that storage space is not optimized –e.g., the record for a single IMS would have unnecessary fields for information on the spouse and children.

Although the PC workstations in the IPO are networked, the "student" file consisting of IMS data stored in international student assistant's PC workstation is not accessible to the other staff members. Therefore, detailed IMS data is printed out on a regular basis and distributed to the other staff members by the international student assistant. Since most of the administrative activities performed by the IPO staff are based on the IMS information, the whole system turns out to be a manual, paper-based system.

b. Organizing the IP Activities

Currently the IP coordinator relies on a manual and paper-based system for organizing and reporting the IP activities. Paper records and forms are used for tracking activity applications, generating selection lists, and controlling attendance to the activities.

This manual system of record keeping means that a lot of effort is required to maintain the data. The retrieval of records associated with IP activities is also both time consuming and inefficient. As these records are being constantly referred to in the selection of IP participants, it is anticipated that an efficient retrieval system would give a considerable amount of boost to the efficiency of the selection process. As there are frequent changes to the participant selection list, a manual system has a greater potential for data inaccuracies. Even after the activity has taken place, there is a need to transfer the participants' records manually for the activity into the master record file for each of the participants.

c. Hardware and Networking

The IPO is equipped with five PC workstations, one for each staff member. Table 1 presents the current features of the workstations.

CPU	RAM (Mega byte)	HARD DISK (Mega byte)	OPERATING SYSTEM	USER
Pentium 75 Mhz	16	540	M.S. Windows 3.1	Assistant Director
Pentium 75 Mhz	16	540	M.S. Windows 3.1	International Student Assistant
Pentium 75 Mhz	16	540	M.S. Windows 3.1	Office Automation Clerk
486DX-2 66 Mhz	16	540	M.S. Windows 3.1	Director
486DX-2 66 Mhz	16	540	M.S. Windows 3.1	IP Coordinator

Table 1. Features of the IPO Workstations

The IPO PC workstations are networked, but none of them is configured as a file or database server. Therefore, the IPO staff members are not able to share resources possessed by each other's workstation. Staff members can access several servers through the networks maintained by the NPS Management Information Services with a data rate of 10 Mbps. Figure 3 depicts the current network structure of the IPO.

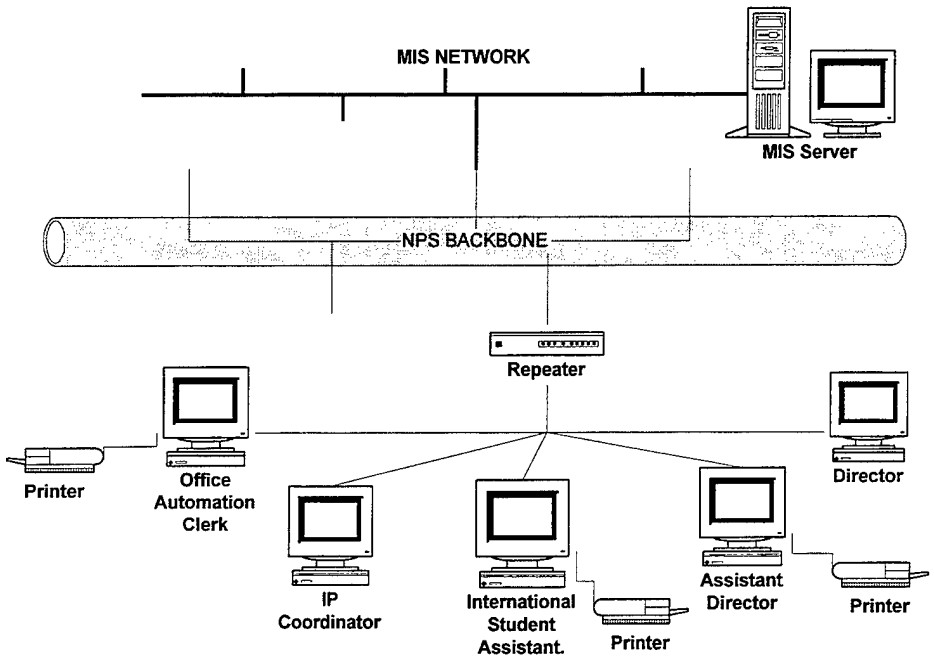


Figure 3. IPO Network Structure

B. PRELIMINARY FEASIBILITY ISSUES

The following paragraphs discuss the first-cut analysis of preliminary feasibility issues of the project.

1. Organizational Feasibility

The assistant director and members of the IPO staff are highly supportive of the project as observed during preliminary investigation. As the project does not alter the functional relationships and responsibilities in the IPO, there appears to be no cause for staff dissatisfaction or resistance.

The IPO staff have already been equipped with basic computer skills (i.e., being familiar with Windows operating system, word processing, etc.). Furthermore, it is anticipated that the development of the applications and their interfaces will involve the use of prototype designs. The use of prototypes would allow the staff to participate in system design and also acquaint themselves with the systems operation, reducing the need for extensive training at a later stage. Therefore, I envisage that there would be minimal training requirement for the staff.

2. Technical Feasibility

The IPO is equipped with networked PC workstations for each staff member (see Table 1.). The workstations run Microsoft Windows 3.1 and would be compatible to common DBMS product likely to be used in the project. Given that the workstations are already networked, it would be relatively easy to leverage on the existing network resources for shared database and applications.

3. Schedule Feasibility

Based on the findings of the preliminary investigation, with detailed system analysis begun December 1995, implementation of a fully operational system is scheduled for completion by June 1996. This leaves three months for maintenance and fine tuning of the system before the completion of the thesis.

4. Economic Feasibility

At the outset, the IPO made it clear that no funds were available to support the new system development or reengineering the existing system for a certain period. Therefore, acquisition of additional software or hardware should not be considered during the system development in order to complete the project on schedule.

a. Costs

The development and implementation cost of this system is anticipated to be limited to my personal time and effort and IPO staff members' time. Since my personal time in NPS and effort for the thesis process do not have a monetary cost, they would be assumed to be free. A total 10 working hours of the IPO staff members were spent during preliminary investigation (i.e., interviews with staff), and approximately 30 staff working hours are expected to be spent during the prototype previews. Both these add up to 40 staff working hours for the development and implementation cost.

The operation cost of the system could be negligible if the system would be designed as robust and flexible. There would not be any requirements for maintenance.

Since there is no formal charging policy for computer usage in NPS, marginal cost of computer resources consumed throughout the life cycle would be assumed to be free. Consequently, the total cost of the system is anticipated to be limited to 40 staff working hours throughout the life cycle of the system.

b. Benefits

The anticipated benefits which would be provided by the new system are:

- All required data associated with the IPO would be stored in easily maintained, robust and flexible database which can be accessed by all staff members through the network. This would allow the IPO staff to analyze the IP data which is not possible with the current manual system,
- Reduce time required to perform administrative tasks (submitting standardized reports, selecting IP participants, etc.),

- Reduce errors or inconsistencies which occur during the manual transcription processes.
- Improve quality of managing the Information and Sponsor Programs,
- Reducing time and improving services will save staff time and make it possible to reduce the size of the staff.

III. REQUIREMENTS ANALYSIS

This chapter describes the requirements analysis phase of the system development. The analysis provides conceptual data and process models associated with the IPO's administrative activities.

At the outset of the system development effort, specific data and application requirements which should support the new system were defined by employing the structured development methodology.

During the early stages of this phase, the data and process models were constructed using a hybrid approach in such a way to allow for extension. Analysis of the standardized reports and sample forms allowed a "bottom-up" approach which was essential in adding detail to the basic attributes of the semantic objects used to model data requirements. Interviews, held during preliminary investigation phase, were very helpful "top-down" information sources. In the following phases of the system development the visual development methodology was employed, allowing dynamic user requirements to be incorporated to refine the results of preliminary requirements analysis.

A. INFORMATION REQUIREMENTS

During the preliminary investigation various fact-finding techniques were applied, including review of all documentation (procedures, standardized reports, forms, etc.) related to the administrative activities held by the IPO. Interviews were conducted with the IPO staff, the IPO MIS network consultant, international students, and student sponsors. Based on the results of these fact-finding efforts, the IPO's data and application requirements were defined.

1. Data Requirements

Assistant director Cynthia Graham made it clear that the student database maintained by various NPS offices is not detailed and adequate enough to satisfy the IPO's information requirements. She insisted on having a stovepipe database in which the IPO can have full control and rights to determine its structure. Based on her 14 years

experience in the IPO, she explained the IPO's reasons to refuse to make use of a NPS student database.

> The IPO is the first place that has the initial information about an IMS and for an IMS our office is the first check-in place. Thus we have the first information (about an IMS). Although there are several offices or agencies just to update student database, we have the most recent data about them. Our office is the first place where an IMS wants to report change of status, because periodically we remind IMSs to update their data, and they (IMSs) usually forget to get their data updated by the other offices. The other offices know this and they always ask us information about students (IMSs). We need detailed information other than provided by the NPS database. We absolutely need accurate data on time. Finally, availability of the data is important.

In order to achieve its goals, the IPO should capture detailed data about IMSs, sponsors, dependents, IP activities, participation in IP activities, vendors providing services to the IP activities, and former students (alumni).

Data captured by the system should be modeled in a flexible structure to allow further enhancement or future maintenance without losing any information in the database. Duplication of data in the system must be kept minimal.

2. Application Requirements

Since the data captured by the IPO has personal information about IMSs, sponsors and dependents, it must be protected in accordance with the Privacy Act 74. Therefore, multilevel security must be provided.

Applications must be user friendly so that any user with basic Microsoft Windows skills can use them without extensive training.

Applications must be fully functional even though they have to run standalone.

B. CONCEPTUAL DATA MODEL

The conceptual data model has been designed using SALSA™, a semantic object data modeling and database design tool. Total of eight semantic objects (SO) were created to represent the data model: IMS, IMS DEPENDENT, SPONSOR, SPONSOR DEPENDENT, COUNTRY, CURRICULUM, IP ACTIVITY, and VENDOR objects.

The comprehensive semantic object model of the data that is maintained by the system's database is depicted in Appendix A as a Semantic Object Diagram. The semantic object diagram summarizes the knowledge of the objects and presents it to the users in an unambiguous fashion (Kroenke 1995). The attributes of semantic objects and the domain definitions of these attributes are described in Appendix A as Semantic Object and Attribute Reports.

1. IMS (International Military Student)

Since most of the IPO's administrative activities rely on the information associated with the international students from various foreign military and government civilian agencies, the core object in the model is the IMS SO. Data captured in this SO may represent either an incoming IMS, an onboard IMS, or an alumnus. When an IMS's attendance at NPS is approved by NETSAFA, the IPO will be informed by NETSAFA with the "acceptance message" about incoming student. Typically, the information about the incoming student is known one to two months prior to his/her arrival at NPS. These limited preliminary data are used by the IPO to alert curricular offices of the expected IMS enrollment as well as to determine suitable sponsor assignments among available candidates or to scout for prospective sponsors from the U.S. student or civilian population. The data captured at this point in time will be matched to the official travel orders when an IMS reports to NPS.

IMS SO has a large number of simple attributes (each of which represents a single piece of data, such as Rank, Service, Tuition Code, etc.), and group attributes (which represent several pieces of data, such as Name, which groups First, Last, and Middle names). These attributes are described in Appendix A as the Attribute Report. The arrival date attribute distinguishes incoming IMS. The graduation attribute distinguishes alumni.

There are several simple attributes that could function as the key to this SO. However, the preference of the IPO is to use the "Student Control Number" (SCN). This unique identifier is assigned by NETSAFA at the first training activity attended.

Being the core object in the model, the IMS SO has relations with the other semantic objects in the model. These relations will be elaborated as the other SOs are discussed.

2. IMS Dependent

This SO captures data about an IMS's official dependents, including the relationship and date of birth. IMS dependent(s) may or may not be in the U.S. with IMS. The simple attribute named "DepStatus" describes this situation. IMS DEPENDENT SO also captures the data about dependent's attendance to the IP classes.

A simple relation for this SO is that with the IMS. There is a zero-to-many (0:N) relation between these SOs –i.e., an IMS may have more than one dependent, and in the case of a bachelor IMS, there would be no dependent.

3. Sponsor

In accordance with the International Sponsor Program, the IPO tries to find locals or U.S. students who are willing to assist in the orientation of incoming or onboard IMS. The IPO tries to provide the best match between a sponsor and the IMS based on the marital status, curriculum, sponsor's preferences, etc. Therefore, SPONSOR SO contains a number of attributes to achieve this goal.

Sponsors are registered when they apply to this program. Therefore, a sponsor may be registered but he/she may not be assigned immediately to an IMS. On the other hand, a registered sponsor may be assigned to more than one IMS. Similarly, an IMS may have more than one sponsor. However, the sponsors, particularly the U.S. students due to the graduate, may leave the program, thus the IMS may not have any sponsors at the later stages of his/her stay, which results in a 0:N relation in both IMS and SPONSOR SOs. In general, the relation between both SOs can be characterized by a many-to-many relation, though in practice a one-to-one relation is more common.

4. Sponsor Dependent

This SO is similar to the IMS DEPENDENT SO mentioned earlier. The only difference is that SPONSOR DEPENDENT does not capture IP Classes data and additionally captures SPONSOR DEPENDENT's hobbies and interests.

A simple relation for this SO is that with the SPONSOR. There is a 0:N relation between these SOs –i.e., a SPONSOR may have more than one dependent, and in the case of a bachelor SPONSOR there would be no dependent.

5. Country

The COUNTRY SO allows the IPO to capture data associated with the countries where the IMSs are from.

In accordance with the Security Assistance Management Manual, countries in the system are represented with a two character alphabetical code named "Country Code" which uniquely identifies a country. For reporting and viewing purposes several attributes are added to the SO –e.g., Country Name which captures the name of the country such as Turkey. Since each country represented by IMSs in NPS has a senior officer, this data is also inserted into the COUNTRY SO.

The IMS SO has a one-to-many relation with this SO –i.e., every IMS must be from only one country, a country may be represented by more than one IMS, and a country may exist in the database without any IMS representative.

6. Curriculum

The CURRICULUM SO allows the IPO to capture data associated with the curricula in NPS.

Curricula in NPS are represented with a three digit number code, named "CurrNumber" which uniquely identifies a curriculum.

Since an incoming IMS's curriculum information may not be known beforehand and a sponsor may be civilian not attending NPS, the IMS and SPONSOR SOs have zero-to-many relation with this SO.

7. IP Activity

The term "IP activity" refers to all types of Information Program events organized by the IPO (field trips, seminar classes, parties, etc.). IP ACTIVITY SO captures the data associated with each of these events, such as participation, services used for these events, etc.

The relations with this SO are built in its group attributes (e.g., services used). VENDOR, IMS, and SPONSOR SOs have many-to-many relations with this SO.

8. Vendor

This SO allows the IPO to capture data about the businesses that the IPO needs to deal with for each IP activity.

The point of contact data would be useful for making arrangements for repeated activities. In addition, the IPO may consider adding vendor evaluation information in order to identify poor service providers that should be avoided in future dealings. A similar argument can also be made for those vendors that provide good service and pricing. Therefore, a "comment" attribute is added.

The VENDOR SO has many-to-many relation with IP ACTIVITY, because a vendor may provide service for more than one IP activity and an IP activity may use services of more than one vendor.

C. PROCESS MODEL

The data flow diagrams were created following the explosion approach. Figure 4 shows the decomposition diagram which gives the overview of the process model and the processes which make up the system.

The data flows and processes within the system were depicted in the data flow diagrams presented in Appendix B. A context diagram, representing the first level of data flow, was included as the first page of Appendix B. The context diagram was next decomposed by exploding the primary process into more detailed subprocesses. This "exploding" further enhanced the examination of process details. Process explosion continued until all requirements were fully visible.

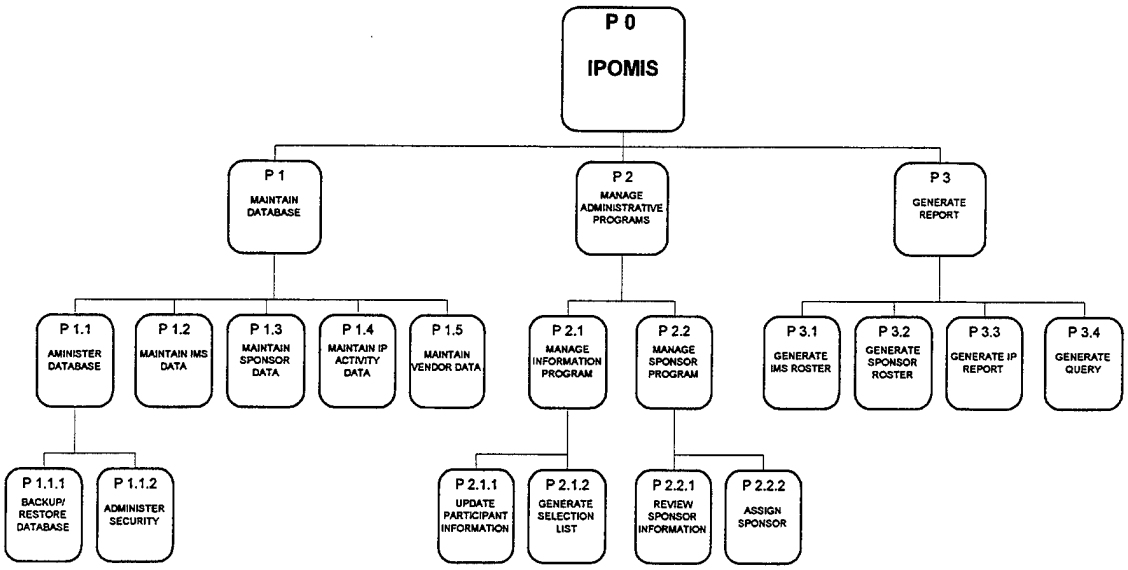


Figure 4. IPOMIS Decomposition Diagram

1. Level 0

The root process is defined as the International Programs Office Management Information System (IPOMIS). The process interacts with NETSAFA, Foreign Country Liaison Office, Sponsor, IMS, Alumnus, Vendor, Curriculum Office, and NPS Administration external entities. The interactions are depicted with the context diagram in Appendix B. Most of the external entities interact with IPOMIS directly, but NETSAFA, Foreign Country Liaison Office, Vendor, and NPS Administration indirectly use and experience the system via its outputs.

2. Level 1

The essential root process is exploded into three subprocesses: Maintain Database, Manage Administrative Programs, and Generate Report, including IPOMIS data model.

The maintain database (P 1) focuses on administering the database and maintaining basic data captured by the IPOMIS data model. The raw data maintained

here is used for managing administrative programs and report generation. This data is entered and updated regularly and covers all attributes associated with Vendor, IP Activity, IMS, Sponsor, and their dependents.

The Manage Administrative Programs subsystem (P 2) focuses on managing IP activities and managing sponsor program.

Generate Report (P 3) focuses on generating the standardized reports for which the IPO is responsible.

3. Level 2

The Maintain Database subsystem is decomposed into five subprocesses. First of them deals with administering the IPOMIS database. The other four processes focuses on changing and maintaining IMS, sponsor, IP activity, and vendor data.

The Manage Administrative Programs subsystem is decomposed into two subprocesses: Manage Information Program (P 2.1), and Manage Sponsor Program (P 2.2).

The Generate Report subsystem is decomposed into four subprocesses. Generate IMS Roster (P 3.1), Generate Sponsor Roster (P 3.2), and Generate IP Report (P 3.3) subprocesses individually focus on retrieving the relevant information from the database to prepare a specific report and viewing the report before printing. Generate Query (P 3.4) subprocess models how the users generate ad hoc reports based on the query results.

4. Level 3

The Administer Database subprocess (P 1.1) explodes to two subprocesses. Backup/Restore Database (P 1.1.1) models basic database administration tasks: generating backups of the database and restoring the backups. Administer Security (P 1.1.2) models how the system database administrator adds and deletes user names and modifies a user's parameters, such as password and user ID.

The Manage Information Program subprocess (P 2.1) explodes to two subprocesses. Update Participant Information (P 2.1.1) captures the participation data from IMS and Sponsor's IP activity applications –i.e., sign-ups, keeps track of services

provided, and stores relevant data to the IP Participation data storage in the appropriate format. Generate Selection List (P 2.1.2) generates a selection list for a particular planned IP activity based on the IP Participation Information. It retrieves data from IMS, Sponsor, IP Activity, and IP Participation data storage, then it processes this data to create the selection list in accordance with the selection criteria. It allows the IP coordinator to review and edit the generated selection list.

The Manage Sponsor Program subprocess (P 2.2) explodes to two subprocesses. Review Sponsor Information (P 2.2.1) models how the users access and review sponsor information to provide a better match between sponsor and IMS. Assign Sponsor (P 2.2.2) models how the users assign a sponsor to an IMS.

IV. SYSTEM DESIGN

This chapter discusses the details of the system design phase of the system development. Two components of the system design, logical database design and application design, are discussed. The chapter addresses issues, associated with designing an appropriate system and selecting appropriate application development tools.

A. LOGICAL DATABASE DESIGN

In designing the database, the semantic object model developed in the previous chapter was transformed into a relational schema – a "blueprint" of the database structure. The eight semantic objects were transformed into 12 relations. Figure 5, the relational schema, shows the relations that resulted. Each of the 12 relations are reflections of the original semantic objects with appropriate foreign keys included. In Figure 5, primary keys are denoted by underlining, foreign keys are denoted by italicizing, and foreign keys used as primary keys are denoted by both underlining and italicizing. Detailed descriptions of each relations are represented in Appendix C.

B. APPLICATION DESIGN

In accordance with the VDM, the developer cooperated closely with users to determine what the basic version of the application should do. They examined the business processes being modeled and determined the basic, core functionality that the new system is to provide. Then focus was placed on building a system that provides this core functionality:

- **Maintaining Database** focuses on updating and maintaining the IPOMIS database, particularly data associated with the core object (i.e., IMS Object).
- **Managing Sponsor Program** focuses on selecting and assigning the appropriate sponsor to IMS and includes updating the relevant portions of the database (e.g., relation between sponsor and IMS).

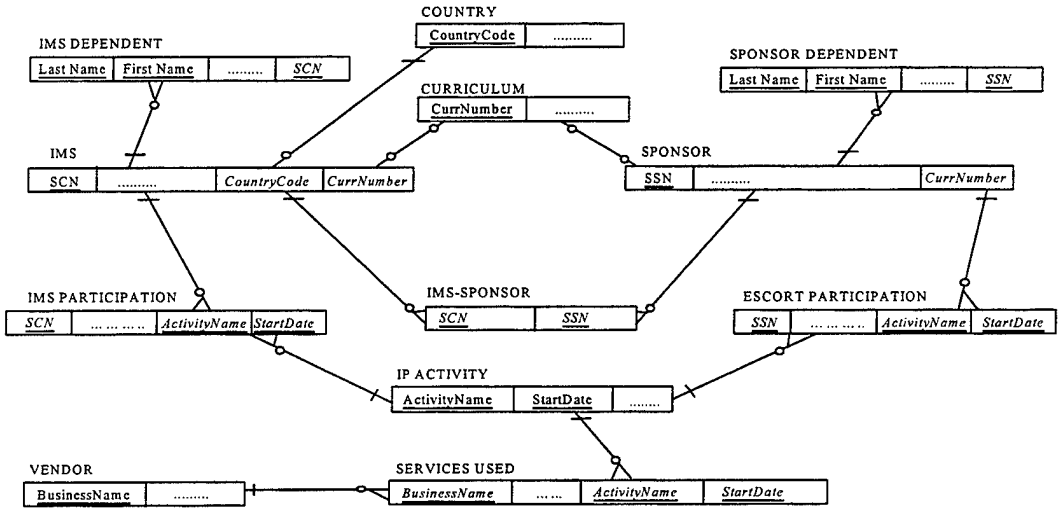


Figure 5. Relational Schema

- **Managing Information Program** focuses on arranging the IP activities, determining the participation to these activities, and updating relevant portions of the database (e.g., data associated with the services consumed during an activity).

Once a good grasp of these core functionality was gained, the developer began to develop the initial interfaces and the logic for the basic functions of the applications. Delivering a product that provides the basic functionality for each application was the central focus. As described in the next chapter, these early versions of the applications gave the developer and users a common reference point for further development.

During this phase, and indeed all throughout the process, users could have a strong input into how the application would look and operate, as this was anticipated to be a big factor in the application's acceptance and the productivity level of users as they employ the system.

Getting the interface right at the outset saved coding changes later, though visual tools would make any interface changes very easy. The developer, at this point built a

skeleton for the actual system, presenting user with input and output screens, report views, and other interface elements. Visual tools allowed the developer to make changes, often right in the presence of users, to ensure that the interface design was satisfactory and met the needs and desires of those to be using the system.

C. APPLICATION DEVELOPMENT TOOL –BORLAND'S DELPHI

The visual development methodology approach uses a visual object-oriented application development tool, in this case Borland's Delphi™, to rapidly develop an application prototype. Borland's Delphi is a development tool specifically designed to create client-server applications.

This type of tool allows the developers to build highly sophisticated applications with very little code writing. These products provide the developer with a library of pre-packaged modules that can be visually combined into complete applications. The real power of these tools stems from this library of components that allow an application to be assembled with connections to databases, video, imaging, and messaging (Borland 1995).

The purpose in using a visual tool is to provide the user with a prototype as quickly as possible. The user can then play a significant role in the applications development life cycle by continually providing the developer with feedback. This iterative process is commonly referred to as RAD, and promises to greatly improve the software development process that for years has suffered from rigid methodologies (Hodges 1995).

V. SYSTEM IMPLEMENTATION AND MAINTENANCE

This chapter discusses the details of the system implementation and maintenance phases of system development.

The primary objective of the implementation effort was to build fully functional applications that would satisfy the end users. During the implementation, several prototypes were presented to the end users and expanded to include all functionality fully integrated into the applications, as well as the database.

Appendix D, the User's Manual, provides documentation which details the final database and applications' features and operations.

A. DATABASE IMPLEMENTATION

Two major tasks were performed for implementing the IPOMIS relational database:

1. Selecting Database Management System (DBMS)

A Database Management System (DBMS) is generally defined as a collection of computer programs used to create, maintain, access, update, and protect one or more databases (Kroenke 1995).

In order to meet data requirements presented in Chapter III, Borland's InterBase™ Server, an SQL-compliant relational DBMS, was selected among many other relational DBMS products, such as Borland's Paradox™ and Microsoft's Access™. IPOMIS applications can access an InterBase database through the Borland Database Engine (BDE) and the InterBase SQL Link (Borland SQL Links for Windows®).

Figure 6 shows the relationships between the InterBase Server and the associated connections for data access.

InterBase offers all the benefits of a fully relational DBMS. The following list provides some of the key InterBase features which affected the DBMS selection process:

a. SQL Support

InterBase conforms to entry-level SQL-92 requirements. It supports declarative referential integrity, updatable views, and outer joins. InterBase also supports extended SQL features, some of which anticipate SQL3 extensions to the SQL standard.

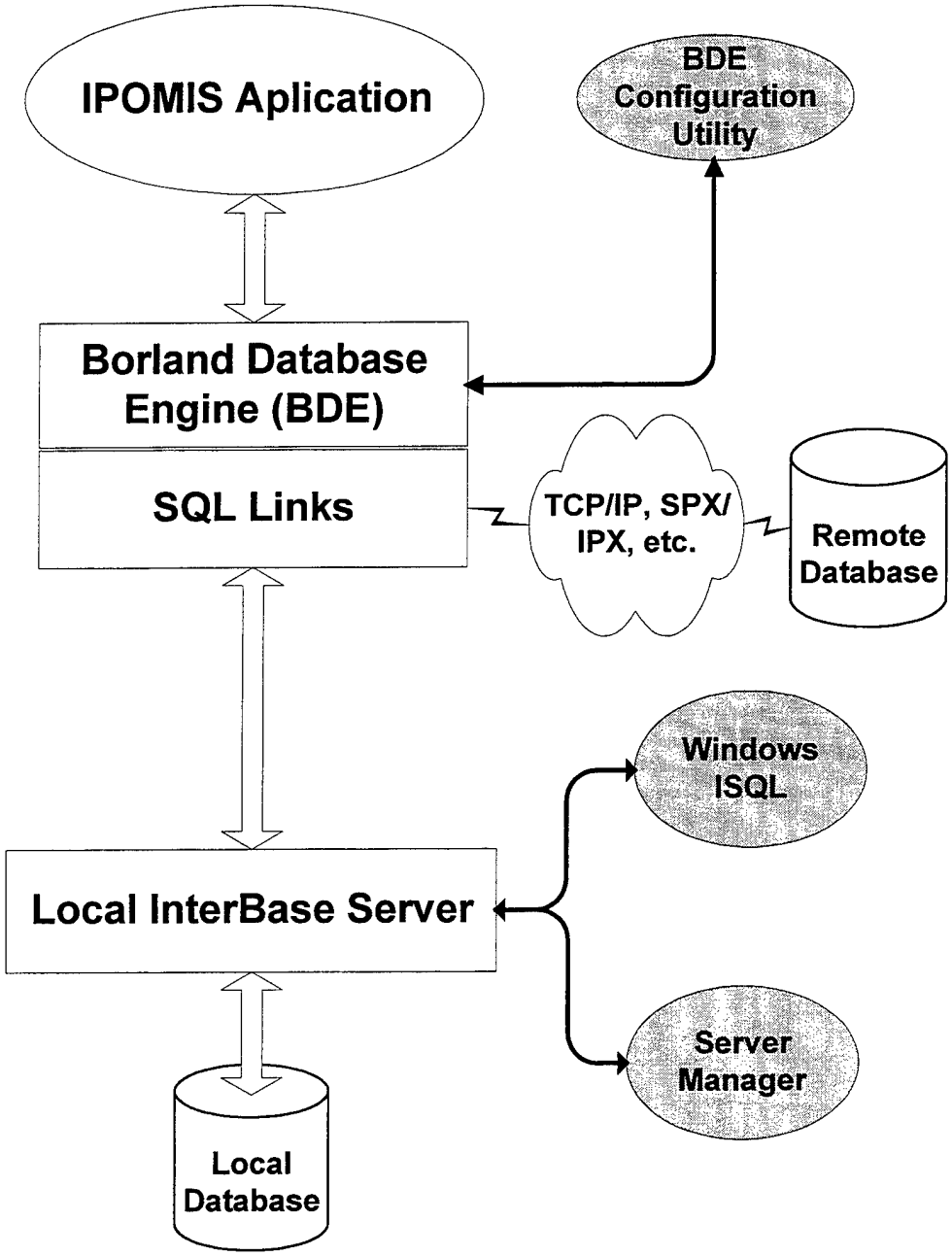


Figure 6. InterBase Connections

These include stored procedures, triggers, and segmented BLOB support.

b. Transaction Management

Client applications can start multiple simultaneous transactions. InterBase provides full and explicit transaction control for starting, committing, and rolling back transactions. The statements and functions that control starting a transaction also control transaction behavior.

InterBase transactions can be isolated from changes made by other concurrent transactions. For the life of these transactions, the database will appear to be unchanged except for the changes made by the transaction. Records deleted by another transaction will exist, newly stored records will not appear to exist, and updated records will remain in the original state.

c. Multiple Database Access

InterBase provides simultaneous access to multiple databases, –i.e., one application can access many databases at the same time and concurrent multiple applications can access to the database at the same time.

d. Query optimization

The server optimizes queries automatically, or user may manually specify query plan.

e. Multigenerational architecture

InterBase provides expedient handling of time-critical transactions through support of data concurrency and consistency in mixed use (query and update) environments. InterBase uses a multigenerational architecture, which creates and stores multiple versions of each data record. By creating a new version of a record, InterBase allows all users to read a version of any record at any time, even if another user is changing that record. InterBase also uses transactions to isolate groups of database changes from other changes.

f. Royalty-free Redistribution

The Local InterBase Server includes Windows ISQL, an interactive data definition and query tool for Windows; and the Server Manager, a Windows tool for database backup, restoration, maintenance, and security. The entire software package can be redistributed by the developer without any additional cost to the system development.

2. Defining Database Structure

Based on the relations described in the design of the logical database in previous chapter, the structure of the database was defined using the data definition language (DDL) to the DBMS. Twelve tables were defined:

- COUNTRY,
- CURRICULUM,
- IMS,
- IMS_DEPENDENT,
- SPONSOR,
- SPONSOR_DEPENDENT,
- IMS_SPONSOR,
- IP_ACTIVITY,
- VENDOR,
- IP_ACTIVITY_SERVICES_USED,
- IP_ACTIVITY_ESCORT_PART,
- IP_ACTIVITY_IMS_PART.

In addition to these tables, an ALUMNUS table was defined. As described in Chapter III, an alumnus is a former IMS who already graduated from NPS. The data about an alumnus was captured and modeled in the IMS SO. Therefore IMS relation was supposed to include alumnus data. After taking the following factors into consideration, creating an ALUMNUS table was decided in order to contain alumni records separately:

1. The records contained in the IMS table has a number of fields which are not required to be kept for an alumnus. Creating a new record in the IMS table for an alumnus

would create overhead in the database. Also the reduced data set would allow effective maintenance of key information about the alumni.

2. Onboard IMS data is very dynamic and subject to high query traffic; in contrast alumnus data is static due to its archive nature.
3. Since there are currently more than 800 records associated with alumni in the database and this number will be growing by approximately 30 at the end of each quarter, soon the records contained in the IMS table will reach more than 1000. This increasing number of records will affect the query performance. For each query on the IMS table about onboard IMSs will face with an overhead caused by alumni records.

DDL scripts used to implement IPOMIS database are represented in Appendix E. On completion of the database creation, data stored in the previous database file was mapped into IPOMIS database tables. Some of the data could not be transferred, and the previous database was not designed to capture all the data required by the IPO. Therefore, the database administrator had to review the new database and student files in order to complete and update records. This was the most time consuming process.

B. IPOMIS DATABASE ADMINISTRATION

The IPOMIS Database Administrator (DBA) performs the database administration by using InterBase Server Manager which enables the DBA to:

- Manage server security.
- Back up and restore a database.
- Perform database maintenance.
- View database and lock manager statistics.

1. Managing Server Security

InterBase maintains a list of user names and passwords in a secure database. The security database allows clients to connect to an InterBase database on a server if a user name and password supplied by the client match a valid user name and password

combination in the security database on the server. The DBA can add and delete user names and modify a user's parameters, such as password and user ID.

In order to provide data security and integrity, seven users, including a "guest" user, have been defined to the IPOMIS database. Table 2 presents the detailed permissions granted to the users. The DDL scripts used to grant these permissions were presented in Appendix E.

USER	USER ID	READ ONLY ACCESS	READ-WRITE ACCESS
DIRECTOR	DIRECTOR	ALL TABLES	-
ASSISTANT DIRECTOR	ASSTDIR	ALL TABLES	-
INTERNATIONAL STUDENT ASSISTANT	IMSASST	IP_ACTIVITY IP_ACTIVITY_SERVICES_USED VENDOR	ALUMNUS COUNTRY CURRICULUM IMS IMS_DEPENDENT IMS_SPONSOR IP_ACTIVITY_IMS_PART IP_ACTIVITY_ESCORT_PART SPONSOR SPONSOR_DEPENDENT
IP COORDINATOR	IPCOORD	ALUMNUS COUNTRY CURRICULUM IMS_SPONSOR SPONSOR_DEPENDENT	IMS IMS_DEPENDENT IP_ACTIVITY IP_ACTIVITY_IMS_PART IP_ACTIVITY_ESCORT_PART IP_ACTIVITY_SERVICES_USED SPONSOR VENDOR
OFFICE AUTOMATION CLERK	OACLERK	ALL TABLES	-
SYSTEM DATABASE ADMINISTRATOR	DBA	-	ALL TABLES
GUEST	GUEST	ALL TABLES	-

Table 2. IPOMIS Database Users and Permissions Granted

2. Performing Database Backup and Recovery

The Server Manager can back up a database and then restore it on any supported operating system. A backup can run concurrently with other processes because it does not require exclusive access to the database.

3. Maintaining the Database

The Server Manager can also be used for maintaining a database and preparing it for shutdown. If the IPOMIS database incurs minor problems, such as an operating system write error, these tools enable the DBA to sweep a database without taking the database off-line.

4. Viewing Statistics

The Server Manager enables the DBA to monitor the status of a database by viewing statistics from the database header page and an analysis of tables and indexes.

C. APPLICATION IMPLEMENTATION

During the design phase, the developer delivered earlier versions of the applications that contain the basic functionalities for each application. These basic functionalities provided users with some capability that could prove useful to their jobs. At this point, users could evaluate the applications, uncover some defects, and even use applications to support some of their tasks. They provided feedback as to the success or failure of the design decisions made during the requirement analysis.

Once the initial versions of the applications were deployed, the developer began the development process again. After evaluating each version of applications, users provided valuable feedback in the form of bug reports, ideas for enhancement, and ideas for new functionality. Once users were actually using the application, they had a much better idea what the applications could and should do. The developer gathered and prioritized the feedback, mixing his own expertise with the requests and knowledge of the users. The developer then determined how the applications would be improved and implemented those features were determined to be of the highest priority.

1. Applications Implemented

Based on the process model described in Chapter III, three windows-based applications were implemented. Processes were grouped to contain functionality which can be performed by the standalone applications. In addition to the applications

implemented, two utility programs – Interbase Server Manager and Windows ISQL were provided to administer the database and generate queries.

a. IMS Information

The IMS Information application helps the user, –specifically International Student Assistant– to perform most of his/her tasks associated with the IPO's administrative activities. It provides a user-friendly interface to access the IPOMIS database where the user can maintain IMS, IMS_DEPENDENT, COUNTRY, CURRICULUM, ALUMNUS tables (see Figure 7) and to generate various structured IMS rosters and reports. Therefore, IMS Information contains the functionality represented by P 1.2 and P 3.1 processes described in Chapter III.

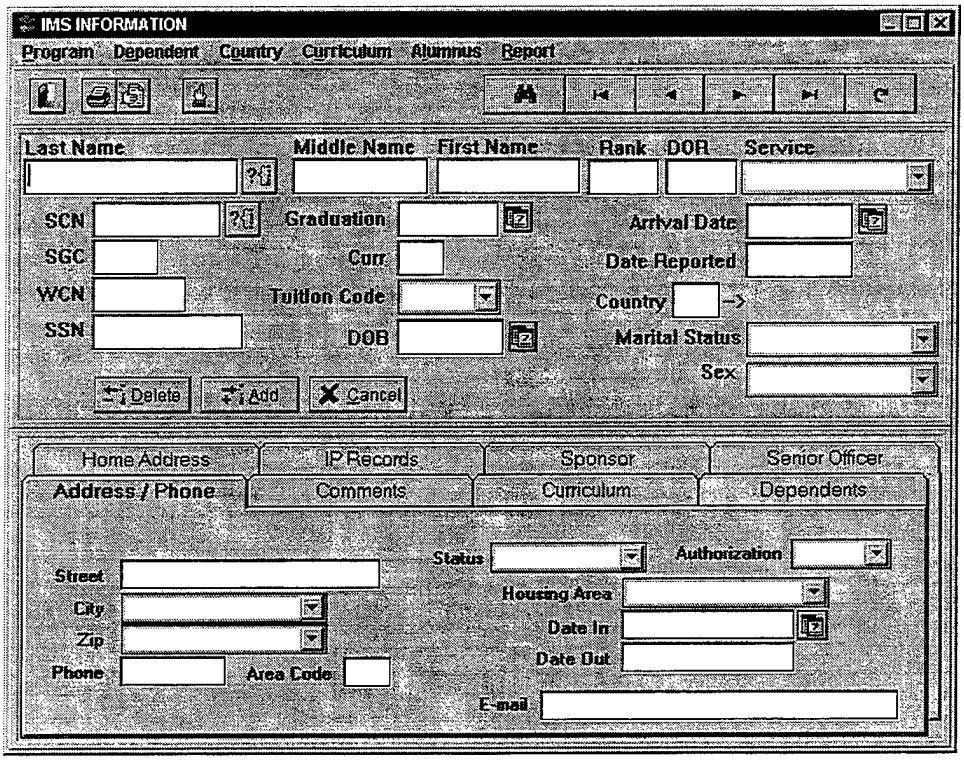


Figure 7. IMS Information Interface

A read-only version of this application was also deployed to allow the users other than International Student Assistant to review information related to IMS. The read-only version does not support reporting features of the read-write version.

b. Sponsor Program Manager

The International Student Assistant uses this application to perform tasks associated with the sponsor program. It allows the user to select and assign the appropriate sponsor to IMS and to maintain SPONSOR and SPONSOR_DEPENDENT tables and to generate various structured sponsor rosters and reports (see Figure 8).

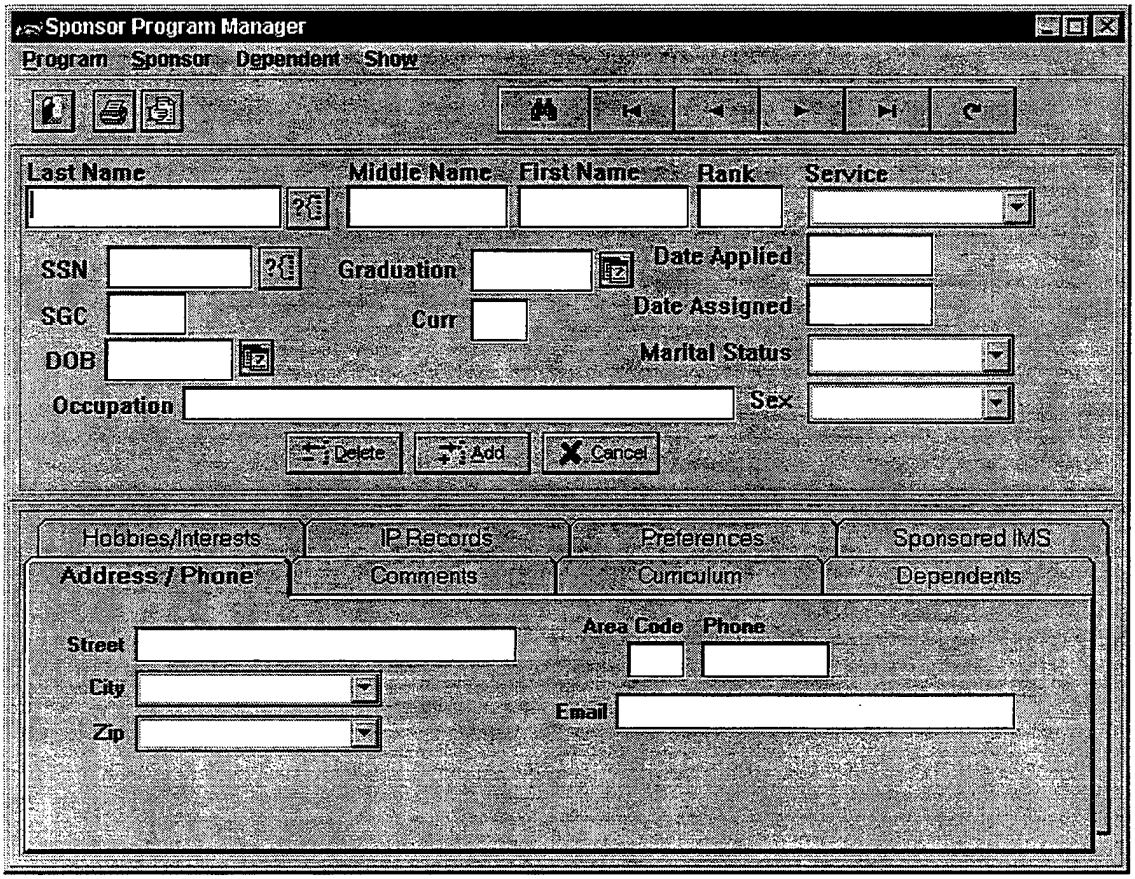


Figure 8. Sponsor Program Manager Interface

Therefore, Sponsor Program Manager contains the functionality represented by P 1.3, P 2.2.1, P 2.2.2, and P 3.2 processes described in Chapter III.

The read-only version of this application allows the other users to review information related to sponsors. The read-only version does not support reporting features of the read-write version.

c. Information Program Manager

The Information Program Manager helps Information Program Coordinator to perform his/her tasks associated with managing and coordinating the IP activities, such as preparing a selection list, reviewing and approving applications, etc. Therefore, Sponsor Program Manager contains the functionality represented by P 1.4, P 2.1.1, P 2.1.2, and P 3.3 processes described in Chapter III.

This application provides a user interface that incorporates various data representation features in order to provide the user detail information about an IP activity (see Figure 9).

2. Application Assessment

The quality of the applications developed is a testimony to the power of such visual programming tools as Delphi™. Designing of applications of equal functionalities without the use of a visual programming tool would be a tremendous task. These applications have been designed within six months.

3. Acceptance

All throughout the development process, end users could have strong inputs into how the applications would look. This became a big factor in the applications' acceptance and the productivity level of users as they employed the system.

4. Source Codes

For the use of future enhancements and maintenance, source codes of the applications developed are documented and presented to the IPO. A sample of source code can be found in Appendix F.

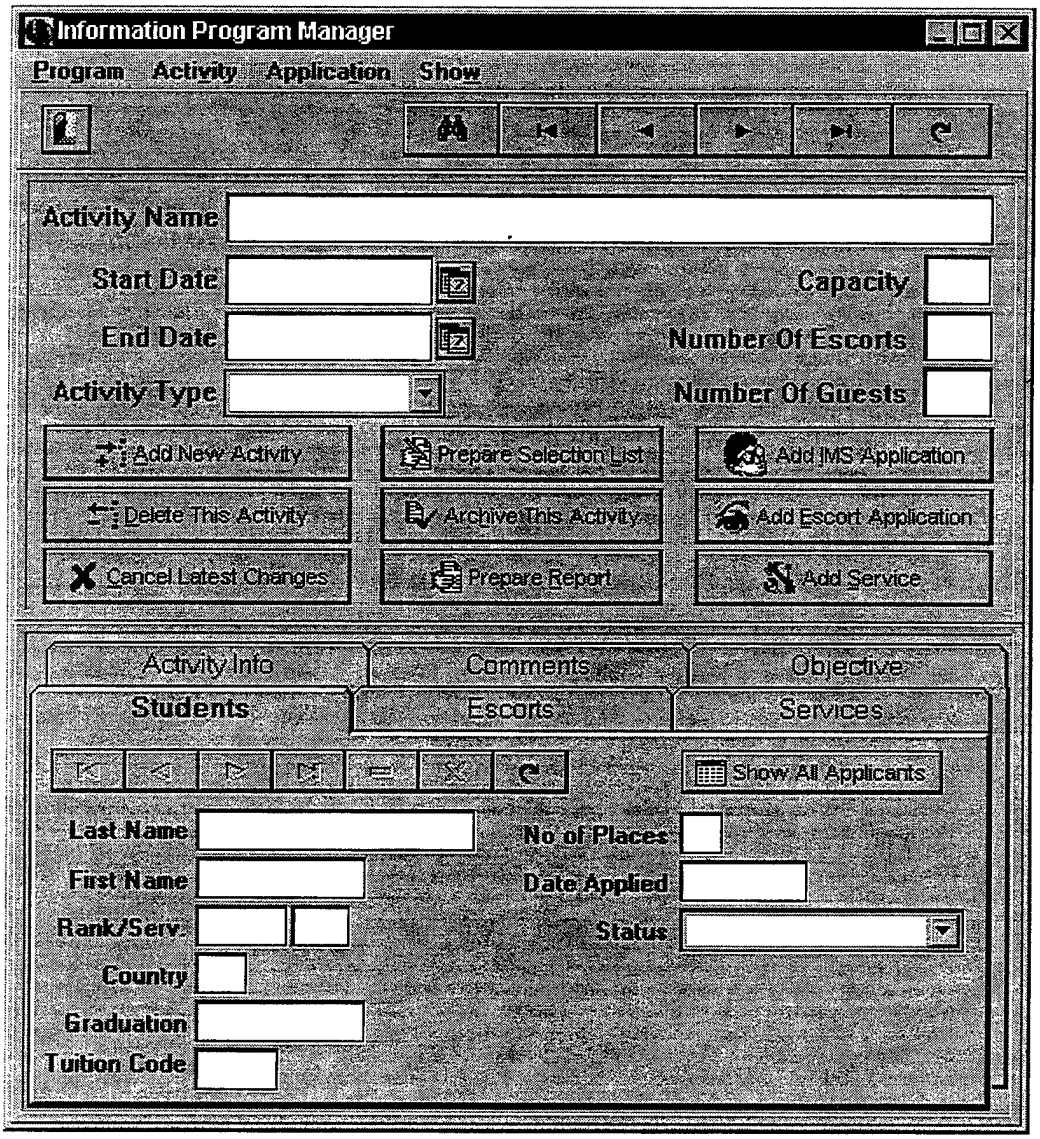


Figure 9. Information Program Manager Interface

D. MAINTENANCE

In looking at the VDM, it becomes clear that the concept of system development phases becomes very blurred. The difference between requirements analysis and design, or the difference between implementation and maintenance, becomes harder to see because the building of the applications pervades virtually every area of the entire development cycle.

As displayed in previous chapters, development begins in the very early stages of investigating the problem to be solved. Solving the problem identified is not the completion of the development process. Instead, this process will continue indefinitely. As long as there is a need for the application, developers should continue working on it. What was maintenance becomes implementation (Hodges 1995).

The VDM means a redefining of the meaning of the maintenance phase. The VDM does not draw a line between maintenance and development. Maintenance becomes a part of the original development process itself and begins very early in the development effort. Therefore, the project presented in this thesis should be viewed as being in the maintenance mode right from the beginning, rather than at the end of the development life cycle.

Throughout the development process, Delphi allowed the developer to make changes, often right in the presence of the users, to ensure that the interface design was satisfactory and met the needs and desires of those to be using the system.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The end results of this thesis are well defined information requirements and three scaleable, robust client/server applications that have been useful since their initial deployment.

A mixture of both Structured Development Methodology (SDM) and Visual Development Methodology (VDM) were followed for developing the information system required by the IPO. In order to analyze the information requirements of the IPO thoroughly, preliminary investigation and requirements analysis were conducted by studying the discipline of the SDM. Because the SDM requires to freeze the specifications after the design specifications are complete, it fails to produce applications that meet users current needs. Therefore, in order to achieve the other objectives of this thesis, the VDM was employed to address the shortcomings of the structured approach.

This system development effort illustrates the potential of the visual development methodology. An inexperienced developer was able, in a manner of six months, to build three robust applications that could have been immediately deployed and used by the International Programs Office, yet which could be easily extended and enhanced to meet the future information requirements of the IPO. In this regard, IPOMIS is one of the best validations of the efficacy of the visual development methodology.

Semantic Object Modeling and Data Flow Diagrams were utilized to organize and document the system's data and functionality. As the implementation-independent models, Semantic Object model and Data Flow Diagrams helped the developer to create a graphical representation of reality without the biases that might be the result of the way the previous system was implemented or the way that any person thinks the system might be developed and implemented.

The visual tool Delphi™, an object-oriented-Pascal-based development tool, made possible the very rapid development of basic user interfaces. Delphi allowed the

developer to build highly sophisticated applications with very little code writing. It also facilitated the straightforward connection to a relational DBMS. In this case the Borland's Local InterBase Server™ was used, but the system could very easily be scaled up to connect to the Oracle™ server that will soon come on-line at NPS.

Delphi supports Object Oriented paradigm as well as a real time environment. Since Delphi uses an optimizing native code compiler to generate machine code instead of p-code for interpreted environments, it creates faster, portable, and directly executable applications. Although Borland states that Delphi applications run 30-40 times faster, for the applications, such as IPOMIS applications which access database heavily and run SQL scripts, this dramatic speed difference will be less. In this case, underlying database engine and the forms of accessing to the database makes the difference as well as how good the developer is in writing efficient SQL scripts.

B. RECOMMENDATIONS

These delivered applications represented the core functionality of the new information system, yet they were designed from the ground up to be extensible. It would be very straightforward to add more customized reports, on-line help, and effective memory usage features.

All throughout the development process, end users could have strong inputs into how the applications would look. This became a big factor in the applications' acceptance and the productivity level of users as they employed the system. Although user input is very important to the development process, it certainly should not dictate how the application is developed. The input of the developer and his decisions about user inputs should be the real driving force behind design decisions. User inputs are important, but they must be balanced with the knowledge and analysis provided by the developer.

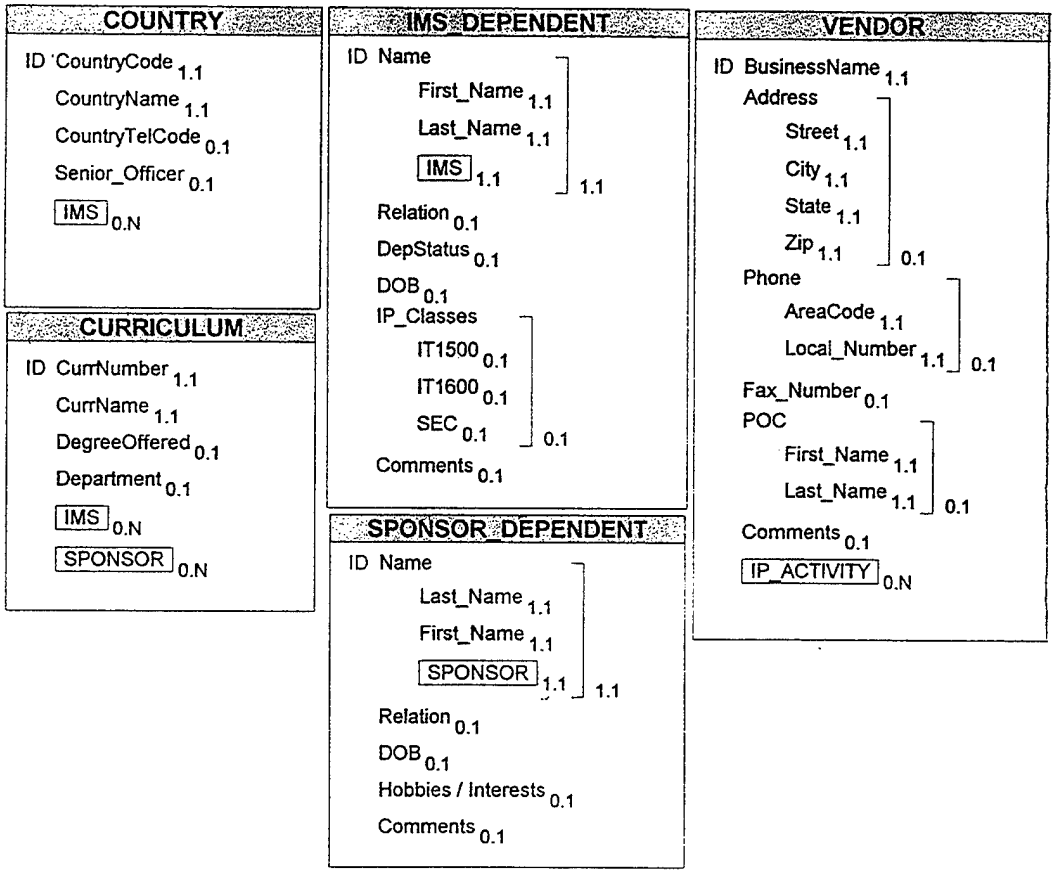
Utilizing the implementation-independent modeling tools, such as Semantic Object modeling and Data Flow Diagrams empowers the developer to overcome the "we have always done it that way" syndrome, reduce the risk of missing data and process

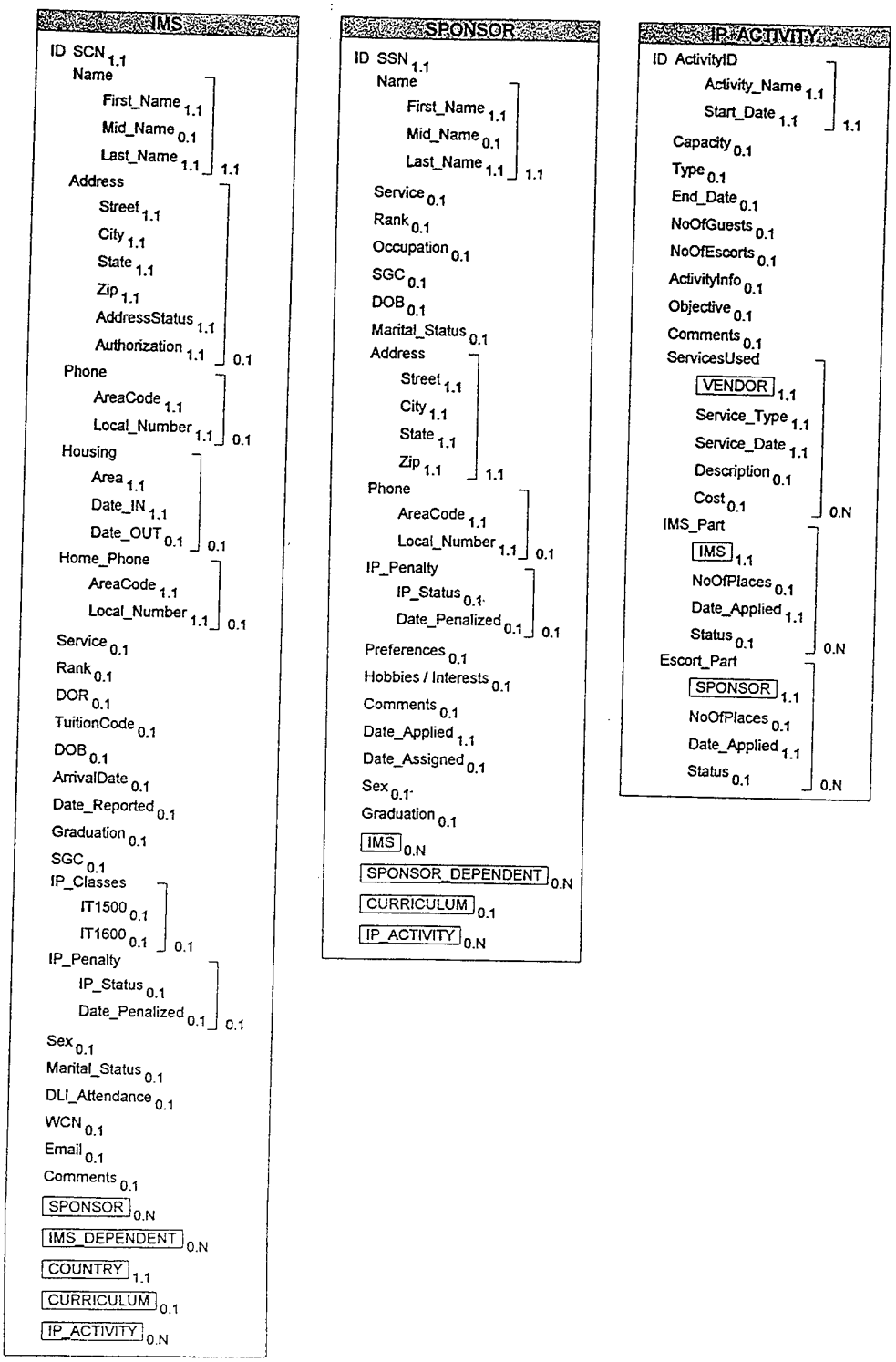
requirements by separating what the system must do from how the system will do it, and communicate with the users in less technical language.

Getting the interface right at the outset saved coding changes later, though visual tools made any interface changes very easy.

Delivering the applications into users hands as early in the development cycle as possible allowed the developer to make corrections and fix defects early in the development cycle rather than late, when they would be far more difficult to deal with.

APPENDIX A. DATA MODEL




IMS
ID SCN 1.1
Name
First_Name 1.1
Mid_Name 0.1
Last_Name 1.1
1.1
Address
Street 1.1
City 1.1
State 1.1
Zip 1.1
AddressStatus 1.1
Authorization 1.1
0.1
Phone
AreaCode 1.1
Local_Number 1.1
0.1
Housing
Area 1.1
Date_IN 1.1
Date_OUT 0.1
0.1
Home_Phone
AreaCode 1.1
Local_Number 1.1
0.1
Service 0.1
Rank 0.1
DOR 0.1
TuitionCode 0.1
DOB 0.1
ArrivalDate 0.1
Date_Reported 0.1
Graduation 0.1
SGC 0.1
IP_Classes
IT1500 0.1
IT1600 0.1
0.1
IP_Penalty
IP_Status 0.1
Date_Penalized 0.1
0.1
Sex 0.1
Marital_Status 0.1
DLI_Attendance 0.1
WCN 0.1
Email 0.1
Comments 0.1
SPONSOR 0.N
IMS_DEPENDENT 0.N
COUNTRY 1.1
CURRICULUM 0.1
IP_ACTIVITY 0.N
SPONSOR
ID SSN 1.1
Name
First_Name 1.1
Mid_Name 0.1
Last_Name 1.1
1.1
Service 0.1
Rank 0.1
Occupation 0.1
SGC 0.1
DOB 0.1
Marital_Status 0.1
Address
Street 1.1
City 1.1
State 1.1
Zip 1.1
1.1
Phone
AreaCode 1.1
Local_Number 1.1
0.1
IP_Penalty
IP_Status 0.1
Date_Penalized 0.1
0.1
Preferences 0.1
Hobbies / Interests 0.1
Comments 0.1
Date_Applied 1.1
Date_Assigned 0.1
Sex 0.1
Graduation 0.1
IMS 0.N
SPONSOR_DEPENDENT 0.N
CURRICULUM 0.1
IP_ACTIVITY 0.N
IP_ACTIVITY
ID ActivityID
Activity_Name 1.1
Start_Date 1.1
1.1
Capacity 0.1
Type 0.1
End_Date 0.1
NoOfGuests 0.1
NoOfEscorts 0.1
ActivityInfo 0.1
Objective 0.1
Comments 0.1
ServicesUsed
VENDOR 1.1
Service_Type 1.1
Service_Date 1.1
Description 0.1
Cost 0.1
0.N
IMS_Part
IMS 1.1
NoOfPlaces 0.1
Date_Applied 1.1
Status 0.1
0.N
Escort_Part
SPONSOR 1.1
NoOfPlaces 0.1
Date_Applied 1.1
Status 0.1
0.N

Semantic Object Report

Album: IPOMIS.ALB

COUNTRY Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
CountryCode	Unique	1	1	Text	2	
CountryName	None	1	1	Text	20	
CountryTelCode	None	0	1	Text	3	
Senior_Officer	None	0	1	Text	9	
IMS	None	0	N (No Limit)	Semantic Object		

CURRICULUM Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
CurrNumber	Unique	1	1	Text	3	
CurrName	None	1	1	Text	50	
DegreeOffered	None	0	1	Text	60	
Department	None	0	1	Text	40	
IMS	None	0	N (No Limit)	Semantic Object		
SPONSOR	None	0	N (No Limit)	Semantic Object		

IMS Semantic Object

Caption: International Military Student
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
SCN	Unique	1	1	Text	9	
Name	None	1	1	Group		
First_Name	None	1	1	Text	20	
Mid_Name	None	0	1	Text	20	
Last_Name	None	1	1	Text	35	
Address	None	0	1	Group		
Street	None	1	1	Text	35	
City	None	1	1	Text	25	
State	None	1	1	Text	2	
Zip	None	1	1	Text	5	
AddressStatus	None	1	1	Text	9	
Authorization	None	1	1	Text	3	
Phone	None	0	1	Group		
AreaCode	None	1	1	Text	3	
Local_Number	None	1	1	Text	8	
Housing	None	0	1	Group		
Area	None	1	1	Text	10	
Date_IN	None	1	1	Date		
Date_OUT	None	0	1	Date		
Home_Phone	None	0	1	Group		
AreaCode	None	1	1	Text	5	
Local_Number	None	1	1	Text	10	
Service	None	0	1	Text	2	
Rank	None	0	1	Text	5	
DOR	None	0	1	Text	5	
TuitionCode	None	0	1	Text	4	
DOB	None	0	1	Date		
ArrivalDate	None	0	1	Date		
Date_Reported	None	0	1	Date		
Graduation	None	0	1	Date		
SGC	None	0	1	Text	4	
IP_Classes	None	0	1	Group		
IT1500	None	0	1	Text	3	
IT1600	None	0	1	Text	3	
IP_Penalty	None	0	1	Group		
IP_Status	None	0	1	Tiny Integer		
Date_Penalized	None	0	1	Date		
Sex	None	0	1	Text	1	
Marital_Status	None	0	1	Text	1	
DLI_Attendance	None	0	1	Text	3	
WCN	None	0	1	Text	4	
Email	None	0	1	Text	35	
Comments	None	0	1	Memo		
SPONSOR	None	0	N (No Limit)	Semantic Object		
IMS_DEPENDENT	None	0	N (No Limit)	Semantic Object		
COUNTRY	None	1	1	Semantic Object		
CURRICULUM	None	0	1	Semantic Object		
IP_ACTIVITY	None	0	N (No Limit)	Semantic Object		

IMS_DEPENDENT Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Name	Unique	1	1	Group		
First_Name	None	1	1	Text	20	
Last_Name	None	1	1	Text	35	
IMS	None	1	1	Semantic Object		
Relation	None	0	1	Text	1	
DepStatus	None	0	1	Text	3	
DOB	None	0	1	Date		
IP_Classes	None	0	1	Group		
IT1500	None	0	1	Text	3	
IT1600	None	0	1	Text	3	
SEC	None	0	1	Text	3	
Comments	None	0	1	Memo		

IP_ACTIVITY Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
ActivityID	Unique	1	1	Group		
Activity_Name	None	1	1	Text	35	
Start_Date	None	1	1	Date		
Capacity	None	0	1	Short Integer		
Type	None	0	1	Text	8	
End_Date	None	0	1	Date		
NoOfGuests	None	0	1	Short Integer		
NoOfEscorts	None	0	1	Short Integer		
ActivityInfo	None	0	1	Memo		
Objective	None	0	1	Memo		
Comments	None	0	1	Memo		
ServicesUsed	None	0	N (No Limit)	Group		
VENDOR	None	1	1	Semantic Object		
Service_Type	None	1	1	Text	15	
Service_Date	None	1	1	Date		
Description	None	0	1	Text	50	
Cost	None	0	1	Currency		
IMS_Part	None	0	N (No Limit)	Group		
IMS	None	1	1	Semantic Object		
NoOfPlaces	None	0	1	Short Integer		
Date_Applied	None	1	1	Date		
Status	None	0	1	Text	1	
Escort_Part	None	0	N (No Limit)	Group		
SPONSOR	None	1	1	Semantic Object		
NoOfPlaces	None	0	1	Short Integer		
Date_Applied	None	1	1	Date		
Status	None	0	1	Text	1	

SPONSOR Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
SSN	Unique	1	1	Text	9	
Name	None	1	1	Group		
First_Name	None	1	1	Text	20	
Mid_Name	None	0	1	Text	20	
Last_Name	None	1	1	Text	35	
Service	None	0	1	Text	2	
Rank	None	0	1	Text	5	
Occupation	None	0	1	Text	35	
SGC	None	0	1	Text	4	
DOB	None	0	1	Date		
Marital_Status	None	0	1	Text	1	
Address	None	1	1	Group		
Street	None	1	1	Text	35	
City	None	1	1	Text	25	
State	None	1	1	Text	2	
Zip	None	1	1	Text	5	
Phone	None	0	1	Group		
AreaCode	None	1	1	Text	3	
Local_Number	None	1	1	Text	8	
IP_Penalty	None	0	1	Group		
IP_Status	None	0	1	Tiny Integer		
Date_Penalized	None	0	1	Date		
Preferences	None	0	1	Memo		
Hobbies / Interests	None	0	1	Memo		
Comments	None	0	1	Memo		
Date_Applied	None	1	1	Date		
Date_Assigned	None	0	1	Date		
Sex	None	0	1	Text	1	
Graduation	None	0	1	Date		
IMS	None	0	N (No Limit)	Semantic Object		
SPONSOR_ DEPENDENT	None	0	N (No Limit)	Semantic Object		
CURRICULUM	None	0	1	Semantic Object		
IP_ACTIVITY	None	0	N (No Limit)	Semantic Object		

SPONSOR_DEPENDENT Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
Name	Unique	1	1	Group		
Last_Name	None	1	1	Text	35	
First_Name	None	1	1	Text	20	
SPONSOR	None	1	1	Semantic Object		
Relation	None	0	1	Text	1	
DOB	None	0	1	Date		
Hobbies / Interests	None	0	1	Memo		
Comments	None	0	1	Memo		

VENDOR Semantic Object

Caption:
Description:

Data Attributes:

Attribute Name	ID Status	Minimum Required	Maximum Allowed	Value Type	Length	Formula Expression
BusinessName	Non-unique	1	1	Text	30	
Address	None	0	1	Group		
Street	None	1	1	Text	35	
City	None	1	1	Text	25	
State	None	1	1	Text	2	
Zip	None	1	1	Text	5	
Phone	None	0	1	Group		
AreaCode	None	1	1	Text	3	
Local_Number	None	1	1	Text	8	
Fax_Number	None	0	1	Text	8	
POC	None	0	1	Group		
First_Name	None	1	1	Text	20	
Last_Name	None	1	1	Text	35	
Comments	None	0	1	Memo		
IP_ACTIVITY	None	0	N (No Limit)	Semantic Object		

Attribute Report

Activity_Name	Type: Simple Value Profile: Activity_Name Contained in: IP_ACTIVITY.ActivityID Caption: Description: The name of the IP activity ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: Initial Value:		
ActivityID	Type: Group Profile: ActivityID Contained in: IP_ACTIVITY Caption: Description: Uniquelly describes an IP activity ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	Activity_Name Start_Date
ActivityInfo	Type: Simple Value Profile: ActivityInfo Contained in: IP_ACTIVITY Caption: Description: Includes information about an activi ty organized ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Initial Value:		
Address	Type: Group Profile: Address_1 Contained in: VENDOR Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	Street City State Zip
Address	Type: Group Profile: Address Contained in: IMS Caption: Description: Address of an IMS in U.S. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	Street City State Zip AddressStatus Authorization
Address	Type: Group Profile: Address_1 Contained in: SPONSOR Caption: Description: Address of a person in U.S. ID Status: None Minimum Required: 1 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	Street City State Zip

AddressStatus	Type: Simple Value Profile: AddressStatus Contained in: IMS.Address Caption: Status of Address Description: An address may be Temporary or Per manent ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 9 Format: Temporary, Permanent Initial Value:
Area	Type: Simple Value Profile: Area Contained in: IMS.Housing Caption: Description: The Area name of the housing / BOQ ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 10 Format: BOQ, La Mesa, POM Annex (Fort Ord) Initial Value:
AreaCode	Type: Simple Value Profile: AreaCode Contained in: IMS.Home_Phone Caption: Area Code Description: Telefon system area code number ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 5 Format: NNNNN Initial Value:
AreaCode	Type: Simple Value Profile: AreaCode Contained in: VENDOR.Phone Caption: Area Code Description: US telefon system area code number ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 3 Format: NNN Initial Value: 408
AreaCode	Type: Simple Value Profile: AreaCode Contained in: IMS.Phone Caption: Area Code Description: US telefon system area code number ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 3 Format: NNN Initial Value: 408

AreaCode	Type: Simple Value Profile: AreaCode Contained in: SPONSOR.Phone Caption: Area Code Description: US telefon system area code number ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 3 Format: NNN Initial Value: 408
ArrivalDate	Type: Simple Value Profile: ArrivalDate Contained in: IMS Caption: Arrival Date to the U.S. Description: The date when an IMET student lands on the U.S., for a FMS student it is t he same as reporting date. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Authorization	Type: Simple Value Profile: Authorization Contained in: IMS.Address Caption: Description: Authorization to release an IMS's ad dress. ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 3 Format: Yes / No Initial Value: No
BusinessName	Type: Simple Value Profile: BusinessName Contained in: VENDOR Caption: Description: Business name of the IP activity ven dor. ID Status: Non-unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 30 Format: Initial Value:
Capacity	Type: Simple Value Profile: Capacity Contained in: IP_ACTIVITY Caption: Capacity of the Activity Description: Describes the max number of IMS or t heir dependents who can participate th e activity. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Short Integer Length: Format: NN Initial Value:

City	Type: Simple Value Profile: City Contained in: SPONSOR.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 25 Format: Initial Value:
City	Type: Simple Value Profile: City Contained in: VENDOR.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 25 Format: Initial Value:
City	Type: Simple Value Profile: City Contained in: IMS.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 25 Format: Initial Value:
Comments	Type: Simple Value Profile: Comments Contained in: SPONSOR Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Plain text Initial Value:
Comments	Type: Simple Value Profile: Comments Contained in: IP_ACTIVITY Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Plain text Initial Value:

Comments	Type: Simple Value Profile: Comments Contained in: IMS_DEPENDENT Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Plain text Initial Value:
Comments	Type: Simple Value Profile: Comments Contained in: SPONSOR_DEPENDENT Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Plain text Initial Value:
Comments	Type: Simple Value Profile: Comments Contained in: IMS Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Plain text Initial Value:
Comments	Type: Simple Value Profile: Comments Contained in: VENDOR Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Plain text Initial Value:
Cost	Type: Simple Value Profile: Cost Contained in: IP_ACTIVITY.ServicesUsed Caption: Description: Cost of services consumed during an IP activity ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Currency Length: Format: Initial Value:
COUNTRY	Type: Object Link Profile: COUNTRY Contained in: IMS Caption: Description: Includes information about a country ID Status: None Minimum Required: 1 Maximum Allowed: 1

CountryCode	Type: Simple Value Profile: CountryCode Contained in: COUNTRY Caption: Country Code Letters Description: Two position alphabetical code used to identify a foreign country, in accordance with the Security Assis tance Management Manual ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 2 Format: AA Initial Value:
CountryName	Type: Simple Value Profile: CountryName Contained in: COUNTRY Caption: Country Name in English Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 20 Format: All uppercase Initial Value:
CountryTelCode	Type: Simple Value Profile: CountryTelCode Contained in: COUNTRY Caption: Country International Telefon Code Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 3 Format: NNN Initial Value:
CURRICULUM	Type: Object Link Profile: CURRICULUM Contained in: SPONSOR Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
CURRICULUM	Type: Object Link Profile: CURRICULUM Contained in: IMS Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1
CurrName	Type: Simple Value Profile: CurrName Contained in: CURRICULUM Caption: Curriculum Name Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 50 Format: Initials Uppercase Initial Value:

CurrNumber	Type: Simple Value Profile: CurrNumber Contained in: CURRICULUM Caption: Curriculum Number Description: The number of the curriculum attende d by the student. ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 3 Format: NNN Initial Value:
Date_Applied	Type: Simple Value Profile: Date_Applied Contained in: IP_ACTIVITY.IMS_Part Caption: Description: The date when an IMS applied to part icipate to an IP activity. ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Date_Applied	Type: Simple Value Profile: Date_Applied Contained in: IP_ACTIVITY.Escort_Part Caption: Description: The date when an Sponsor applied to participate to an IP activity. ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Date_Applied	Type: Simple Value Profile: Date_Applied Contained in: SPONSOR Caption: Description: The date when a sponsor applied to b e a sponsor. ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Date_Assigned	Type: Simple Value Profile: Date_Assigned Contained in: SPONSOR Caption: Description: The date when the sponsor officially assigned as a sponsor. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:

Date_IN	Type: Simple Value Profile: Date_IN Contained in: IMS.Housing Caption: Description: The date when IMS got the keys to mi litary housing or BOQ ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Date_OUT	Type: Simple Value Profile: Date_OUT Contained in: IMS.Housing Caption: Description: The date when IMS moved out from m ilitary housing or BOQ ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Date_Penalized	Type: Simple Value Profile: Date_Penalized Contained in: IMS.IP_Penalty Caption: Description: Date when an IMS or sponsor is penal ized because of late confirmation or n ot showing up. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Date_Penalized	Type: Simple Value Profile: Date_Penalized Contained in: SPONSOR.IP_Penalty Caption: Description: Date when an IMS or sponsor is penal ized because of late confirmation or n ot showing up. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Date_Reported	Type: Simple Value Profile: Date_Reported Contained in: IMS Caption: Description: The date when an IMS reports to the NPS IPO. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:

DegreeOffered	Type: Simple Value Profile: DegreeOffered Contained in: CURRICULUM Caption: Description: Academic Degree Offered by a Curric ulum ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 60 Format: Initials Uppercase Initial Value:
Department	Type: Simple Value Profile: Department Contained in: CURRICULUM Caption: Description: The Name of the NPS academic depar tment ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 40 Format: Initials Uppercase Initial Value:
DepStatus	Type: Simple Value Profile: DepStatus Contained in: IMS_DEPENDENT Caption: Dependent Status Description: Describes whether an IMS's dependen t is with him/her in the US. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 3 Format: Yes, No Initial Value:
Description	Type: Simple Value Profile: Description Contained in: IP_ACTIVITY.ServicesUsed Caption: Description: Description of the service provided for an activity ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 50 Format: Initial Value:
DLI_Attendance	Type: Simple Value Profile: DLI_Attendance Contained in: IMS Caption: Description: Some IMS may be attending ESL cour ses in DLI/Texas prior to NPS attendance. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 3 Format: Yes/No Initial Value: No

DOB	Type: Simple Value Profile: DOB Contained in: IMS Caption: Date Of Birth Description: Date Of Birth of a person ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
DOB	Type: Simple Value Profile: DOB Contained in: SPONSOR Caption: Date Of Birth Description: Date Of Birth of a person ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
DOB	Type: Simple Value Profile: DOB Contained in: IMS_DEPENDENT Caption: Date Of Birth Description: Date Of Birth of a person ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
DOB	Type: Simple Value Profile: DOB Contained in: SPONSOR_DEPENDENT Caption: Date Of Birth Description: Date Of Birth of a person ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
DOR	Type: Simple Value Profile: DOR Contained in: IMS Caption: Date Of Rank Description: Date Of Rank of a military service m ember. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 5 Format: MM/YY Initial Value:

Email	Type: Simple Value Profile: Email Contained in: IMS Caption: E-mail address Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 35 Format: Initial Value:	
End_Date	Type: Simple Value Profile: End_Date Contained in: IP_ACTIVITY Caption: IP Activity End Date Description: Describes the date when an IP activi ty ends ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:	
Escort_Part	Type: Group Profile: Escort_Part Contained in: IP_ACTIVITY Caption: Description: Includes information about Sponsor/E scort who participated the activity. ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit) Minimum Count: 0 Maximum Count: ALL	Attributes Contained: SPONSOR NoOfPlaces Date_Applied Status
Fax_Number	Type: Simple Value Profile: Fax_Number Contained in: VENDOR Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 8 Format: NNN-NNNN Initial Value:	
First_Name	Type: Simple Value Profile: First_Name Contained in: IMS.Name Caption: Description: First name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 20 Format: Initial Uppercase Initial Value:	

First_Name	Type: Simple Value Profile: First_Name Contained in: IMS_DEPENDENT.Name Caption: Description: First name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 20 Format: Initial Uppercase Initial Value:
First_Name	Type: Simple Value Profile: First_Name Contained in: VENDOR.POC Caption: Description: First name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 20 Format: Initial Uppercase Initial Value:
First_Name	Type: Simple Value Profile: First_Name Contained in: SPONSOR_DEPENDENT.Name Caption: Description: First name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 20 Format: Initial Uppercase Initial Value:
First_Name	Type: Simple Value Profile: First_Name Contained in: SPONSOR.Name Caption: Description: First name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 20 Format: Initial Uppercase Initial Value:
Graduation	Type: Simple Value Profile: Graduation Contained in: IMS Caption: Graduation Date Description: The graduation date of a student fro m NPS. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:

Graduation	Type: Simple Value Profile: Graduation Contained in: SPONSOR Caption: Graduation Date Description: The graduation date of a student fro m NPS. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:		
Hobbies / Interests	Type: Simple Value Profile: Hobbies / Interests Contained in: SPONSOR_DEPENDENT Caption: Description: Hobbies / Interests of sponsor or hi s/her dependents. Used for the sponsor assignment purpo ses. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Plain Text Initial Value:		
Hobbies / Interests	Type: Simple Value Profile: Hobbies / Interests Contained in: SPONSOR Caption: Description: Hobbies / Interests of sponsor or hi s/her dependents. Used for the sponsor assignment purpo ses. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Plain Text Initial Value:		
Home_Phone	Type: Group Profile: Home_Phone Contained in: IMS Caption: Phone Number in Home Country Description: IMS's phone number in his/her home c ountry. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	AreaCode Local_Number
Housing	Type: Group Profile: Housing Contained in: IMS Caption: Military Housing / BOQ Info Description: Includes information related with ho using or BOQ ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	Area Date_IN Date_OUT

IMS	Type: Object Link Profile: IMS Contained in: COUNTRY Caption: International Military Student Description: Students accepted to NPS from foreig n military and government civilian services ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)	
IMS	Type: Object Link Profile: IMS Contained in: CURRICULUM Caption: International Military Student Description: Students accepted to NPS from foreig n military and government civilian services ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)	
IMS	Type: Object Link Profile: IMS Contained in: SPONSOR Caption: International Military Student Description: International Student Accepted to at tend NPS ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)	
IMS	Type: Object Link Profile: IMS Contained in: IMS_DEPENDENT.Name Caption: International Military Student Description: Students accepted to NPS from foreig n military and government civilian services ID Status: None Minimum Required: 1 Maximum Allowed: 1	
IMS	Type: Object Link Profile: IMS Contained in: IP_ACTIVITY.IMS_Part Caption: International Military Student Description: Students accepted to NPS from foreig n military and government civilian services ID Status: None Minimum Required: 1 Maximum Allowed: 1	
IMS_DEPENDENT	Type: Object Link Profile: IMS_DEPENDENT Contained in: IMS Caption: Description: International Military Student's Dep endent e.g. Spouse or child ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)	
IMS_Part	Type: Group Profile: IMS_Part Contained in: IP_ACTIVITY Caption: Description: Includes information about IMS who p articipated the activity. ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit) Minimum Count: 0 Maximum Count: ALL	Attributes Contained: IMS NoOfPlaces Date_Applied Status

IP_ACTIVITY	Type: Object Link Profile: IP_ACTIVITY Contained in: VENDOR Caption: Information Program Activity Description: IP Activity is designed to assist IM Ss in acquiring a balanced understandi ng of U.S. society, instuitions and goals. ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)		
IP_ACTIVITY	Type: Object Link Profile: IP_ACTIVITY Contained in: IMS Caption: Information Program Activity Description: IP Activity is designed to assist IM Ss in acquiring a balanced understandi ng of U.S. society, instuitions and goals. ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)		
IP_ACTIVITY	Type: Object Link Profile: IP_ACTIVITY Contained in: SPONSOR Caption: Information Program Activity Description: IP Activity is designed to assist IM Ss in acquiring a balanced understandi ng of U.S. society, instuitions and goals. ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)		
IP_Classes	Type: Group Profile: IP_Classes Contained in: IMS Caption: Description: Classes sponsored by IPO for IP purp oses. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	IT1500 IT1600
IP_Classes	Type: Group Profile: IP_Classes_1 Contained in: IMS_DEPENDENT Caption: Description: Classes sponsored by IPO for IP purp oses, including Spouses English Class ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	IT1500 IT1600 SEC
IP_Penalty	Type: Group Profile: IP_Penalty Contained in: SPONSOR Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	IP_Status Date_Penalized

IP_Penalty	Type: Group Profile: IP_Penalty Contained in: IMS Caption: Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	IP_Status Date_Penalized

IP_Status	Type: Simple Value Profile: IP_Status Contained in: SPONSOR.IP_Penalty Caption: IP Penalty Status Description: If an IMS or sponsor does not confir m his/her participation in allowable t ime or does not show up after he\she c onfirmed he/she will be penalized. If he/she is penalized twice he/she c annot participate IP activities for th e next year. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Tiny Integer Length: Format: 0= none, 1= once, 2=twice which means IMS is not eligible to pa rticipate IP_Activities for a year. Initial Value: 0

IP_Status	Type: Simple Value Profile: IP_Status Contained in: IMS.IP_Penalty Caption: IP Penalty Status Description: If an IMS or sponsor does not confir m his/her participation in allowable t ime or does not show up after he\she c onfirmed he/she will be penalized. If he/she is penalized twice he/she c annot participate IP activities for th e next year. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Tiny Integer Length: Format: 0= none, 1= once, 2=twice which means IMS is not eligible to pa rticipate IP_Activities for a year. Initial Value: 0

IT1500	Type: Simple Value Profile: IT1500 Contained in: IMS.IP_Classes Caption: IT 1500 Enrollment Description: Describes the status of IMS's enroll ment in IT 1500 class ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 3 Format: Yes / No Initial Value: No

IT1500	Type: Simple Value Profile: IT1500 Contained in: IMS_DEPENDENT.IP_Classes Caption: IT 1500 Enrollment Description: Describes the status of IMS's enroll ment in IT 1500 class ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 3 Format: Yes / No Initial Value: No
IT1600	Type: Simple Value Profile: IT1600 Contained in: IMS.IP_Classes Caption: IT 1600 Enrollment Description: Describes the status of IMS's enroll ment in IT 1500 class ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 3 Format: Yes / No Initial Value: No
IT1600	Type: Simple Value Profile: IT1600 Contained in: IMS_DEPENDENT.IP_Classes Caption: IT 1600 Enrollment Description: Describes the status of IMS's enroll ment in IT 1500 class ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 3 Format: Yes / No Initial Value: No
Last_Name	Type: Simple Value Profile: Last_Name Contained in: IMS_DEPENDENT.Name Caption: Description: Last Name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: All Uppercase Initial Value:
Last_Name	Type: Simple Value Profile: Last_Name Contained in: IMS.Name Caption: Description: Last Name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: All Uppercase Initial Value:

Last_Name	Type: Simple Value Profile: Last_Name Contained in: SPONSOR.Name Caption: Description: Last Name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: All Uppercase Initial Value:
Last_Name	Type: Simple Value Profile: Last_Name Contained in: VENDOR.POC Caption: Description: Last Name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: All Uppercase Initial Value:
Last_Name	Type: Simple Value Profile: Last_Name Contained in: SPONSOR_DEPENDENT.Name Caption: Description: Last Name of a person ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: All Uppercase Initial Value:
Local_Number	Type: Simple Value Profile: Local_Number Contained in: IMS.Phone Caption: Description: Local telefon number ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 8 Format: NNN-NNNN Initial Value:
Local_Number	Type: Simple Value Profile: Local_Number Contained in: IMS.Home_Phone Caption: Local Telefon Number Description: IMS's home country local telefon num ber. ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 10 Format: NNNNNNNNNN Initial Value:

Local_Number	Type: Simple Value Profile: Local_Number Contained in: VENDOR.Phone Caption: Description: Local telefon number ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 8 Format: NNN-NNNN Initial Value:
Local_Number	Type: Simple Value Profile: Local_Number Contained in: SPONSOR.Phone Caption: Description: Local telefon number ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 8 Format: NNN-NNNN Initial Value:
Marital_Status	Type: Simple Value Profile: Marital_Status Contained in: SPONSOR Caption: Marital Status Code Description: Code describing the marital status o f a person ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 1 Format: M=Married, S=Single, G=Geo-Bachelor Initial Value:
Marital_Status	Type: Simple Value Profile: Marital_Status Contained in: IMS Caption: Marital Status Code Description: Code describing the marital status o f a person ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 1 Format: M=Married, S=Single, G=Geo-Bachelor Initial Value:
Mid_Name	Type: Simple Value Profile: Mid_Name Contained in: SPONSOR.Name Caption: Middle Name Description: Middle name of a person ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 20 Format: Initial Uppercase Initial Value:

Mid_Name	Type: Simple Value Profile: Mid_Name Contained in: IMS.Name Caption: Middle Name Description: Middle name of a person ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 20 Format: Initial Uppercase Initial Value:		
Name	Type: Group Profile: Name_22 Contained in: SPONSOR_DEPENDENT Caption: Full Person Name Description: Includes Person First and Last name ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	Last_Name First_Name SPONSOR
Name	Type: Group Profile: Name_20 Contained in: IMS_DEPENDENT Caption: Person Name Description: Includes Person First and Last name ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	First_Name Last_Name IMS
Name	Type: Group Profile: Name Contained in: SPONSOR Caption: Full Person Name Description: Includes Person First, Middle and Last name ID Status: None Minimum Required: 1 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	First_Name Mid_Name Last_Name
Name	Type: Group Profile: Name Contained in: IMS Caption: Full Person Name Description: Includes Person First, Middle and Last name ID Status: None Minimum Required: 1 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	First_Name Mid_Name Last_Name
NoOfEscorts	Type: Simple Value Profile: NoOfEscorts Contained in: IP_ACTIVITY Caption: Number Of Escorts Description: Describes the number of escorts allo wed to participate this activity ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Short Integer Length: Format: N Initial Value:		

NoOfGuests	Type: Simple Value Profile: NoOfGuests Contained in: IP_ACTIVITY Caption: Number of Guests Description: Number of guests invited/participate d to the IP activity ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Short Integer Length: Format: NN Initial Value:
NoOfPlaces	Type: Simple Value Profile: NoOfPlaces Contained in: IP_ACTIVITY.Escort_Part Caption: Description: # of places required by an IP activi ty participant ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Short Integer Length: Format: N Initial Value: 1
NoOfPlaces	Type: Simple Value Profile: NoOfPlaces Contained in: IP_ACTIVITY.IMS_Part Caption: Description: # of places required by an IP activi ty participant ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Short Integer Length: Format: N Initial Value: 1
Objective	Type: Simple Value Profile: Objective Contained in: IP_ACTIVITY Caption: Objective of the IP Activity Description: Describes the objevtive of the activ ity in accordance with the IP guidelines. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Initial Value:
Occupation	Type: Simple Value Profile: Occupation Contained in: SPONSOR Caption: Description: If Sponsor is civilian, his/her occupation is required. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 35 Format: Initial Value:

Phone	Type: Group Profile: Phone Contained in: IMS Caption: Phone Number Description: Phone number of a person in US ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	AreaCode Local_Number
Phone	Type: Group Profile: Phone Contained in: SPONSOR Caption: Phone Number Description: Phone number of a person in US ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	AreaCode Local_Number
Phone	Type: Group Profile: Phone Contained in: VENDOR Caption: Phone Number Description: Phone number of a person in US ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	AreaCode Local_Number
POC	Type: Group Profile: POC Contained in: VENDOR Caption: Point Of Contact Description: ID Status: None Minimum Required: 0 Maximum Allowed: 1 Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	First_Name Last_Name
Preferences	Type: Simple Value Profile: Preferences Contained in: SPONSOR Caption: Description: Sponsor's desire or preferences on I MS whom he/she wants to host/sponsor. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Memo Length: Format: Initial Value:		
Rank	Type: Simple Value Profile: Rank Contained in: IMS Caption: Rank / Title Description: U.S. equivalent rank of IMS's rank. For civilians Mr./Mrs or Ms. will be used. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 5 Format: Initial Value:		

Rank

Type: Simple Value
Profile: Rank
Contained in: SPONSOR
Caption: Rank / Title
Description: U.S. armed forces rank.
For civilians Mr./Mrs or Ms. will be used.
ID Status: None
Minimum Required: 0
Maximum Allowed: 1
Value Type: Text
Length: 5
Format:
Initial Value:

Relation

Type: Simple Value
Profile: Relation
Contained in: IMS_DEPENDENT
Caption: Relationship Code
Description: Relationship between IMS or SPONS
OR and their dependents.
ID Status: None
Minimum Required: 0
Maximum Allowed: 1
Value Type: Text
Length: 1
Format: H=Husband, W=Wife, D=Daughter,
S=Son, etc.
Initial Value:

Relation

Type: Simple Value
Profile: Relation
Contained in: SPONSOR_DEPENDENT
Caption: Relationship Code
Description: Relationship between IMS or SPONS
OR and their dependents.
ID Status: None
Minimum Required: 0
Maximum Allowed: 1
Value Type: Text
Length: 1
Format: H=Husband, W=Wife, D=Daughter,
S=Son, etc.
Initial Value:

SCN

Type: Simple Value
Profile: SCN
Contained in: IMS
Caption: Student Control Number
Description: A nine position identification numbe
r consisting of country code,
calendar year,
julian date and consecutive number.
Used to uniquely identify each IMS.
Assigned by NETSAFA at the first trai
ning activity attended.
ID Status: Unique
Minimum Required: 1
Maximum Allowed: 1
Value Type: Text
Length: 9
Format: AANNNNNNN
Initial Value:

SEC

Type: Simple Value
Profile: SEC
Contained in: IMS_DEPENDENT.IP_Classes
Caption: Spouses English Classes
Description: Describes the status of an IMS's dep
endent enrollment in Spouses English
class
ID Status: None
Minimum Required: 0
Maximum Allowed: 1
Value Type: Text
Length: 3
Format: Yes / No

Senior_Officer	Type: Simple Value Profile: Senior_Officer Contained in: COUNTRY Caption: SCN of Senior Officer Description: Describes the SCN of senior officer of a country ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 9 Format: AANNNNNNN Initial Value:
Service	Type: Simple Value Profile: Service Contained in: SPONSOR Caption: Service Code Description: Country Service code ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 2 Format: N= NAVY, A=ARMY, F=AIRFORCE, M=MARINES, CG=COAST GUARD, NG=NATIONAL GUARD, C=CIVILIAN Initial Value: N
Service	Type: Simple Value Profile: Service Contained in: IMS Caption: Service Code Description: Country Service code ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 2 Format: N= NAVY, A=ARMY, F=AIRFORCE, M=MARINES, CG=COAST GUARD, NG=NATIONAL GUARD, C=CIVILIAN Initial Value: N
Service_Date	Type: Simple Value Profile: Service_Date Contained in: IP_ACTIVITY.ServicesUsed Caption: Description: Date when a service is used during an IP activity ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:
Service_Type	Type: Simple Value Profile: Service_Type Contained in: IP_ACTIVITY.ServicesUsed Caption: Service Type Description: Type of the service provided by a vendor for an activity ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 15 Format: MEALS, TRAVEL, NO COST, LODGING, OTHER Initial Value:

ServicesUsed	Type: Group Profile: ServicesUsed Contained in: IP_ACTIVITY Caption: Description: Describes the services consumed duri ng IP activity ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit) Minimum Count: 0 Maximum Count: ALL	Attributes Contained:	VENDOR Service_Type Service_Date Description Cost
Sex	Type: Simple Value Profile: Sex Contained in: SPONSOR Caption: Description: The condition or character of being female or male ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 1 Format: F=Female, M=Male Initial Value: M		
Sex	Type: Simple Value Profile: Sex Contained in: IMS Caption: Description: The condition or character of being female or male ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 1 Format: F=Female, M=Male Initial Value: M		
SGC	Type: Simple Value Profile: SGC Contained in: SPONSOR Caption: Student Guardmail Center Description: SGC box number ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 4 Format: NNNN Initial Value:		
SGC	Type: Simple Value Profile: SGC Contained in: IMS Caption: Student Guardmail Center Description: SGC box number ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 4 Format: NNNN Initial Value:		
SPONSOR	Type: Object Link Profile: SPONSOR Contained in: SPONSOR_DEPENDENT.Name Caption: IMS Sponsor Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1		

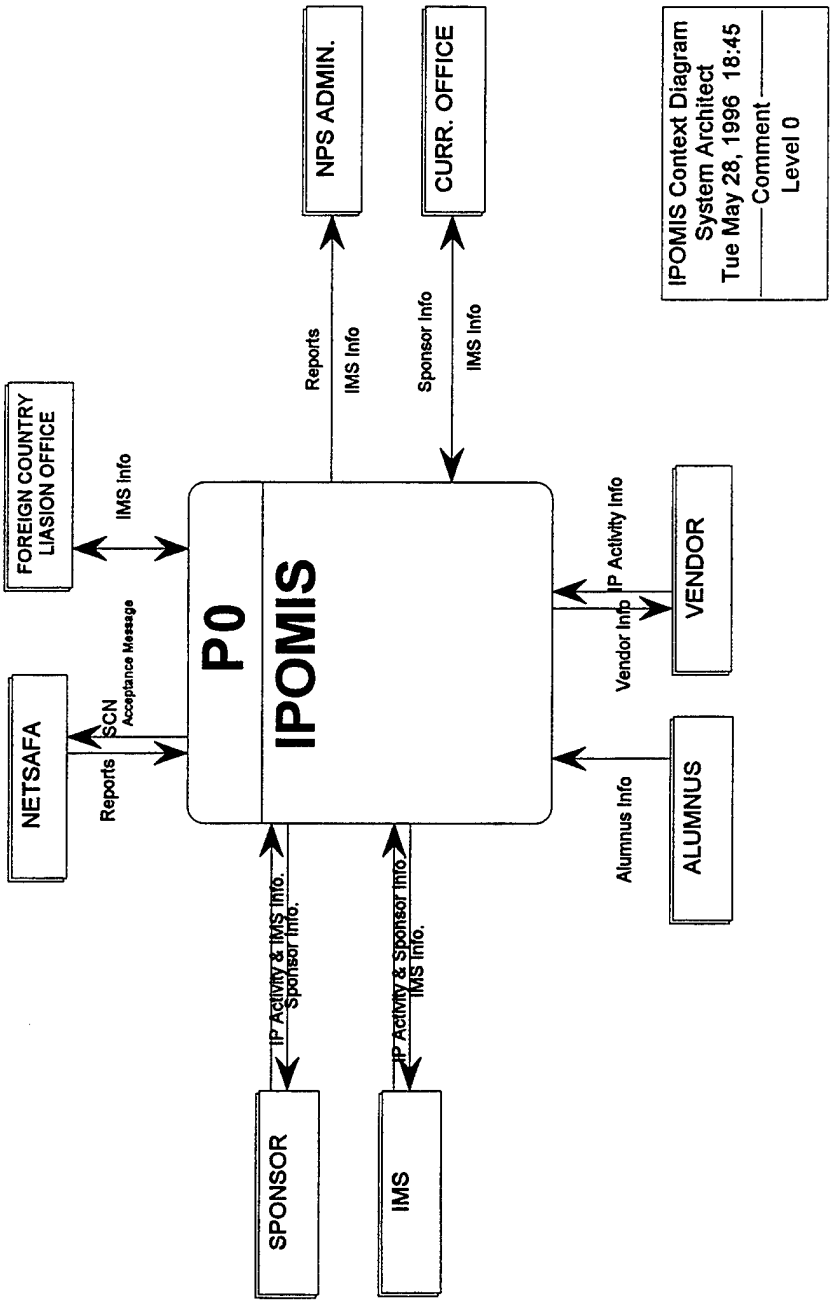
SPONSOR	Type: Object Link Profile: SPONSOR Contained in: IP_ACTIVITY.Escort_Part Caption: IMS Sponsor Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1
SPONSOR	Type: Object Link Profile: SPONSOR Contained in: IMS Caption: IMS Sponsor Description: IMS's Sponsor ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)
SPONSOR	Type: Object Link Profile: SPONSOR Contained in: CURRICULUM Caption: IMS Sponsor Description: ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)
SPONSOR_ DEPENDENT	Type: Object Link Profile: SPONSOR_DEPENDENT Contained in: SPONSOR Caption: Description: A sponsor's dependent ID Status: None Minimum Required: 0 Maximum Allowed: N (No Limit)
SSN	Type: Simple Value Profile: SSN Contained in: SPONSOR Caption: Social Security Number Description: Uniquely identifier of a person in t he U.S. ID Status: Unique Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 9 Format: NNNNNNNNN Initial Value:
Start_Date	Type: Simple Value Profile: Start_Date Contained in: IP_ACTIVITY.ActivityID Caption: IP Activity starting date Description: Describes the date when an IP activi ty starts ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Date Length: Format: DD/MM/YY Initial Value:

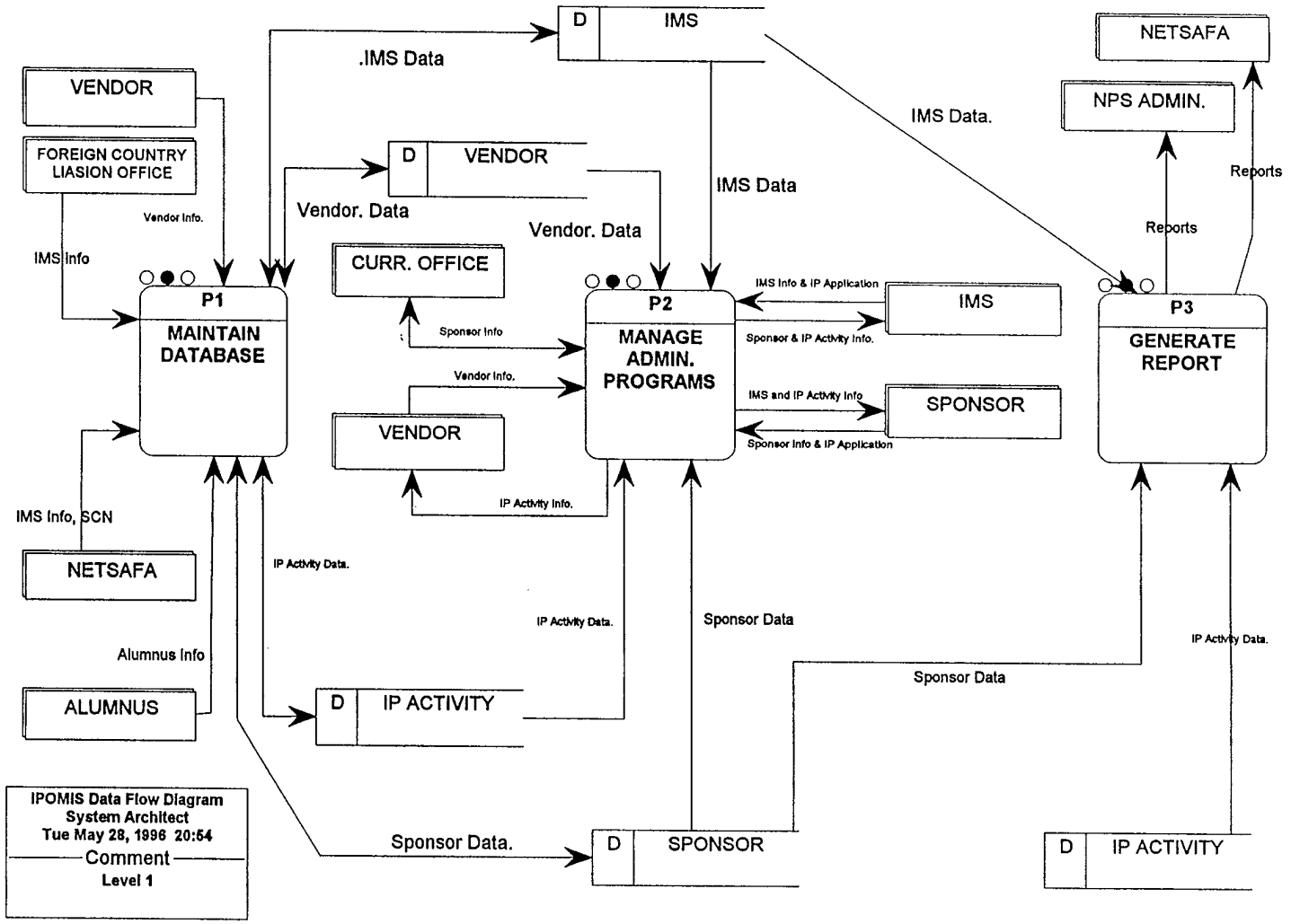
State	Type: Simple Value Profile: State Contained in: SPONSOR.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 2 Format: AA Initial Value: CA
State	Type: Simple Value Profile: State Contained in: IMS.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 2 Format: AA Initial Value: CA
State	Type: Simple Value Profile: State Contained in: VENDOR.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 2 Format: AA Initial Value: CA
Status	Type: Simple Value Profile: Status Contained in: IP_ACTIVITY.Escort_Part Caption: Description: Describes the participation status o f the participant ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 1 Format: S=Selected, C=Confirmed, W=Waiting, X=Cancelled, D=Declined, P=Prior Part. , N=No Show up Initial Value:
Status	Type: Simple Value Profile: Status Contained in: IP_ACTIVITY.IMS_Part Caption: Description: Describes the participation status o f the participant ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 1 Format: S=Selected, C=Confirmed, W=Waiting, X=Cancelled, D=Declined, P=Prior Part. , N=No Show up Initial Value:

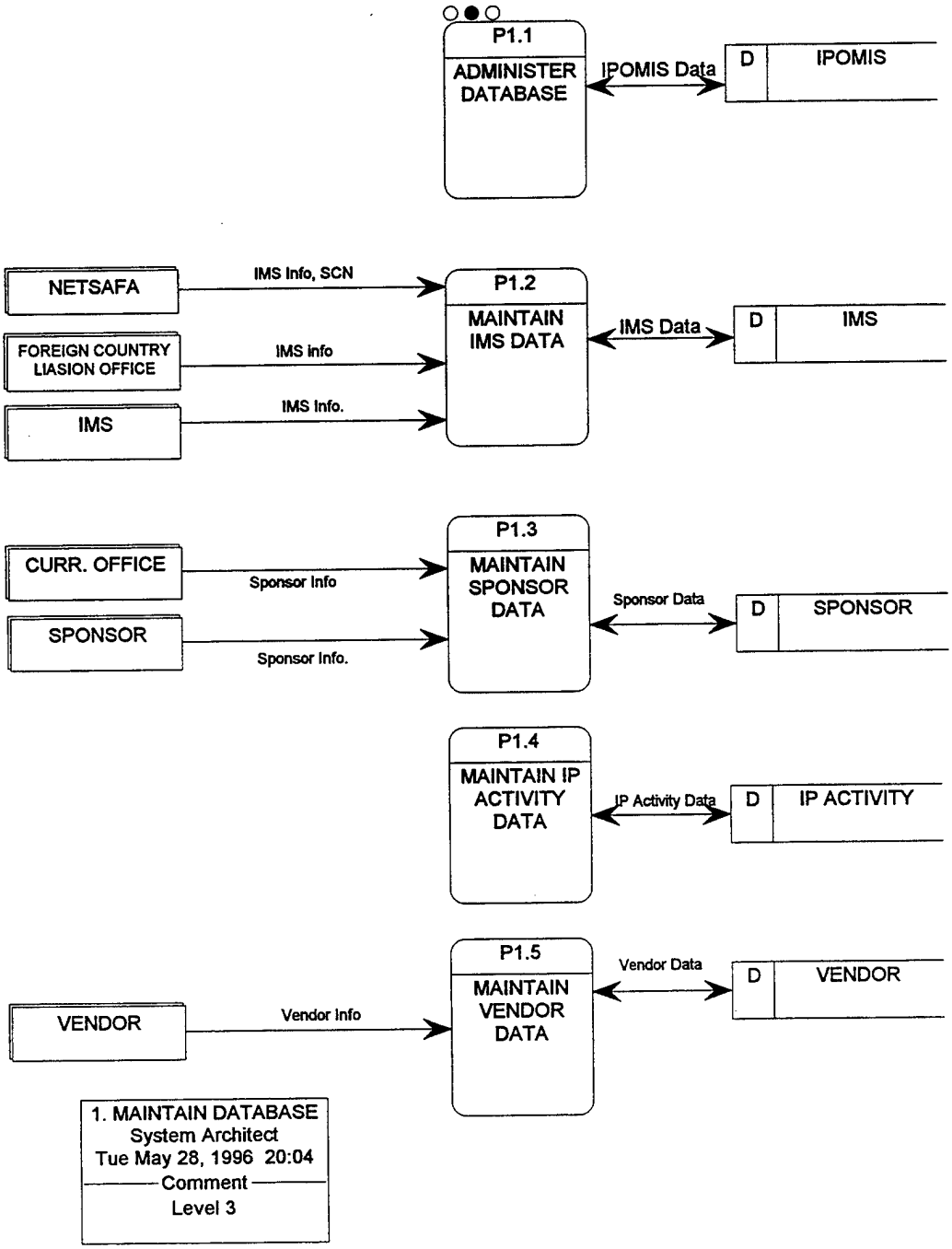
Street	Type: Simple Value Profile: Street Contained in: IMS.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: Initial Value:
Street	Type: Simple Value Profile: Street Contained in: VENDOR.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: Initial Value:
Street	Type: Simple Value Profile: Street Contained in: SPONSOR.Address Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 35 Format: Initial Value:
TuitionCode	Type: Simple Value Profile: TuitionCode Contained in: IMS Caption: Description: Describes the tuition status of IMS. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 4 Format: IMET or FMS Initial Value: FMS
Type	Type: Simple Value Profile: Type Contained in: IP_ACTIVITY Caption: Type of IP Activity Description: Describes the category of the IP act ivity. Sudent: only IMS can participat e, Adult: Spouses can participate too, Family: All family members can partic ipate on space available basis ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 8 Format: Student, Adult, Family -All Uppercase Initial Value:

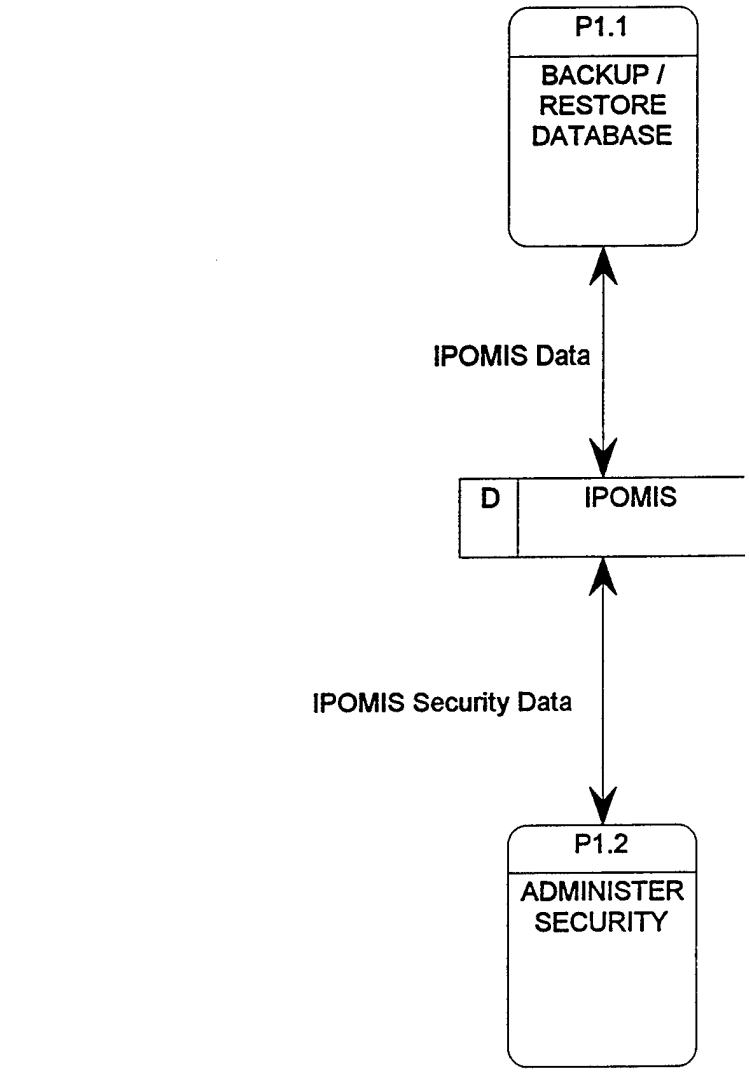
VENDOR	Type: Object Link Profile: VENDOR Contained in: IP_ACTIVITY.ServicesUsed Caption: Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1
WCN	Type: Simple Value Profile: WCN Contained in: IMS Caption: Worksheet Control Number Description: A reference number used for data con trol purposes and assigned by country liasion officer to each item of traini ng in a country's program. ID Status: None Minimum Required: 0 Maximum Allowed: 1 Value Type: Text Length: 4 Format: No Standard Format (Typical Formats: A NNN, NNNA, NNN, and NN) Initial Value:
Zip	Type: Simple Value Profile: Zip Contained in: SPONSOR.Address Caption: US or international Postal System Zip Co de Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 5 Format: NNNNN Initial Value:
Zip	Type: Simple Value Profile: Zip Contained in: VENDOR.Address Caption: US or international Postal System Zip Co de Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 5 Format: NNNNN Initial Value:
Zip	Type: Simple Value Profile: Zip Contained in: IMS.Address Caption: Postal System Zip Code Description: ID Status: None Minimum Required: 1 Maximum Allowed: 1 Value Type: Text Length: 5 Format: NNNNN Initial Value:

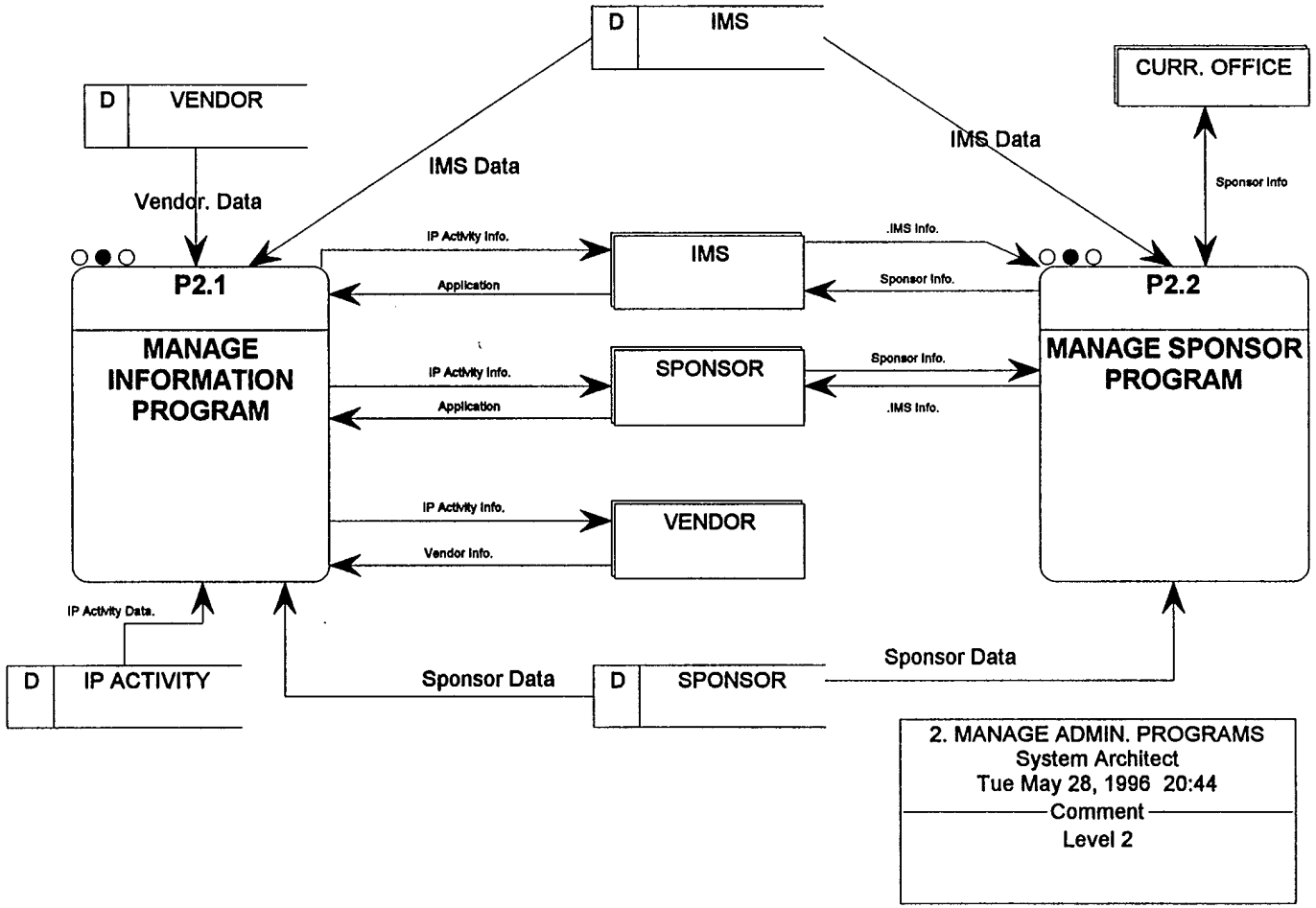
APPENDIX B. PROCESS MODEL

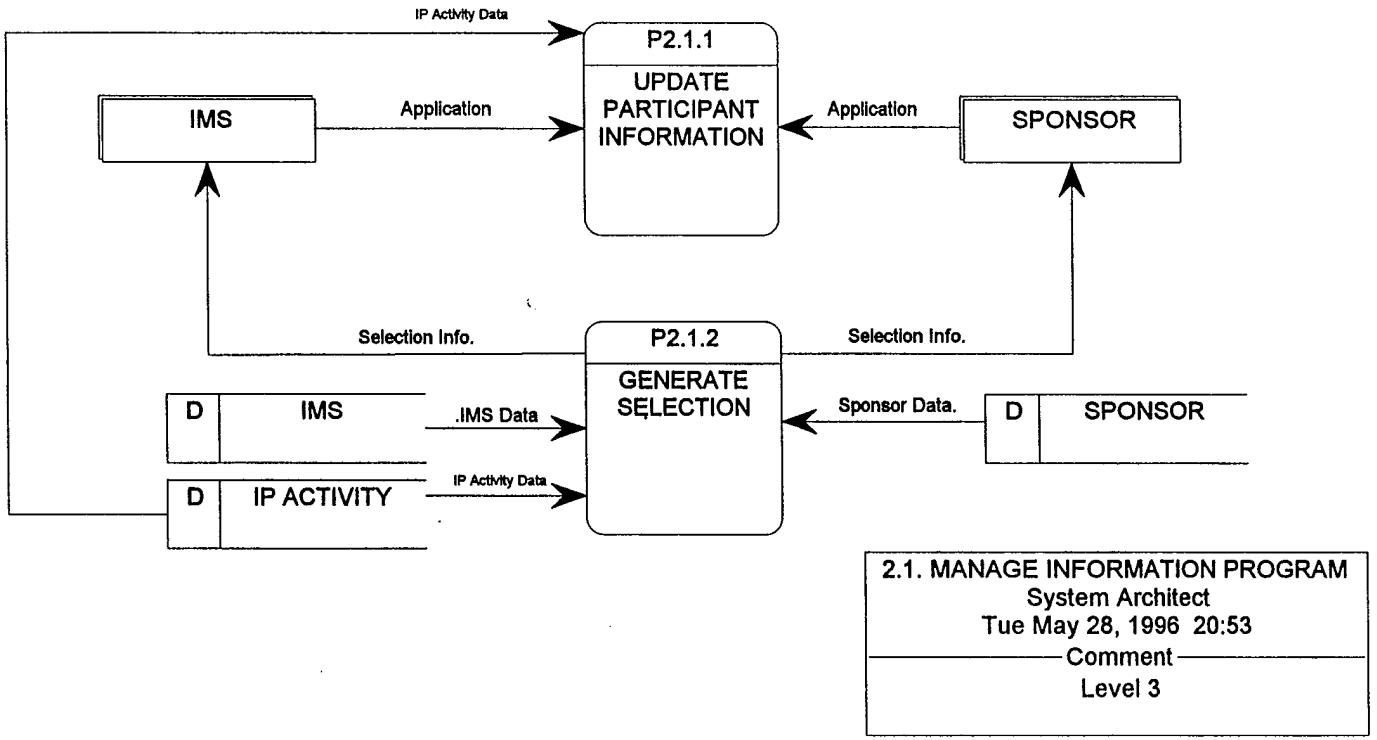


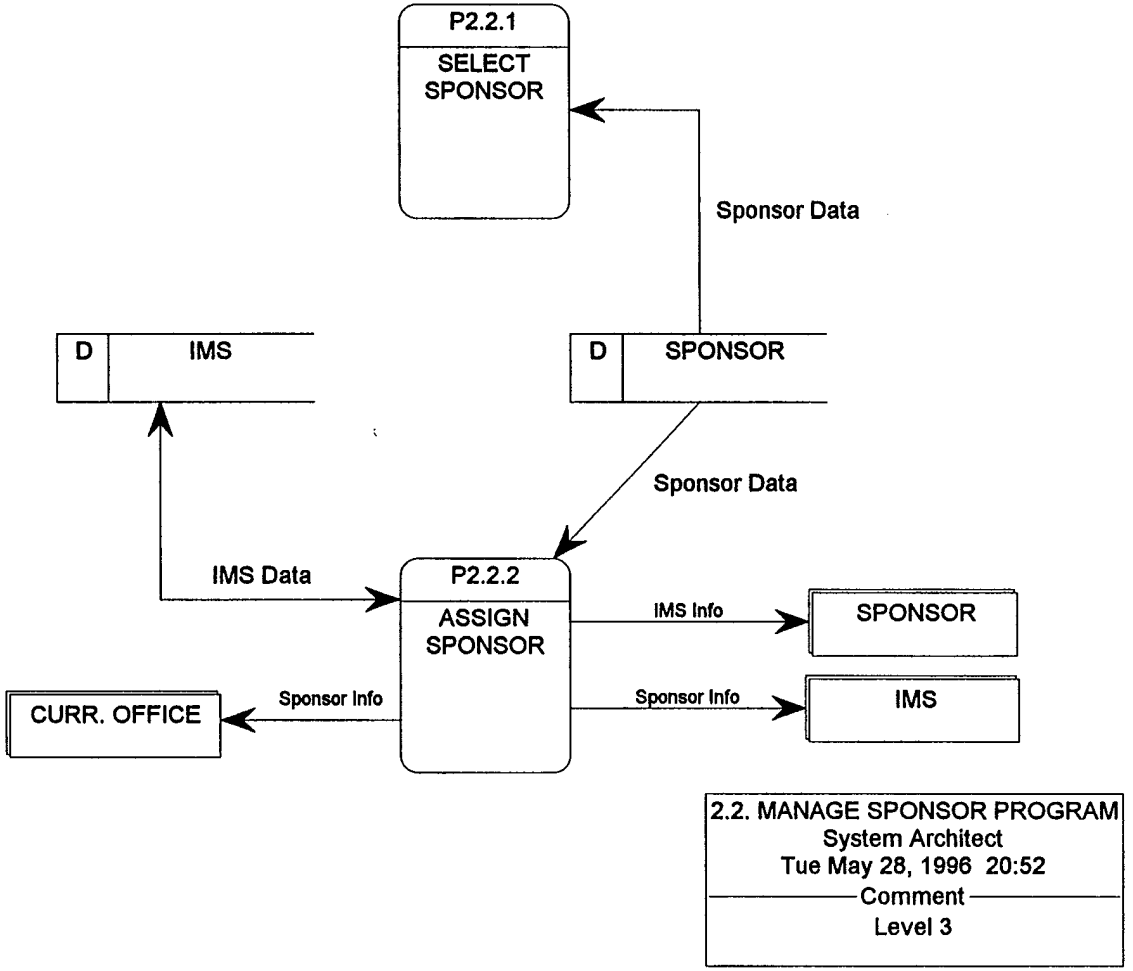

D IMS
.IMS Data
IMS Data.
NETSAFA
NPS ADMIN.
VENDOR
FOREIGN COUNTRY LIASION OFFICE
Vendor Info.
IMS Info
D VENDOR
Vendor. Data
Vendor. Data
IMS Data
Reports
Reports
CURR. OFFICE
P1
MAINTAIN DATABASE
Sponsor Info
Vendor Info.
VENDOR
IP Activity Info.
P2
MANAGE ADMIN. PROGRAMS
IMS Info & IP Application
Sponsor & IP Activity Info.
IMS
IMS and IP Activity Info
Sponsor Info & IP Application
SPONSOR
P3
GENERATE REPORT
IMS Info, SCN
NETSAFA
IP Activity Data.
IP Activity Data.
Sponsor Data
Sponsor Data
IP Activity Data.
Alumnus Info
ALUMNUS
D IP ACTIVITY
IPOMIS Data Flow Diagram
System Architect
Tue May 28, 1996 20:54
Comment
Level 1
Sponsor Data.
D SPONSOR
D IP ACTIVITY

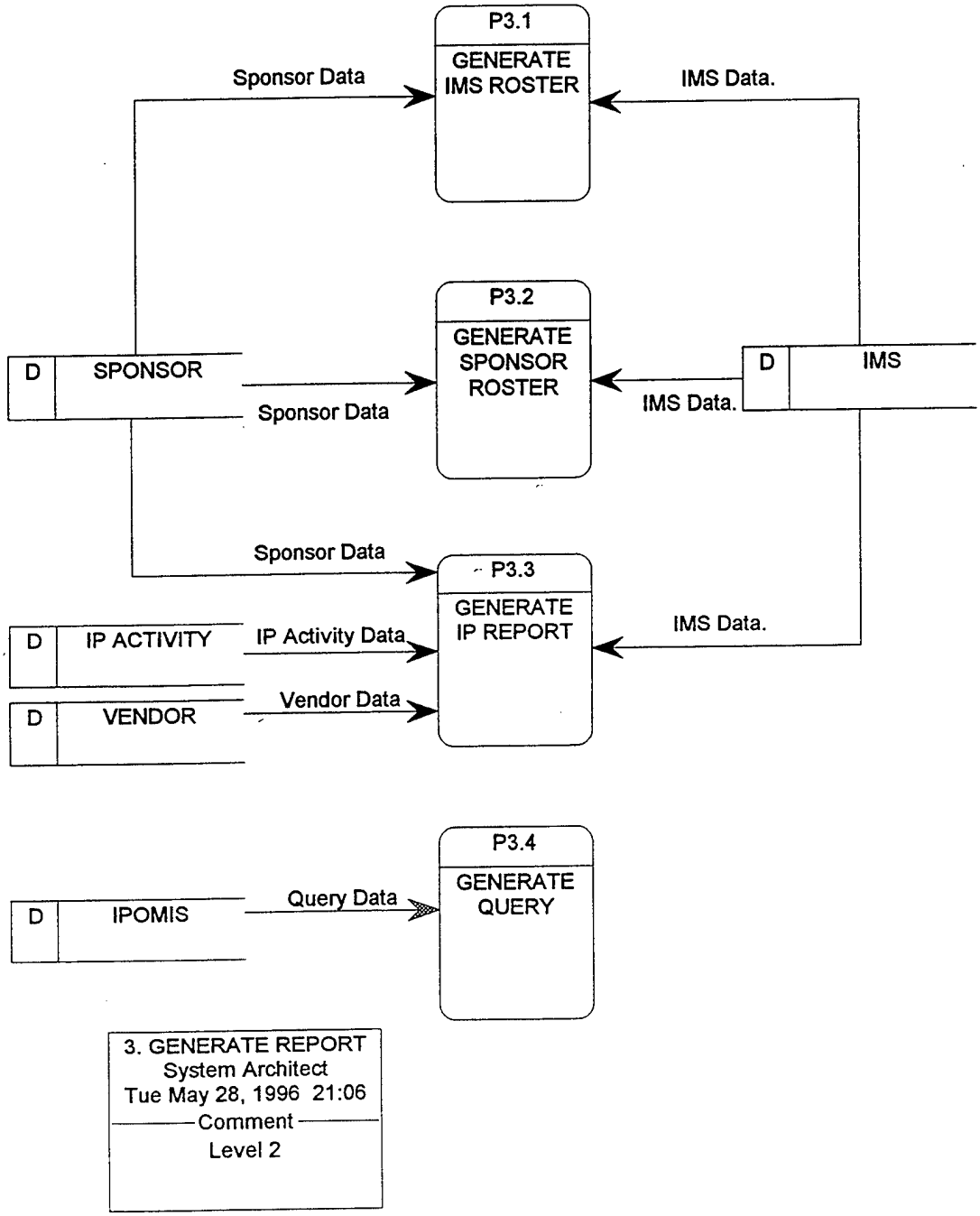

P1.1
ADMINISTER DATABASE
IPOMIS Data
D
IPOMIS
NETSAFA
IMS Info, SCN
FOREIGN COUNTRY LIASION OFFICE
IMS Info
IMS
IMS Info.
P1.2
MAINTAIN IMS DATA
IMS Data
D
IMS
CURR. OFFICE
Sponsor Info
SPONSOR
Sponsor Info.
P1.3
MAINTAIN SPONSOR DATA
Sponsor Data
D
SPONSOR
P1.4
MAINTAIN IP ACTIVITY DATA
IP Activity Data
D
IP ACTIVITY
VENDOR
Vendor Info
P1.5
MAINTAIN VENDOR DATA
Vendor Data
D
VENDOR
1. MAINTAIN DATABASE
System Architect
Tue May 28, 1996 20:04
Comment
Level 3


P1.1
BACKUP / RESTORE DATABASE
IPOMIS Data
D
IPOMIS
IPOMIS Security Data
P1.2
ADMINISTER SECURITY
1. ADMINISTER DATABASE
System Architect
Tue May 28, 1996 20:01
Comment
Level 3


D
IMS
CURR. OFFICE
D
VENDOR
IMS Data
IMS Data
Sponsor Info
Vendor. Data
IP Activity Info.
IMS
.IMS Info.
P2.1
Application
Sponsor Info.
P2.2
MANAGE INFORMATION PROGRAM
IP Activity Info.
SPONSOR
Sponsor Info.
MANAGE SPONSOR PROGRAM
Application
.IMS Info.
IP Activity Info.
VENDOR
Vendor Info.
IP Activity Data.
D
IP ACTIVITY
Sponsor Data
D
SPONSOR
Sponsor Data
2. MANAGE ADMIN. PROGRAMS
System Architect
Tue May 28, 1996 20:44
Comment
Level 2


IP Activity Data
P2.1.1
UPDATE PARTICIPANT INFORMATION
IMS
Application
Application
SPONSOR
Selection Info.
P2.1.2
GENERATE SELECTION
Selection Info.
D
IMS
.IMS Data
Sponsor Data.
D
SPONSOR
D
IP ACTIVITY
IP Activity Data
2.1. MANAGE INFORMATION PROGRAM
System Architect
Tue May 28, 1996 20:53
Comment
Level 3


P2.2.1
SELECT SPONSOR
Sponsor Data
D
IMS
D
SPONSOR
Sponsor Data
IMS Data
P2.2.2
ASSIGN SPONSOR
IMS Info
SPONSOR
CURR. OFFICE
Sponsor Info
Sponsor Info
IMS
2.2. MANAGE SPONSOR PROGRAM
System Architect
Tue May 28, 1996 20:52
Comment
Level 3


P3.1
GENERATE IMS ROSTER
Sponsor Data
IMS Data.
P3.2
GENERATE SPONSOR ROSTER
D
SPONSOR
Sponsor Data
D
IMS
IMS Data.
Sponsor Data
P3.3
GENERATE IP REPORT
IMS Data.
D
IP ACTIVITY
IP Activity Data
D
VENDOR
Vendor Data
P3.4
GENERATE QUERY
D
IPOMIS
Query Data
3. GENERATE REPORT
System Architect
Tue May 28, 1996 21:06
Comment
Level 2

APPENDIX C. RELATION DEFINITIONS

COUNTRY

Number of attributes : 4

Key attribute : COUNTRYCODE

Foreign key : None

Relationship with : IMS 1:0 (One-to-zero)

CURRICULUM

Number of attributes : 4

Key attribute : CURRNUMBER

Foreign key : None

Relationships with : IMS 1:0

SPONSOR 1:0

IMS

Number of attributes : 43

Key attribute : SCN

Foreign keys : CURRNUMBER (references CURRICULUM),

COUNTRYCODE (references COUNTRY)

Relationships with : CURRICULUM 0:1 (Zero-to-one)

COUNTRY 1:1 (One-to-one)

IMS DEPENDENT 0:N (Zero-to-many)

IMS-SPONSOR 0:N

IMS PARTICIPATION 0:N

SPONSOR

Number of attributes : 27

Key attribute : SSN

Foreign key : CURRNUMBER (references CURRICULUM relation)

Relationships with : CURRICULUM 0:1

SPONSOR DEPENDENT 0:N

IMS-SPONSOR 0:N

ESCORT PARTICIPATION 0:N

IMS-SPONSOR

Number of attributes : 2

Key attributes : SCN^{FK}, SSN^{FK}

Foreign keys : SCN (references IMS),

SSN (references SPONSOR),

Relationships with : IMS 0:N

SPONSOR 0:N

IMS DEPENDENT

Number of attributes : 10

Key attributes : LAST NAME, FIRST NAME, SCN^{FK}

Foreign key : SCN (references IMS)

Relationship with : IMS 1:1

SPONSOR DEPENDENT

Number of attributes : 6

Key attributes : LAST NAME, FIRST NAME, SPONSOR SSN^{FK}

Foreign key : SPONSOR SSN (references SPONSOR)

Relationship with : SPONSOR 1:1

IP ACTIVITY

Number of attributes : 10

Key attributes : ACTIVITY NAME, START DATE

Foreign key : None

Relationships with : ESCORT PARTICIPATION 0:N

IMS PARTICIPATION 0:N

SERVICES USED 0:N

ESCORT PARTICIPATION

Number of attributes : 6

Key attributes : SSN^{FK}, ACTIVITY $NAME^{FK}$, START $DATE^{FK}$

Foreign keys : SSN (references SPONSOR),

ACTIVITY NAME, START DATE (references IP ACTIVITY)

Relationships with : IP ACTIVITY 1:1

SPONSOR 1:1

IMS PARTICIPATION

Number of attributes : 6

Key attributes : SCN^{FK}, ACTIVITY $NAME^{FK}$, START $DATE^{FK}$

Foreign keys : SCN (references IMS),

ACTIVITY NAME, START DATE (references IP ACTIVITY)

Relationships with : IP ACTIVITY 1:1

IMS 1:1

SERVICES USED

Number of attributes : 7

Key attributes : SERVICE TYPE, SERVICE DATE, BUSINESS

NAME[FK], ACTIVITY NAME[FK], START DATE[FK]

Foreign keys : BUSINESS NAME (references VENDOR), ACTIVITY NAME, START DATE (references IP ACTIVITY)

Relationships with : IP ACTIVITY 1:1

VENDOR 1:1

VENDOR

Number of attributes : 12

Key attribute : BUSINESS NAME

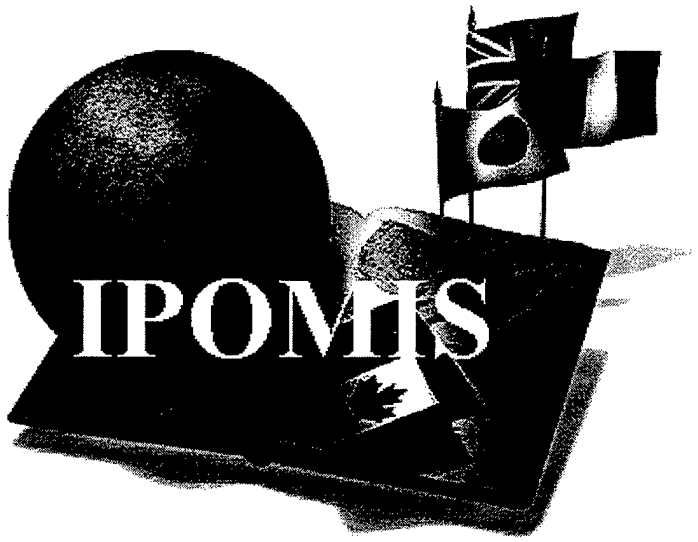
Foreign key : None

Relationship with : SERVICES USED 0:N

APPENDIX D. USER'S MANUAL

IPOMIS

INTERNATIONAL PROGRAMS OFFICE
MANAGEMENT INFORMATION SYSTEM



USER'S MANUAL

Önder Çelebi
LTJG, Turkish Navy
Naval Postgraduate School

CONTENTS

0

INTRODUCTION

Welcome to the Naval Postgraduate School (NPS) International Programs Office Management Information System (IPOMIS). IPOMIS was developed to support the IPO's administrative activities more efficiently by providing timely and accurate information about International Military Students (IMS) attending NPS.

IPOMIS was designed based on the information requirements identified during the extensive interviews with the IPO staff. Prototypes of the system were iteratively developed and demonstrated to ensure that the office end-users were fully satisfied with the final system specifications. The primary objective was to develop a system that was very user friendly without spending any office resources.

USING THIS MANUAL

You can use this manual to install, configure and use the IPOMIS. This manual is organized as chapters to provide detailed information on all the features of the system.

- **Chapter 1 - Getting Started** contains all the information about installing and configuring IPOMIS, and the common features of the IPOMIS applications.
- **Chapter 2 - IMS Applications** contains detail information about IPOMIS's IMS Information and IMS Information Viewer applications.

- **Chapter 3 - Sponsor Applications** contains detail information about IPOMIS's Sponsor Program Manager and Sponsor Information Viewer applications.
- **Chapter 4 - Information Program Manager** provides detailed information about IPOMIS's Information Program Manager Application.
- **Chapter 5 - System Utilities** introduces the InterBase Server Manager, Windows Interactive SQL, and IDAPI configuration utility.
- **Appendix - Notes to System Developer/Maintainer** provides important notes to the system programmer

SYSTEM OVERVIEW

Basically IPOMIS includes a relational database, a relational database management system with database administration utilities, and three main database applications (see Figure 1).

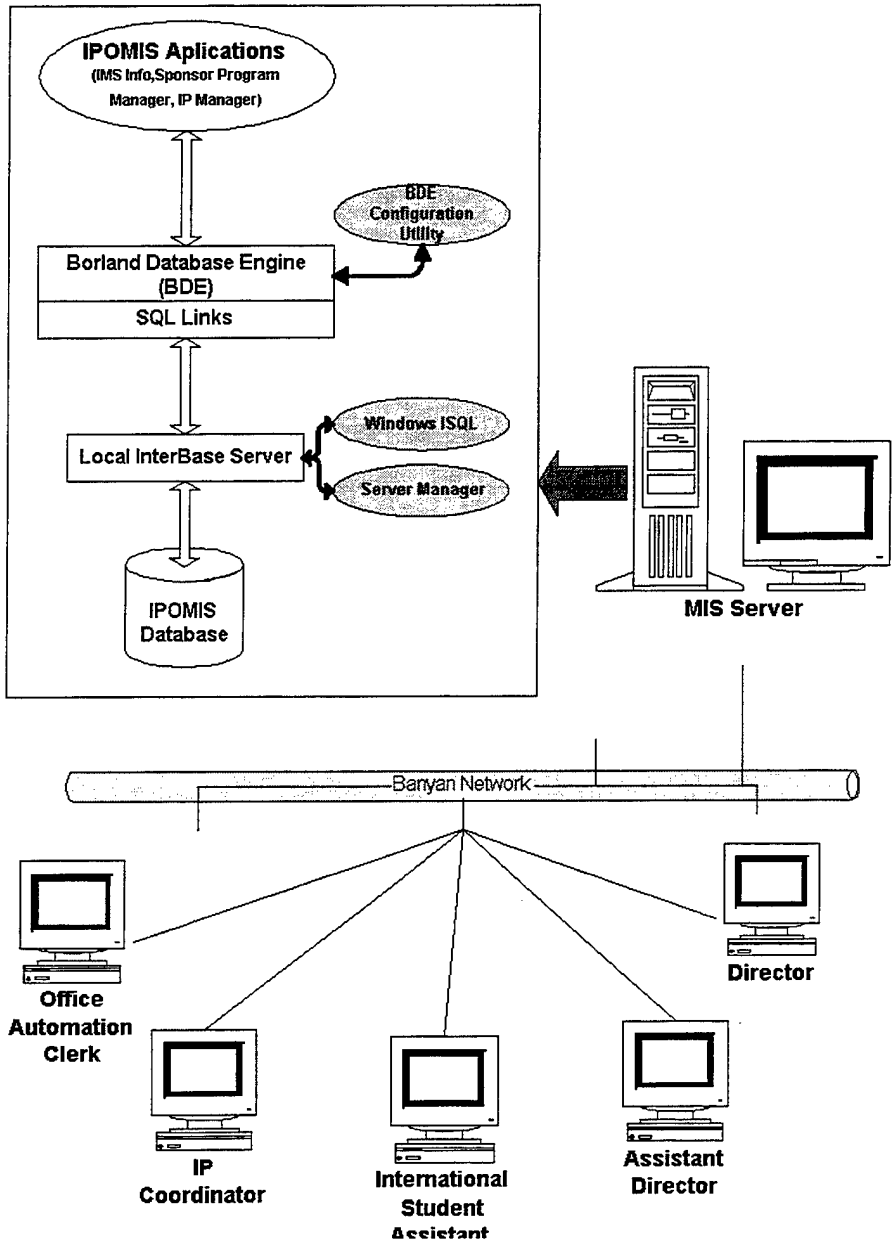


Figure 1 IPOMIS System Layout

1. IPOMIS Database and Utilities

Borland's Local InterBase® Server, an SQL-compliant relational database management system (DBMS), is utilized to develop system's database.

The Local InterBase Server includes database utilities supporting IPOMIS database administration:

- **Windows ISQL**, an interactive data definition and query tool for Windows;
- **Server Manager**, a Windows tool for database backup, restoration, maintenance, and security.

IPOMIS applications access system's database through the Borland Database Engine (BDE) / IDAPI. BDE is a Borland's shared software component that provides a common database layer for database applications and DBMS.

For further information about IPOMIS database refer to Onder Celebi's Master Thesis.

2. IPOMIS Applications

IPOMIS consists of three Microsoft Windows-based applications to support the IPO's administrative activities:

a. IMS Information and IMS Information Viewer

IMS Information application helps user, specifically International Student Assistant to perform most of his/her tasks associated with the IPO's administrative activities. It provides a user friendly interface to access the IPOMIS database where the user can maintain IMS, IMS_DEPENDENT, COUNTRY, CURRICULUM, ALUMNUS tables and generate various structured IMS and sponsor rosters and reports.

IMS Information Viewer is the read-only (i.e., users can only view the information, but change) version of IMS Information. Viewer allows the users other than International Student Assistant to review information related to IMS. Viewer does not support reporting features of IMS Information.

b. Sponsor Program Manager and Sponsor Information Viewer

This application is designed to assist International Student Assistant while performing tasks associated with the Sponsor Program. It allows the user to select and assign the appropriate sponsor to an IMS, maintain SPONSOR and SPONSOR_DEPENDENT tables, and prepare reports.

Sponsor Information Viewer is the read-only version of Sponsor Program Manager application. Viewer allows the other users to review information related to sponsors.

c. Information Program Manager

Information Program Manager is designed to assist Information Program Coordinator to perform his/her administrative tasks associated with managing and coordinating the IP activities, such as preparing selection list and activity report, reviewing and approving applications, etc.

1

GETTING STARTED

This chapter contains all the information about installing and configuring IPOMIS, and the common features of the IPOMIS applications.

1. System Requirements

IPOMIS requires the hardware and software described in Table 1.1

Component	Description / Comments
Microprocessor	80486 or higher
RAM	8MB. (16MB is strongly recommended) Performance will increase with memory
Hard Disk	At least 30 MB. of free disk space to use IPOMIS effectively
Video Monitor	VGA or higher with: resolution 800x600-256 colors-Small Fonts
Mouse	A mouse or another pointing device is required. Most of the system features can only be accessed with mouse.
Operating System	Microsoft Windows 3.x , 95 or NT
System Software	Borland's Local InterBase Server (as DBMS) Borland Database Engine (BDE) / IDAPI Database Utilities • Windows ISQL, • Server Manager. System Client Applications • IMS Information, • Sponsor Program Manager, • Information Program Manager, • IMS Information Viewer, • Sponsor Information Viewer.
Network	IPOMIS Client applications was designed to access system's database through Local InterBase Server (DBMS) and Borland Database Engine (BDE). System's database and DBMS should be networked to allow multiple users to be able to access database.

Table 1.1 System Requirements for IPOMIS

2. Installing IPOMIS

Follow the steps below one by one to complete installation of the system:

1. Create following directories on the local drive of the network server; in this case assume the local drive is "C:\".

- C:\IPOMIS : This directory will be the root directory for the system files.
- C:\IPOMIS\APPS : directory for client applications.
- C:\IPOMIS\APPS\DATA directory for temporary data files created by applications (e.g., select.txt and report.txt files created by Information Program Manager)
- C:\IPOMIS\DBASE : directory for system's database file, "IPOMIS.GDB".
- C:\IPOMIS\IDAPI : directory for IDAPI / BDE program and configuration files.
- C:\IPOMIS\IBLOCAL : directory for Local InterBase Server files including Server Manager and Window ISQL.

2. Copy applications from floppy diskettes to "C:\IPOMIS\APPS" directory. Contents of the floppy diskettes are listed in Table 1.2.

Label	Content
IMS Information	IMSInfo.exe
IMS & Sponsor Information Viewers	IMSView.exe and SponView.exe
Information Program Manager	IpM.exe
Sponsor Program Manager	Sponsor.exe

Table 1.2 Contents of the Application Diskettes

3. Copy database file "IPOMIS.GDB" to "C:\IPOMIS\DBASE" directory.

1. **Install IDAPI / BDE:**

- Insert installation diskette labeled "Borland Database Engine (IDAPI) Disk 1" into floppy drive,

- Using your favorite method of running a Windows program, run the SETUP.EXE program located on the installation diskette,
- Follow the installation instructions appearing on the screen. You will be prompted to define your preferences on the locations of program files and configuration file on the "Borland Database Engine Locations Settings" screen (see Figure 1.1). Your answer should be "C:\IPOMIS\IDAPI" for both locations.

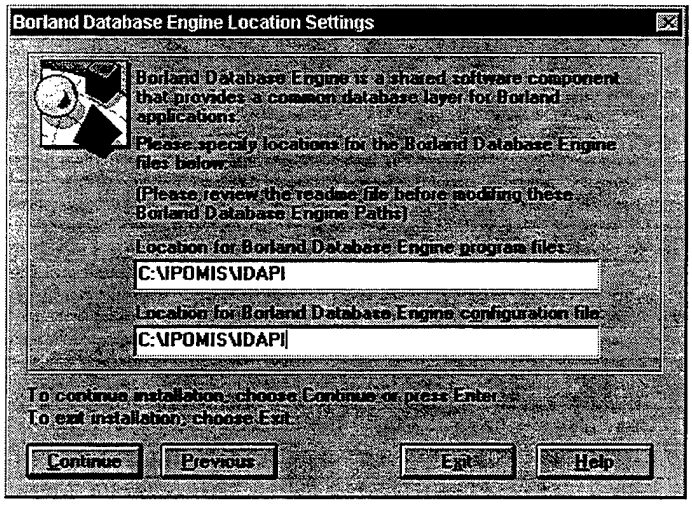


Figure 1.1 BDE Installation Locations Settings Dialog Box

5. **Install Local InterBase Server**

- Insert installation diskette labeled "Local InterBase Server Disk 1" into floppy drive,
- Using your favorite method of running a Windows program, run the SETUP.EXE program located on the installation diskette,
- Follow the installation instructions appearing on the screen. You will be prompted to define your preferences on the locations of program files on the "Customize

Installation" screen (see Figure 1.2). Your answer should be "C:\IPOMIS\IBLOCAL".

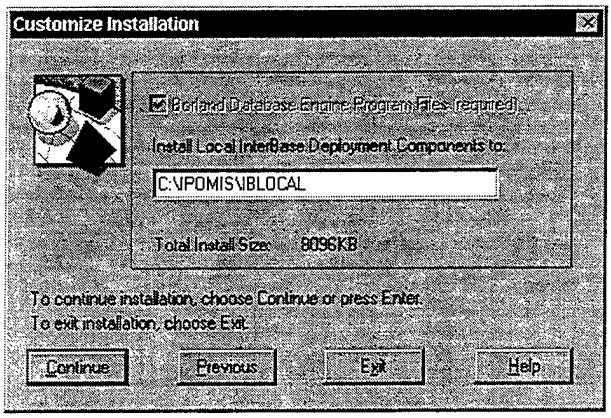


Figure 1.2 Local InterBase Server Customize Installation Dialog Box

3. Configuring IPOMIS

On completion of successful installation, system must be configured before running the client applications.

a. Configuring IDAPI / BDE

The Borland Database Engine configuration utility (BDECFG.EXE) enables you to configure system BDE alias (IPOMISDB) and change the settings reflecting IPOMIS database environment in the BDE configuration file, IDAPI.CFG. For further information on BDE configuration see Chapter 5 Part 3 "Using BDE Configuration Utility".

To run the BDE Configuration Utility, double-click the BDE configuration utility icon in  the IPOMIS program group. The BDE Configuration Utility opens:

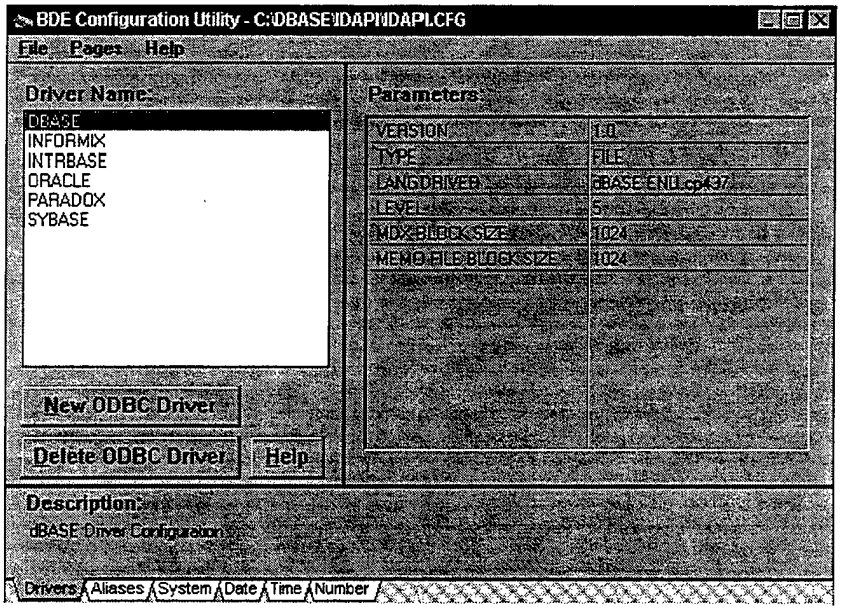


Figure 1.3 BDE Configuration Utility main window

Important Note: IPOMIS client applications use alias named IPOMISDB which points system's database, therefore any alias pointing system's database must be named specifically IPOMISDB.

1. Select the Alias Manager (Aliases page) and choose the New Alias button. The Add New Alias dialog box appears (see Figure 1.4) . The type can be STANDARD or SQL-specific. For IPOMIS type must be INTRBASE.

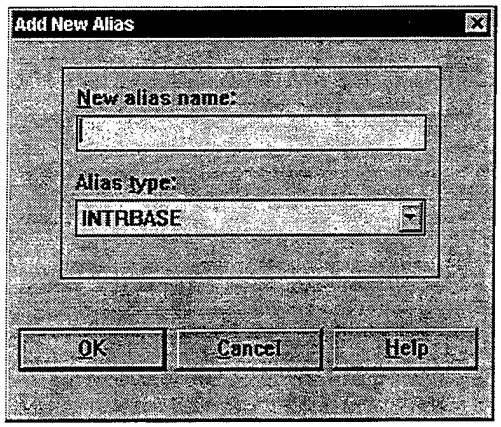


Figure 1.4 Add New Alias Dialog Box

2. Enter the new alias name IPOMISDB and select the SQL-specific alias type INTRBASE. Then choose OK to begin the setup process. The Alias Manager displays all the configuration parameters you can change to customize the new alias.

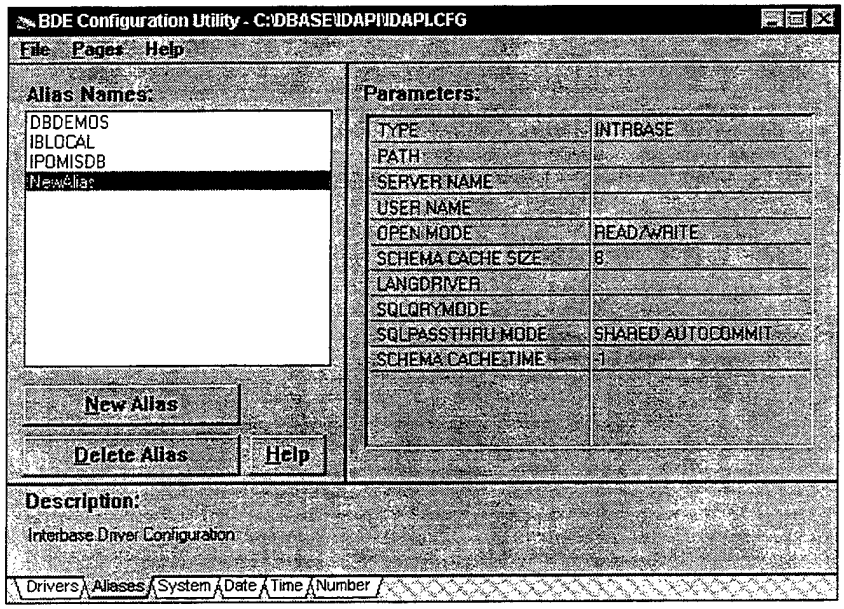


Figure 1.5 Customizing the new alias

3. Edit the settings for the category you selected. Settings must have the values described in the Table 1.5.

CATEGORY	SETTING / COMMENTS
TYPE	INTRBASE
PATH	Blank. Not Available
SERVER NAME	C:\IPOMIS\DBASE\IPOMIS.GDB
USER NAME	Blank.
OPEN MODE	READ/WRITE
SCHEMA CACHE SIZE	8
LANGDRIVER	Blank.
SQLQRYMODE	Blank.
SQLPASSTHRU MODE	SHARED AUTOCOMMIT
SCHEMA CACHE TIME	-1

Table 1.5 IPOMISDB Alias Settings

4. When you are finished, select File|Save to save the new alias in the default configuration file.

Note to Network Administrator: IDAPI directory and IDAPI.CFG file must be read/write accessible by System Database Administrator. The other users should have at least read only access.

b. Configuring user workstations

When the IPOMIS applications and utilities are started, they look for the IDAPI files in the location they read from the WIN.INI file, so those settings must be incorporated into the WIN.INI files of all users running the applications and utilities.

WIN.INI contains a section used by IDAPI. This section has the location of the IDAPI files and the IDAPI configuration file:

- DLLPATH points the location of your IDAPI files.
- CONFIGFILE01 specifies the location and filename of the IDAPI configuration file (Default is IDAPI.CFG).

If you move the IDAPI directory or change its name you will need to modify this settings. The settings must include the drive and full path name to the IDAPI directory. A typical WIN.INI IDAPI section can be as follows:

[IDAPI]

DLLPATH=N:\IPOMIS\IDAPI

CONFIGFILE01=N:\IPOMIS\IDAPI\IDAPI.CFG (N: drive indicates the mapped network drive)

4. Common Features of the IPOMIS Applications

Before getting started to use the IPOMIS applications, it will be very useful to review common features of the applications.

a. How to Login the Database

Because of security concerns, data stored in the system's database must be protected. Therefore, IPOMIS utilizes a secure database. Whenever you attempt to use one of the applications, the first thing after the splash screen appears on the screen will be the database login dialog:

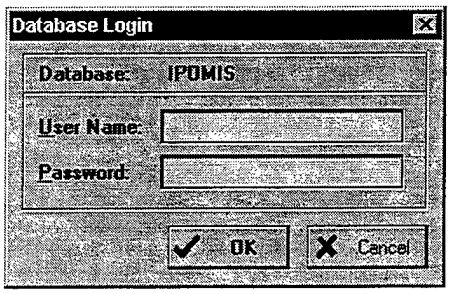


Figure 1.6 Database Login Dialog

As you can see on Figure 1.6, there are two fields you must fill out with the proper user name and password. To get a user name and password consult with the IPOMIS system administrator.

- Type your user name assigned by the system administrator in either uppercase or lowercase.

- You can click to password field or press the Tab key to activate the password field. Type your password. Password field is case sensitive, thus you must type exactly the same password of yours.
- Click OK. If you click Cancel then program will not attempt to connect to the database and terminates itself.

If you type wrong user name or password then program will assume that you are an unauthorized user and terminates itself:

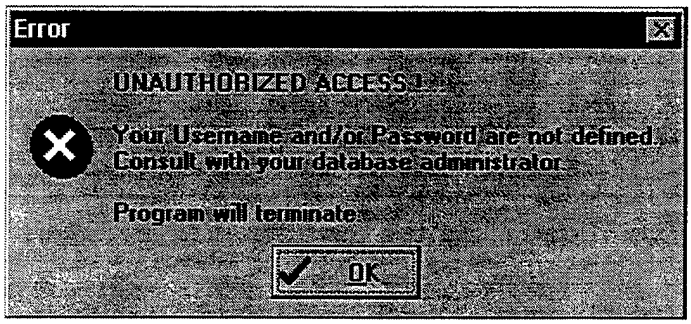
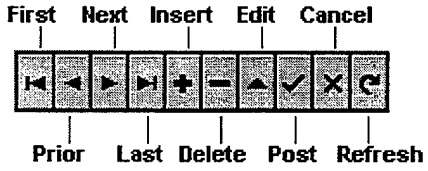


Figure 1.7 Unauthorized Access Error Message

b. How to Use Database Navigator

Database navigator allows you to move through the data extracted from the database, and perform operations on the data, such as inserting a blank record or posting a record.

The database navigator consists of multiple buttons. On most of the IPOMIS application screens you may not see all of them, because their functions are performed by other buttons or menu items. Entire set of buttons are:



When you choose one of the navigator buttons, the appropriate action occurs on the database portion (dataset) the navigator is linked to.

This table describes the buttons on the navigator:

Button	Purpose
First	Sets the current record to the first record in the dataset, disables the First and Prior buttons, and enables the Next and last buttons if they are disabled
Prior	Sets the current record to the previous record and enables the Last and Next buttons if they are disabled
Next	Sets the current record to the next record and enables the First and Prior buttons if they are disabled
Last	Sets the current record to the last record in the dataset, disables the Last and Next buttons, and enables the First and Prior buttons if they are disabled
Insert	Inserts a new record before the current record, and sets the dataset into Insert and Edit states
Delete	Deletes the current record and makes the next record the current record
Edit	Puts the dataset into Edit state so that the current record can be modified
Post	Writes changes in the current record to the database
Cancel	Cancels edits to the current record, restores the record display to its condition prior to editing, and turns off Insert and Edit states if they are active
Refresh	Redisplays the current record from the dataset, thereby updating the display of the record on the form

Table 1.6 Database Navigator Buttons

c. How to Use Pick-a-date Calendar

For your convenience, almost every date field can be filled out utilizing the Pick-a-date calendar.

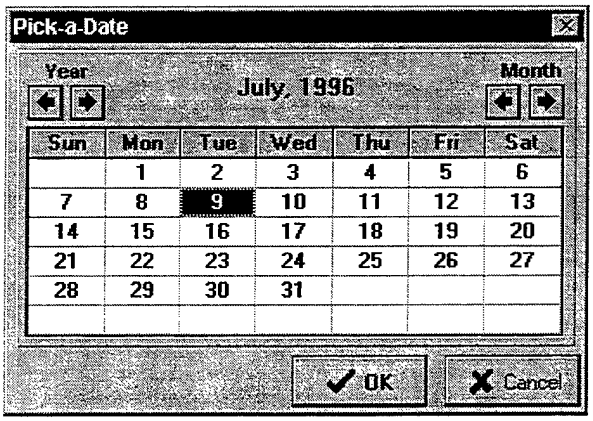


Figure 1.8 Pick-a-date Calendar

- Click  button next to the field you want fill out with a date value. Then the calendar window will open:
- By default the window displays the current date. Use arrow buttons to select next or previous month or year. you can see the month and year selected on the top of the calendar display.
- Click the day on the calendar display.
- Click OK to pick the date or click Cancel to cancel operation. If you choose to cancel operation date field will be left blank or unchanged.

d. How to Use Report Viewer

IPOMIS reports or lists are displayed on the report viewer. Report viewer enables you to:

- Preview the reports or lists with different scales, such as 50%,150%, Fit in window, etc.
- Print the reports or lists
- Save and reload the reports and lists

Whenever you click to review a report or list, the application will generate the report and displays on the report viewer:

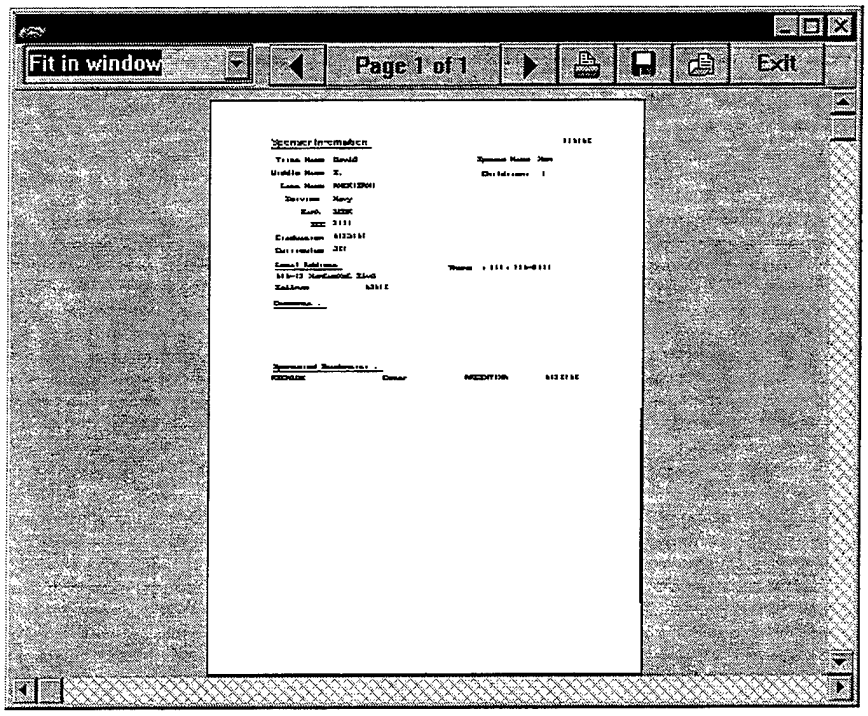


Figure 1.9 Report Viewer

- Click on the report display to toggle between Fit in window and 100% scales or choose an appropriate scale from the drop down list on the left upper corner of the window.
- If the displayed item is more than one page then the page counter will display in the form of " Page Y of X". Click left arrow to preview the next page or click right arrow to preview previous page.
- Click Print Report button  to print out the item.
- Click Save Report button  to save the item.
- Click Load Report button  to load a pre-saved report from disk drive.
- Click Exit button to exit the viewer.

2

IMS Applications

This chapter provides detailed information about IPOMIS's IMS Information and IMS Information Viewer Applications.

IMS Information application (IMS Info) is designed to assist International Student Assistant to perform his/her administrative tasks associated with managing the IMS data.

IMS Information Viewer is designed to enable the office staff to access information about IMSs.

Part 1

IMS Information

IMS Information application enables International Student Assistant to:

- Manage database portions associated with the IMSs:
    - Add/Delete IMSs
    - Add/Delete IMSs' dependents
    - Add/Delete Alumni, Countries, Curriculums, etc.
- Generate IPOMIS reports and lists

1. Starting IMS Info and Main Window

Start IMS Info by clicking the program icon  in the IPOMIS program group. Database Login window will then open and prompt you to enter your User Name and Password (see Chapter 1 "Common Features of the IPOMIS Applications - How to Login Database").

Important Note: Because of security concerns, IMS Info is designed to be used by International Student Assistant. If you attempt to enter as another user then program will terminate and you will not be able use the program. For further information see Chapter 5 "Administrating Security" and consult with your Database Administrator.

If you provide proper user name and password then program login you to the system's database and the IMS Info main window will then open:

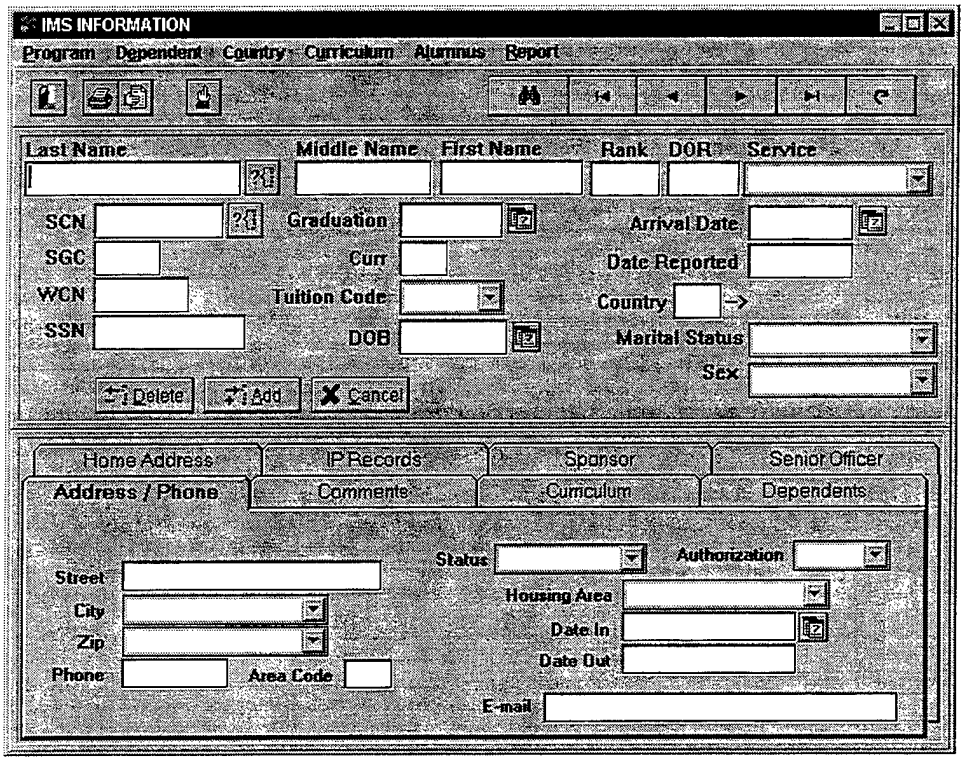


Figure 2.1 IMS Information Main Window

This window consists of the:

- Menu bar, across the top of the window, containing commands you can choose to perform application tasks.
- Speedbar, consists shortcut speed buttons and a row of database navigate buttons to navigate on IMS table, just below the menu bar.

- Information window, displayed below the Speedbar, showing IMS currently selected and Delete/Add/Cancel buttons to maintain IMS data.
- Tabbed notebook, displayed below the Information window. This area displays detail information about the selected IMS, organized as tabs.

a. IMS Information Menus

The menus are the basic way to perform tasks with the IMS Info. There are six menu items and three of them are pull-down menus:

- Program menu: enables you to exit from application, set the printer and see about screen.
- Dependent menu: enables you to add dependents to the selected IMS and edit the dependents.
- Country menu item: enables you to access IMS's country information displayed on the country information window.
- Curriculum menu item: enables you to access the database curriculum information displayed on the curriculum information window.
- Alumnus menu enables you to:
  - Edit Alumnus
  - Transfer selected IMS data to Alumnus table
  - Transfer all graduated or graduating IMSs data to Alumnus table
- Report menu item: enables you to access the report center where you can select the pre-formatted rosters or lists.

Each menu item can be activated clicking them or pressing both Alt button and the underlined letter in the menu item. For example: Program menu item will be activated pressing **Alt+P**. Most pull-down menu items indicate shortcuts (Hot Keys) for each task, such as "Dependent | Add Ctrl+A" indicates that if **Ctrl+A** is pressed then program invokes adding dependent process.

b. Speedbar

Consist of four speed buttons:

 for exiting from the application

 for setting up the printer

 for previewing IMS's information sheet

 for transferring IMS data to Alumnus table

and a row of database navigate buttons to navigate on IMS table (see Chapter 1 "Common Features of the IPOMIS Applications - How to Use Database Navigator").

c. Information Window

This window consists of two parts

- IMS information: where you can see and edit basic IMS data. You can navigate between edit fields pressing Tab / Shift+Tab key or clicking the field you want to edit.
- Task buttons: enable you to perform the database tasks, such as adding new IMS, deleting selected IMS or cancel/undo latest editing.

d. Tabbed Notebook

This tabbed notebook consists of seven tabs:

- Address/Phone : enables you to review or edit selected IMS's address and phone data.
- Comments: enables you to review or edit comments about the selected IMS.
- Curriculum: enables you to review detail information about the curriculum enrolled by the selected IMS.
- Dependent(s): enables you to review information about IMS's dependents.
- Home Address: enables you to review or edit the selected IMS's address in his/her home country.

- IP Records: enables you to review IMS's DLI Attendance status.
- Sponsor(s): enables you to review selected IMS's sponsor(s).
- Senior Officer: enables you to review senior officer of the country where the selected IMS is from.

2. Locating IMS

Before performing any task with the application (except working on the curriculum data or Report tasks) you must select the IMS you want to work on. In order to assist you locating an activity, application has "Search by Last Name" and "Search by SCN" features.

To locate an IMS:

- You can locate an IMS anytime searching by his/her last name or SCN.
  - *To search by last name:* click one of "Search By Last Name" buttons (left of navigator  on the speedbar or next to Last Name field ).

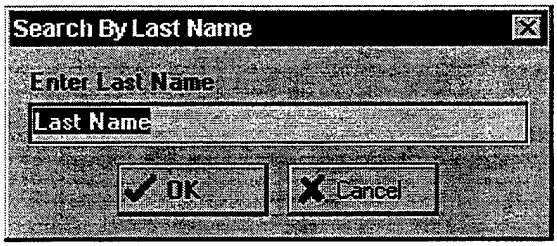


Figure 2.2 Search by Last Name Dialog.

- Enter the last name in either uppercase or lowercase. You do not have to type full name of the IMS. Program will try to locate nearest match.
  - *To search by SCN:* click "Search By SCN" button  next to the SCN field.
- Enter the SCN. You do not have to type full SCN. Program will try to locate nearest match.
- Click OK to search or Cancel to abort search.

Note: After search, be sure that the IMS shown on the Information window is the IMS that you want to work on. If you searched for the nearest match and you could not locate the IMS then use navigate buttons to continue search.

3. Adding New IMS

To add a new IMS:

Click "Add" button or press **Alt+A**. Then all edit fields on the screen will be blank and cursor will jump to the Last Name field to enable you to start editing. You may use **Tab** to jump next or **Shift+Tab** to previous field.

- In order to fill most of the date fields, such as ***Arrival Date, DOB, Graduation,*** etc. fields you may make use of a calendar. Press  button to pick a date from calendar (see Chapter 1 "Common Features of the IPOMIS Applications - How to Use Pick-a-date Calendar").
- Program will generate a temporary ***SCN*** for the new IMS. If you do not know his/her actual SCN use this number. It is preferred to provide actual SCNs to the program.

  Important Note: DO NOT change SCN after assignment or adding dependents. If you attempt to change IMS's SCN after a sponsor is assigned or a dependent is added, program will generate database errors. Because an IMS is uniquely identified with the SCN in the database, every association with the IMS is created using the IMS's SCN. Changing causes to interrupt these associations and database never allows you to interrupt the associations once they are formed.

  If you really need to change an IMS's SCN:

  - Tell the IP Coordinator to note the IMS's IP participation information somewhere and delete it. After you changed IMS's SCN, IP coordinator can reenter the deleted information.
  - Delete IMS's dependents. Before deleting, note the information about dependents somewhere.
  - Delete sponsor's assignment association with the IMS.

- After deleting, reenter dependent information and reassign the sponsor(s).

- After editing fields click  on the navigator to refresh the database and actually post your new IMS record to the database. Before refreshing/posting, remember almost each field on the screen must have been filled in order to add a new IMS successfully. Otherwise program will warn you and error message will appear on the screen indicating that blank field must have a value.
- Before refreshing your data, you can Cancel/Undo the editing you made: click "Cancel" button or press **Alt+C**.

Now you are ready to add dependents, assign sponsor, etc. to your new IMS.

4. Deleting IMS

To delete an IMS:

Locate the IMS you want to delete and Click "Delete" button or press **Alt+D**. You will be prompted to confirm delete. If you choose Yes then delete process will initiate and all dependents, IP activities participated, and sponsors associated with the selected IMS will be deleted. If you choose No the information about the IMS will not be deleted.

> ***Note:*** Before deleting an IMS you must be sure about it. Because when an IMS is deleted there is no way to revitalize the data associated with the IMS.

5. Adding New Dependent

To add a new dependent:

Press **Ctrl+D** or choose "Dependent | Add". IMS's Dependent(s) Information window will then open:

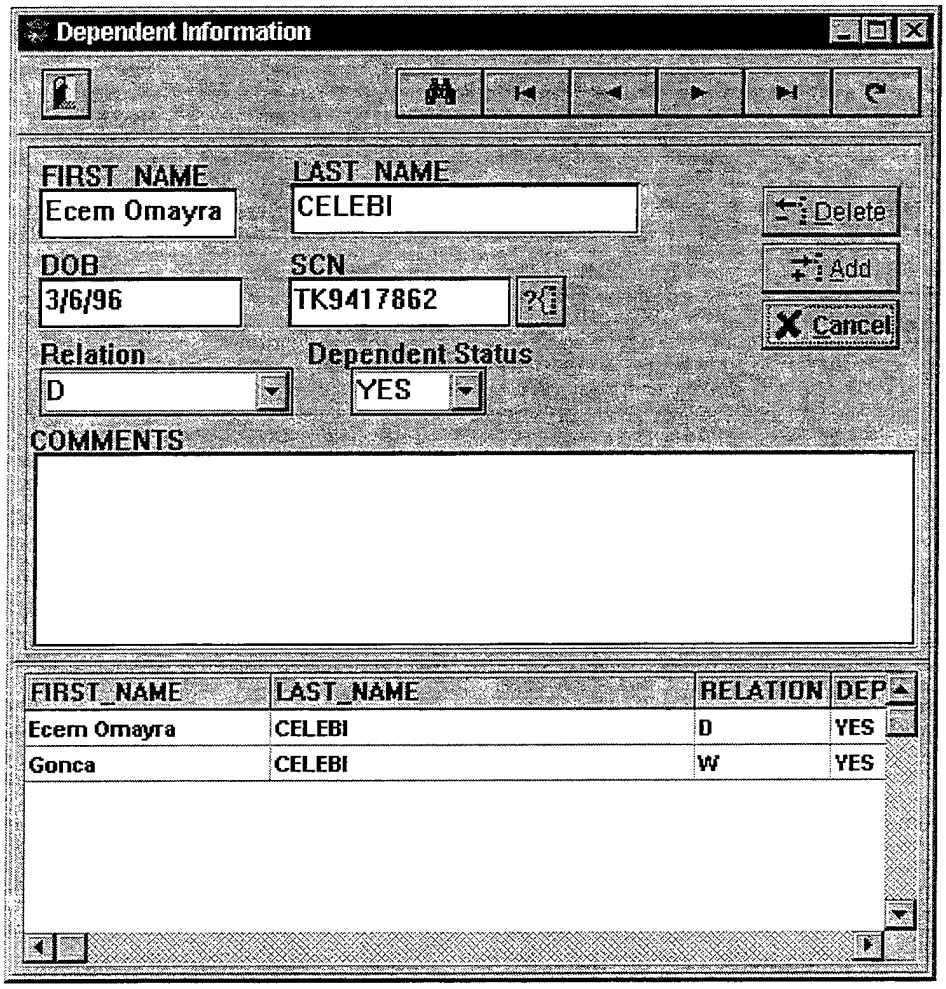


Figure 2.3 IMS's Dependent(s) Information Window

- Program will prompt you to enter: ***First Name:*** type first name of the new dependent with uppercase initial. ***Dependent Status:*** by default it is YES, type NO if the dependent is not in the US with the IMS. ***Relation:*** type W for wife, H for husband, D for daughter, S for son or O for other dependents: cousin, nephew, etc. Choose OK on each dialog boxes. If you choose Cancel then the add dependent process will be canceled.
- By default, program will fill the ***Last Name*** and ***SCN*** fields with the IMS's. You can edit Last Name field anytime clicking it, but you cannot edit SCN field. It is a read-only field.

- ***Comments*** field is a plain text field used to enter detail information about the dependent.
- Click  on the navigator to refresh the database and actually post your new dependent record to the database. Otherwise program will warn you and error message will appear on the screen indicating that blank field must have a value.
- Before refreshing your data, you can Cancel/Undo the editing you made: click "Cancel" button or press **Alt+C**.

6. Editing a Dependent

To edit dependent data:

Press **Ctrl+E** or choose "Dependent | Edit". If the IMS has a dependent "IMS's Dependent(s) Information" window will then open (see Figure 2.3) and dependent data will be displayed. If the IMS has no dependent then program will remind you that.

- Click on the field you want to edit.
- You can use pull down menu to select relation code from the list. It is recommended.
-

7. Adding/Deleting/Editing Country and Senior Officer

Country table includes information about countries sending students to NPS. But it does not include all countries in the world. If a student attends to NPS from a country which has not been recorded to the database, you must add the new country to the database. If you will not add the country to the database you will not be able to create senior officer information and you will not be able to see country information on the rosters/lists or reports.

- Click Country menu item or press "**Alt+O**". Country Information window will then open:

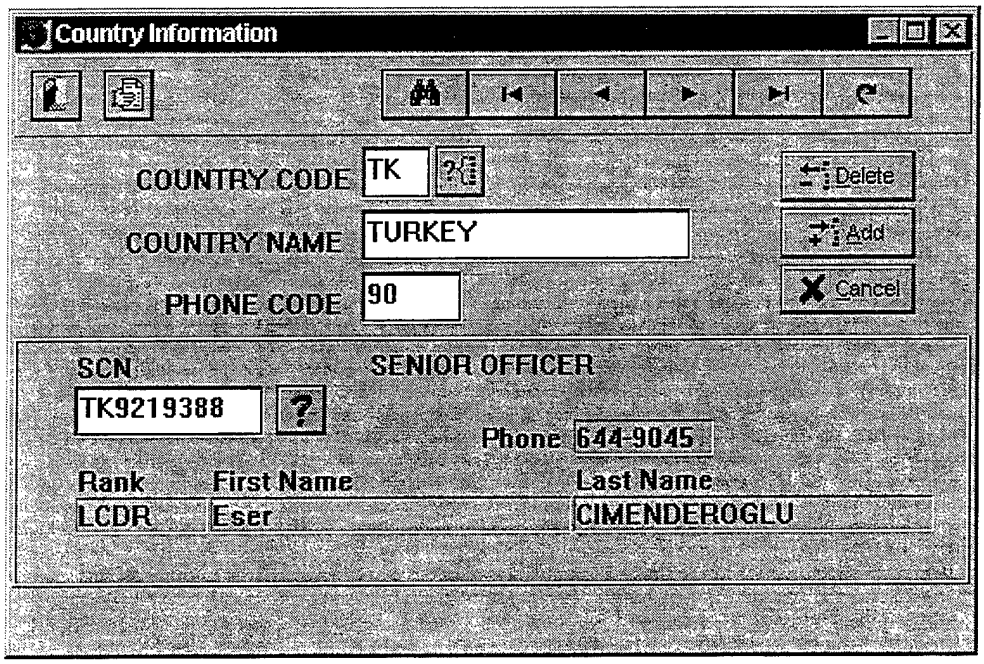


Figure 2.4 Country Information Window

To add a new country:

- Click Add button or press "**Alt+A**" to initiate the adding process.
- Program will clear fields and moves the cursor to the first field Country code.
- After editing fields on the window, click on the navigator to refresh the database and actually post your new country record to the database. Otherwise program will warn you and error message will appear on the screen indicating that blank field must have a value.
- Before refreshing your data, you can Cancel/Undo the editing you made: click "Cancel" button or press **Alt+C**.

To delete a country:

- Click Delete button or press "**Alt+D**" to initiate the Deleting process.
- Program will clear the information about the country. This process will not affect the IMS data from the deleted country.

To edit a country:

- Locate the country you want to edit. Click one of the "Search By Country Code" buttons (left of navigator  or next to the Country Code field ). Then type the code to the dialog box appeared on the screen.
- Click the field you want to edit. If you change the country code you must update all IMSs' country codes one by one. Otherwise you will not be able to access country information of those IMSs.

To edit senior officer data:

- Locate the country you want to edit its senior officer. Click "Show all students from this country" button  next to the senior officer SCN field to see the list of all students from the selected country.
- Double click on the name of the student on the list. Program will assign/record the selected student as the senior officer of the selected country.
- Or: If you know the SCN of the student you want to record as the senior officer type the SCN into the SCN field.

8. Adding/Deleting/Editing Curriculum

Curriculum table includes information about curricula in NPS. All the curricula information in NPS is in the database. To maintain this information – i.e., update, edit, etc.:

- Click Curriculum item or press "**Alt+U**". Curriculum Information window will then open:

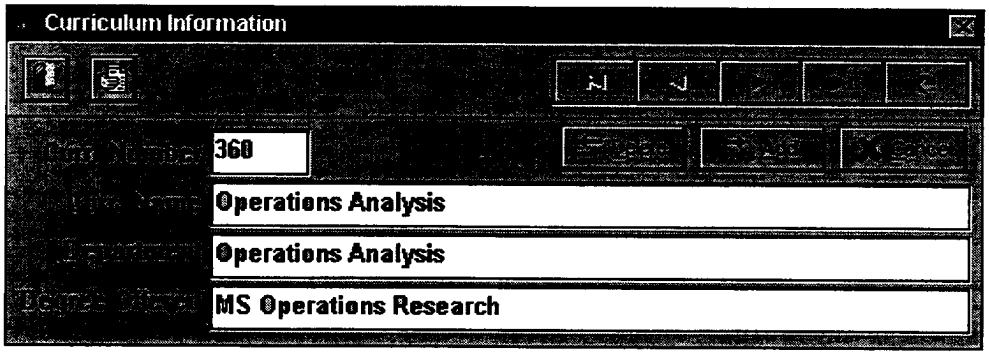


Figure 2.5 Curriculum Information Window

To add a new curriculum:

- Click Add button or press "**Alt+A**" to initiate the adding process.
- Program will clear fields and moves the cursor to the first field Curriculum Number.
- After editing fields on the window, click on the navigator to refresh the database and actually post your new curriculum record to the database. Otherwise program will warn you and error message will appear on the screen indicating that blank field must have a value.
- Before refreshing your data, you can Cancel/Undo the editing you made: click "Cancel" button or press **Alt+C**.

To delete a curriculum:

- Click Delete button or press "**Alt+D**" to initiate the Deleting process.
- Program will clear the information about the curriculum. This process will not affect the IMS data attending the deleted curriculum.

To edit a curriculum:

- Locate the curriculum you want to edit. Use the navigator to locate.
- Click the field you want to edit. If you change the curriculum number you will not be able to access curriculum information of those IMSs attending the curriculum.

9. Transferring IMS to Alumnus table

Alumnus table includes information about the graduated IMSs. When an IMS graduates his/her record must be transferred to the alumnus table. Otherwise IPOMIS applications will assume that the IMS did not graduate and keep the associations, such as IP activities, details of the dependents and sponsors. This will cause erroneous reports and lists, such as statistics report.

To transfer the selected IMS to Alumnus table:

- Click "Transfer IMS to Alumnus" button  or choose "Alumnus | Transfer IMS" menu item.
- Program will prompt you to confirm transfer. If you click OK on the dialog box then the transfer process will be initiated. If you click Cancel transfer will be terminated.
- After the transfer is completed program will inform you that the transfer is completed.

Note: Before transferring graduating or graduated IMS you must be sure about it. Because when you select OK on the Confirm Dialog, some information about the IMS is deleted. There is no way to revitalize them again.

To transfer All graduated IMSs to Alumnus table:

- Choose "Alumnus | Transfer All Graduates" menu item. A dialog box prompting you to enter or confirm the graduation date will then appear on the screen:

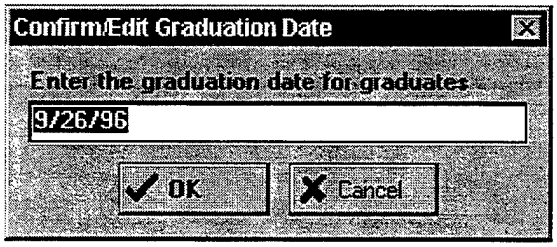


Figure 2.6 Confirm/Edit Graduation Date Dialog

- If the graduation date on the screen is correct for you then click OK. If not then type the graduation date of the IMSs you want to transfer. You can cancel this operation clicking Cancel instead of OK. On completion of the process, program will inform you that all graduates are transferred.

10. Adding/Deleting/Editing Alumni

- Choose "Alumnus" menu item or press "**Alt+L**" then Alumnus Information window will open:

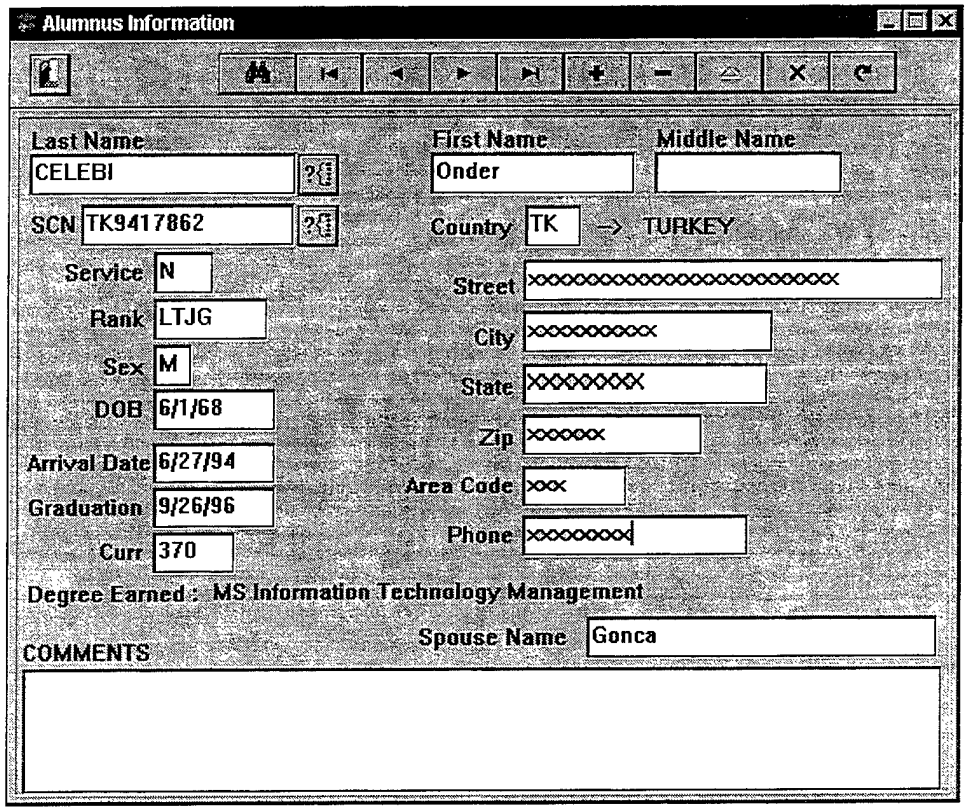


Figure 2.7 Alumnus Information Window

To add an Alumnus:

- Click add button on the navigator.
- Fill out the fields.
- Click refresh button on the navigator.

To delete an Alumnus:

- Click delete button on the navigator.
- Click OK on the confirmation dialog

To edit Alumnus data:

- Locate the alumnus you want to edit. To locate an Alumnus:
  - You can locate an Alumnus anytime searching by his/her last name or SCN.
    - *To search by last name:* click one of "Search By Last Name" buttons (left of navigator  on the speedbar or next to Last Name field ).
  - Enter the last name in either uppercase or lowercase on the search dialog box (see Figure 2.2). You do not have to type full name of the Alumnus. Program will try to locate nearest match.
    - *To search by SCN:* click "Search By SCN" button  next to the SCN field.
  - Enter the SCN. You do not have to type full SCN. Program will try to locate nearest match.
  - Click OK to search or Cancel to abort search.
- Click edit button on the navigator.
- Click the field you want to edit.
- Click refresh button on the navigator after editing.

11. Preparing Reports and lists

IMS Information application consist a variety of pre-formatted reports and lists. You can review, print and save/load these reports. In order to access report features click on the Report menu item or press "**Alt+R**". The Report Center window will then open:

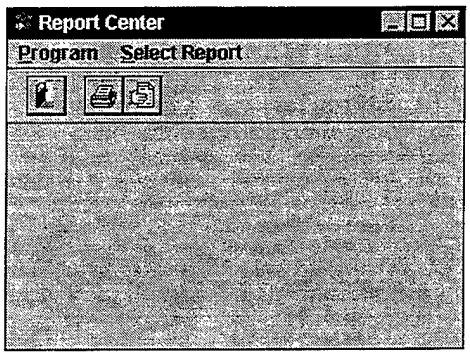


Figure 2.8 Report Center Window

Report center window allows you to select the report to be reviewed. Choose XXXX menu item. Make your selection from the drop-down list. Some of the item will directly initiates the report preparing process, some of them will require further selection. If the item requires further selection then a drop-down list which allows you to choose the detail selection will appear at the bottom portion of the window:

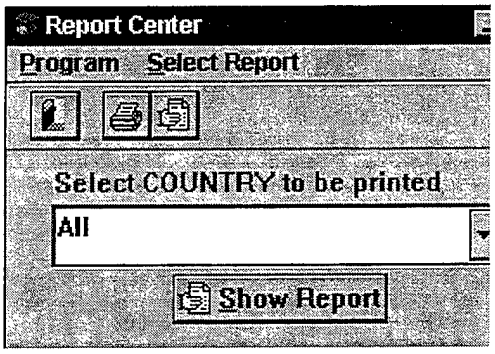


Figure 2.9 Report Selection Drop-down List

Reports and lists will be displayed on the report viewer screen. See Chapter 1 "Common Features of the IPOMIS Applications - How to Use Report Viewer".

Part 2

IMS Information Viewer

IMS Information Viewer is designed to enable the office staff to access information about IMSs.

You can view detail information about IMSs, their dependents, and their sponsors

1. Starting the Viewer

Start the viewer by clicking the IMS Information Viewer icon  in the IPOMIS program group. Database Login window will then open and prompt you to enter your User Name and Password (see Chapter 1 "Common Features of the IPOMIS Applications - How to Login Database").

> ***Important Note:*** Because of security concerns, the viewer is designed to be used by International Programs Office staff. If you attempt to enter as another user then program will terminate and you will not be able use the program. Consult with the IPO Database Administrator.

If you provide proper user name and password then program login you to the system's database and the SIV window will then open:

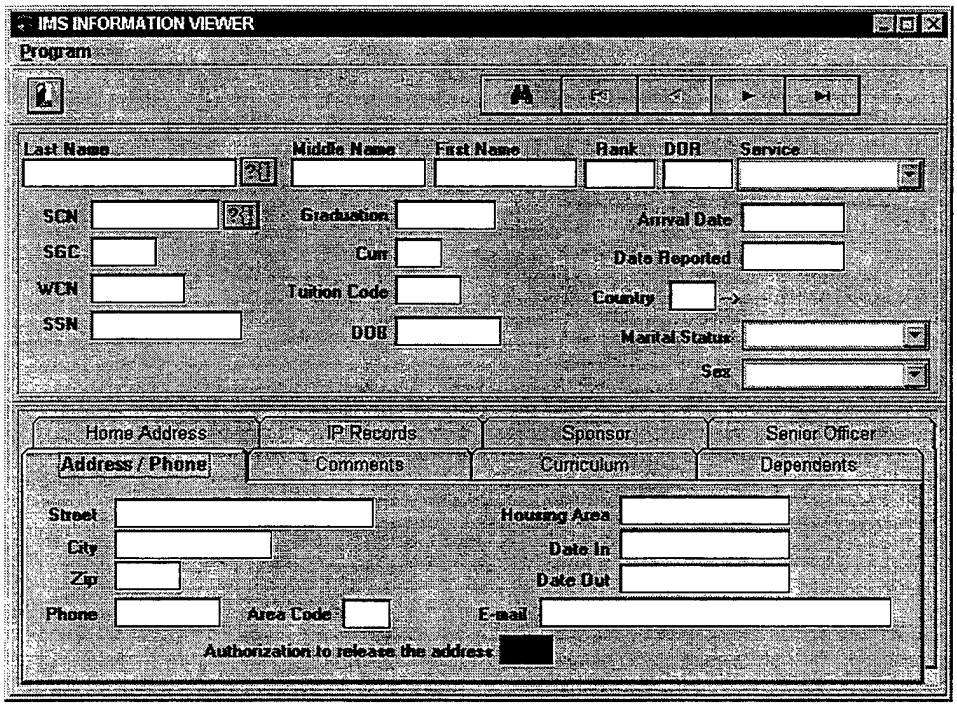


Figure 3.12 IMS Information Viewer

This window consists of the:

- Menu bar, across the top of the window, contains Program menu item.
- Speedbar, consists shortcut speed button for exiting from the program and a row of database navigate buttons to navigate on IMS table, just below the menu bar.
- IMS Information window, displayed below the Speedbar, showing the IMS currently selected.
- Tabbed notebook, displayed below the IMS Information window. This area displays detail information about the IMS selected and it is organized as tabs:

a. Program Menu

This menu item consists of two commands:

- *About* to see the about window of the program
- *Exit* to exit the program

b. Tabbed Notebook

This tabbed notebook consists of seven tabs:

- Address/Phone : enables you to review selected IMS's address and phone data.
- Comments: enables you to review comments about the selected IMS.
- Curriculum: enables you to review detail information about the curriculum enrolled by the selected IMS.
- Dependent(s): enables you to review information about IMS's dependents.
- Home Address: enables you to review the selected IMS's address in his/her home country.
- IP Records: enables you to review IMS's DLI Attendance status and IP status.
- Sponsor(s): enables you to review selected IMS's sponsor(s).
- Senior Officer: enables you to review senior officer of the country where the selected IMS is from.

2. Searching for a IMS

See part 1 section "2. Locating IMS".

3

SPONSOR APPLICATIONS

This chapter provides detailed information about IPOMIS's Sponsor Program Manager and Sponsor Information Viewer Applications.

Sponsor Program Manager (SPM) is designed to assist International Student Assistant to perform his/her administrative tasks associated with managing the sponsor program.

Sponsor Information Viewer is designed to enable the office staff to access information about sponsors.

Part 1

Sponsor Program Manager

Sponsor Program Manager (SPM) enables International Student Assistant to:

- Manage database portion associated with Sponsor Program:
    - Add/Delete sponsors
    - Add/Delete sponsor dependents
- Assign sponsors
- Generate sponsor reports and lists

1. Starting SPM and Main Window

Start SPM by clicking the Sponsor Program Manager icon 

in the IPOMIS program group. Database Login window will then open and prompt you to enter your User Name and Password (see Chapter 1 "Common Features of the IPOMIS Applications - How to Login Database").

> ***Important Note:*** Because of security concerns, SPM is designed to be used by International Student Assistant. If you attempt to enter as another user then program will terminate and you will not be able use the program. For further information see Chapter 5 "Administrating Security" and consult with your Database Administrator.

If you provide proper user name and password then program login you to the system's database and the SPM main window will then open:

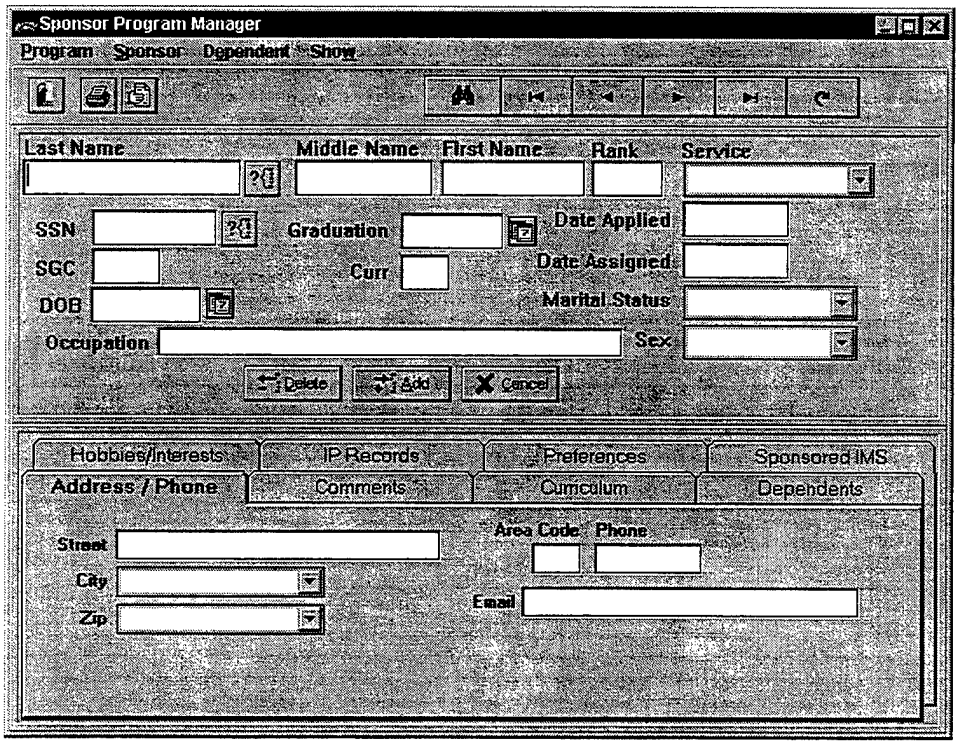


Figure 3.1 Sponsor Program Manager Window

This window consists of the:

- Menu bar, across the top of the window, containing commands you can choose to perform SPM tasks.
- Speedbar, consists shortcut speed buttons and a row of database navigate buttons to navigate on Sponsor table, just below the menu bar.
- Sponsor Information window, displayed below the Speedbar, showing Sponsor currently selected and Delete/Add/Cancel buttons to maintain sponsor data.
- Tabbed notebook, displayed below the Sponsor Information window. This area displays detail information about the sponsor selected organized as tabs.

a. SPM Menus

The SPM menus are the basic way to perform tasks with the SPM. There are four pull-down menus:

- Program menu: enables you to exit SPM, setup the printer and see about screen.
- Sponsor menu: enables you to
  - Assign sponsor,
  - Delete sponsor assignment,
  - Add and delete the selected sponsor,
  - Remove graduating sponsors from the database.
- Dependent menu: enables you to add dependents to the selected sponsor and edit the dependents.
- Show menu: enables you to review various sponsor rosters and information sheet of the selected sponsor.

Each pull-down menu item can be activated clicking them or pressing both Alt button and the underlined letter in the menu item. For example: Program menu item will be activated pressing **Alt+P**. Menu items indicate shortcuts (Hot Keys) for each task,

such as "Sponsor | Add Alt+A" indicates that if **Alt+A** is pressed then program invokes adding sponsor process.

b. Speedbar

Consist of three speed buttons:

 for exiting from the SPM

 for setting up the printer

 for previewing sponsor's information sheet

and a row of database navigate buttons to navigate on Sponsor table (see Chapter 1 "Common Features of the IPOMIS Applications - How to Use Database Navigator").

c. Sponsor Information Window

This window consists of two parts

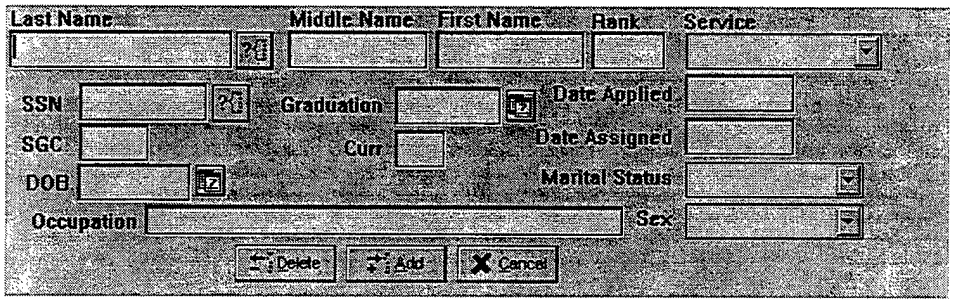


Figure 3.2 Sponsor Information Window

- Sponsor information: where you can see and edit sponsor data. You can navigate between edit fields pressing Tab / Shift+Tab key or clicking the field you want to edit.
- Task buttons: enable you to perform SPM database tasks, such as adding new sponsor, deleting selected sponsor or cancel/undo latest editing. Add and delete tasks can also be performed using the menu items.

d. Tabbed Notebook

This tabbed notebook consists of seven tabs:

- Address/Phone : enables you to review or edit the selected sponsor's address and phone data.
- Comments: enables you to review or edit comments about the selected sponsor.
- Curriculum: enables you to review detail information about the curriculum enrolled by the selected sponsor.
- Dependents: enables you to review information about sponsor's dependents.
- Hobbies/Interests: enables you to review or edit the selected sponsor's hobbies and interests. This information is used while assigning sponsors.
- Preferences: enables you to review or edit selected sponsor's preferences on IMS whom he/she wants to sponsor. This information is used while assigning sponsors.
- Sponsored IMS: enables you to review IMS(s) sponsored by the selected sponsor.

2. Locating Sponsor

Before performing any task with the SPM you must select the sponsor you want to work on. In order to assist you locating an activity, SPM has "Search by Last Name" and "Search by SSN" features.

To locate a sponsor:

- You can locate a sponsor anytime searching by his/her last name or SSN.

- *To search by last name:* click one of "Search By Last Name" buttons (left of navigator on the speedbar or next to Last Name field).

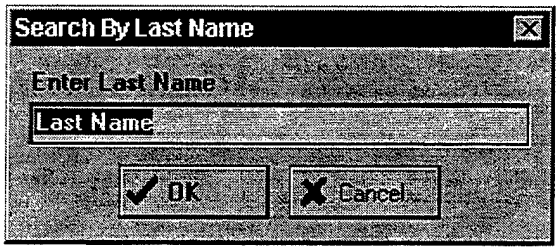


Figure 3.3 SPM Search by Last Name Dialog

- Enter the last name in either uppercase or lowercase. You do not have to type full name of the activity. Program will try to locate nearest match.
  - *To search by SSN:* click "Search By SSN" button next to the SSN field.
- Enter the SSN. You do not have to type full SSN. Program will try to locate nearest match.
- Click OK to search or Cancel to abort search.

  Note: After search, be sure that the sponsor shown on the Sponsor Information window is the sponsor that you want to work on. If you searched for the nearest match and you could not locate the sponsor then use navigate buttons to search.

3. Adding New Sponsor

To add a new sponsor:

Click "Add" button or press **Alt+A** or choose "Sponsor | Add". Then all edit fields on the screen will be blank and cursor will jump to the Last Name field to enable you to start editing. You may use **Tab** to jump next or **Shift+Tab** to previous field.

- In order to fill ***DOB*** or ***Graduation*** fields you may make use of a calendar. Press  button to pick a date from calendar (see Chapter 1 "Common Features of the IPOMIS Applications - How to Use Pick-a-date Calendar").
- Program will generate a temporary ***SSN*** for the new sponsor. If you do not know his/her actual SSN use this number. It is preferred to provide actual SSNs to the program.

  Important Note: DO NOT change SSN after assignment or adding dependents. If you attempt to change sponsor's SSN after assigning as sponsor or adding dependents, program will generate database errors. Because a sponsor is uniquely identified with the SSN in the database every association with a sponsor is created using the sponsor's SSN. Changing causes to interrupt these associations and database never allows you to interrupt the associations once they are formed.

  If you really need to change a sponsor's SSN:

  - Tell IP Coordinator to note somewhere and delete sponsor's IP participation information. After you changed sponsor's SSN, IP coordinator can reenter the deleted information.
  - Delete sponsor's dependents. Before deleting, note somewhere the information about dependents.
  - Delete sponsor's assignments.
  - After deleting, reenter assignments and dependent information.

- After editing each field click  on the navigator to refresh the database and actually post your new sponsor record to the database. Before refreshing/posting,

remember almost each field on the screen must have been filled in order to add a new sponsor successfully. Otherwise program will warn you with an error message indicating that blank field must have a value.

- Before refreshing your data, you can Cancel/Undo the editing you made: click "Cancel" button or press **Alt+C**.

Now you are ready to assign, add dependents, etc. to your new sponsor record.

4. Deleting Sponsor

To delete a sponsor:

Locate the sponsor you want to delete and Click "Delete" button or press **Alt+D** or choose "Sponsor | Delete". You will be prompted to confirm delete. If you choose Yes then delete process will initiate and all dependents, IP activities participated, and assignments associated with the selected sponsor will be deleted. If you choose No the information about the sponsor will not be deleted.

> ***Note:*** Before deleting a sponsor you must be sure about it. Because when a sponsor is deleted there is no way to revitalize the data associated with the sponsor.

3. Adding New Dependent

To add a new dependent:

Press **Ctrl+D** or choose "Dependent | Add". Sponsor's Dependent(s) Information window will then open:

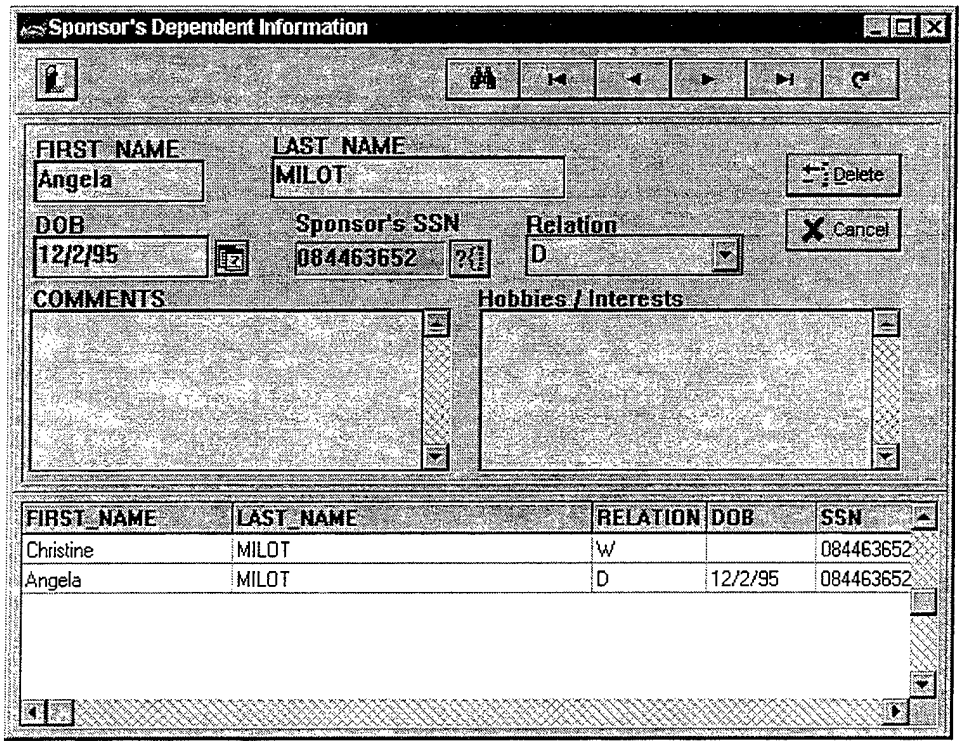


Figure 3.4 Sponsor's Dependent(s) Information Window

- Program will prompt you to enter: ***First Name:*** type first name of the new dependent with uppercase initial. ***Relation:*** type W for wife, H for husband, D for daughter, S for son or O for other dependents: cousin, nephew, etc. Choose OK on each dialog boxes. If you choose Cancel then the add dependent process will be canceled.
- By default program will fill the ***Last Name*** and ***SSN*** fields with the sponsor's. You can edit Last Name field anytime clicking it, but you cannot edit SSN field. It is a read-only field.
- In order to fill ***DOB*** field you may make use of a calendar. Press button to pick a date from calendar (see Chapter 1 "Common Features of the IPOMIS Applications - How to Use Pick-a-date Calendar").
- ***Comments*** and ***Hobbies/Interests*** fields are plain text fields used to enter detail information about the dependent.

- Click  on the navigator to refresh the database and actually post your new dependent record to the database. Otherwise program will warn you and error message will appear on the screen indicating that blank field must have a value.
- Before refreshing your data, you can Cancel/Undo the editing you made: click "Cancel" button or press **Alt+C**.

4. Editing a Dependent

To edit dependent data:

Press **Ctrl+E** or choose "Dependent | Edit". If the sponsor has a dependent "Sponsor's Dependent(s) Information" window will then open (see Figure 3.4) and dependent data will be displayed. If the sponsor has no dependent then program will remind you that.

- Click on the field you want to edit.
- You can use pull down menu to select relation code from the list. It is recommended.
- You can use Pick-a-date calendar to fill DOB field as described in the previous section.

4. Assigning Sponsor

To assign sponsor:

Press **Ctrl+A** or choose "Sponsor | Assign". "Sponsor Assignment" window will then open:

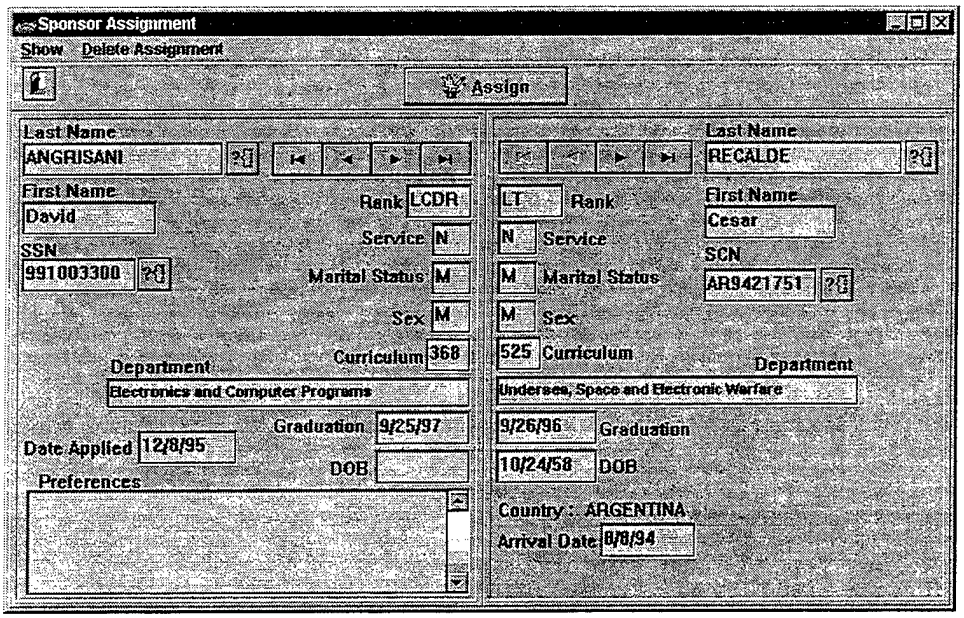


Figure 3.5 Sponsor Assign Window

Sponsor Assignment window consist of four parts: Menu bar, Speedbar, Sponsor info, and IMS info.

- Menu bar, across the top of the window, containing Show and Delete Assignment commands you can choose.
- Speedbar, consists a shortcut speed button to close the assignment window and Assign button to initiate assignment process, just below the menu bar.
- Sponsor Info window, displayed on the left of the window below the Speedbar, showing Sponsor currently selected to be assigned.
- IMS Info window, displayed on the right of the window below the Speedbar, showing IMS currently selected to be sponsored.

- Before assigning sponsor make sure that you selected the right IMS for the right sponsor. You can locate sponsor and IMS using search by SSN, SCN, Last Name buttons  next to the relevant fields.
- After selecting sponsor and IMS click Assign button  or press "**Alt+A**". If the assignment process was completed successfully then program shows a

message stating that "(Sponsor last name) will sponsor (IMS last name)". If the sponsor has already assigned to the selected IMS then program will ignore the assignment and displays an error message dialog.

- You can continue assigning sponsors or close the assignment window clicking close button on the speedbar.

5. Delete Assignment

There are two ways to delete assignments:

Choose "Sponsor | Delete Assignment" on the SPM main window or choose "Delete Assignment" on the Sponsor Assignment window. Both will perform the exact same task and "Delete Assignment" window will open:

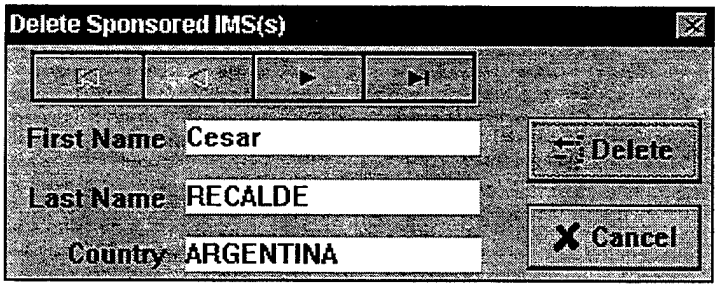


Figure 3.6 Delete Assignment Window

If the sponsor has any assignment it will show up on the delete assignment window. Click Delete to delete assignment, click Cancel to cancel operation.

If the sponsor does not have any assignment the fields on the window will be blank. Click Cancel to cancel operation and exit.

6. Removing Graduates

When the student sponsors graduate you may want to remove their records from the database. You can do this automatically choosing "Sponsor | Remove Graduates" or pressing "**Alt+G**". A dialog box prompting you to enter or confirm the graduation date will then appear on the screen:

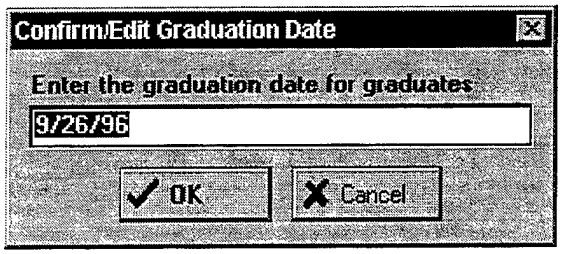


Figure 3.7 Remove Graduates Dialog

If the graduation date on the screen is correct for you then click OK. If not then type the graduation date of the sponsors you want to remove from the database. You can cancel this operation clicking Cancel instead of OK.

> ***Note:*** Before removing graduating or graduated sponsors you must be sure about it. Because when you select OK on the Remove Graduates Dialog, all information about the graduates is deleted. There is no way to revitalize them again.

7. Preparing Reports and Lists

The SPM has three built-in rosters and one sponsor information sheet format.

a. All Registered Sponsors

This roster lists all registered sponsors – i.e. all sponsors with or without assignment in the database. You can use this roster to review some information about sponsors: Last Name, First Name, etc.

Sponsor List (Ordered By Last Name)
(Consists all registered sponsors) 7/8/96

Last Name	First Name	Serv.	Rank	Curr.#	Graduation	M.Stat	# of IMSs Sponsored
ANGRISANI	David	N	LCDR	368	9/25/97	M	1
BARSANTI	Robert	N	LCDR	525	9/26/96	M	0
BAUER	David	N	LCDR	610	9/26/96	M	0
BAUMANN	Gregg	N	LT	570	3/27/97	M	1
BECKMAN	Philip	N	LT	360	3/27/97	M	1

Figure 3.8 All Registered Sponsors Roster

Choose "Show | Roster | All Registered Sponsors" on the SPM main window to preview the roster.

b. All Assigned Sponsors

This roster lists all assigned sponsors – i.e. all sponsors with assignment in the database. You can use this roster to review some information about sponsors and sponsored IMSs:

Sponsor Roster (Ordered by last name)
(Consists all assigned sponsors) 7/8/96

SPONSOR				IMS			
Last Name	First Name	Serv.	Rank	Last Name	First Name	Serv.	Rank
ANGRISANI	David	N	LCDR	RECALDE	Cesar	N	LT
BAUMANN	Gregg	N	LT	MURAT	Mustafa	N	LTJG
BECKMAN	Philip	N	LT	LIU	Yi	N	LT

Figure 3.9 All Assigned Sponsors Roster

Choose "Show | Roster | All Assigned Sponsors" on the SPM main window to preview the roster.

c. Graduating Sponsors

This roster lists graduating sponsors. You can use this roster to review some information about graduating sponsors:

Sponsors Graduating on 9/26/96 7/8/96

Last Name	First Name	SSN	Svc.	Rank	Curr.#	M.Stat	Spouse Name
BARSANTI	Robert	100011073	N	LCDR	525	M	Jean
BAUER	David	100011098	N	LCDR	610	M	Val
BERGMAN	Steve	100011087	N	LT	825	S	
BRADY	Albert	100011003	N	LT	370	M	Jennifer

Figure 3.10 Graduating Sponsors Roster

Choose "Show | Roster | Graduating Sponsors" on the SPM main window to preview the roster. A dialog box prompting you to enter or confirm the graduation date will then appear on the screen (see Figure 3.7). If the graduation date on the

screen is correct for you then click OK. If not then type the graduation date of the sponsors you want to remove from the database. You can cancel this operation clicking Cancel instead of OK.

c. Sponsor Information Sheet

This report is designed to provide detail information about a sponsor on a hard copy. You can use this roster to print detail information about the sponsor selected:

Sponsor Information 7/8/96

First Name Joshua
Middle Name K.
Last Name DOE
Service Navy
Rank LCDR
SGC 9999
Graduation 9/26/96
Curriculum 370

Spouse Name Jane
Child(ren) 2

Local Address
9999 Main St
Monterey 93940

Phone (408) 555 5555

Comments :
Very active and supportive

Sponsored Student(s) :

CELEBI	Onder	TURKEY	9/26/96

Figure 3.11 Sponsor Information Sheet

Choose "Show | Info Sheet" or press "**Ctrl+I**" or click  icon on the SPM main window to preview the roster.

Part 2

Sponsor Information Viewer

Sponsor Information Viewer (SIV) is designed to enable the office staff to access information about sponsors.

You can view detail information about Sponsors and their dependents

1. Starting SIV

Start SIV by clicking the Sponsor Information Viewer icon 

in the IPOMIS program group. Database Login window will then open and prompt you to enter your User Name and Password (see Chapter 1 "Common Features of the IPOMIS Applications - How to Login Database").

> ***Important Note:*** Because of security concerns, SPM is designed to be used by International Programs Office staff. If you attempt to enter as another user then program will terminate and you will not be able use the program. Consult with your the IPO Database Administrator.

If you provide proper user name and password then program login you to the system's database and the SIV window will then open:

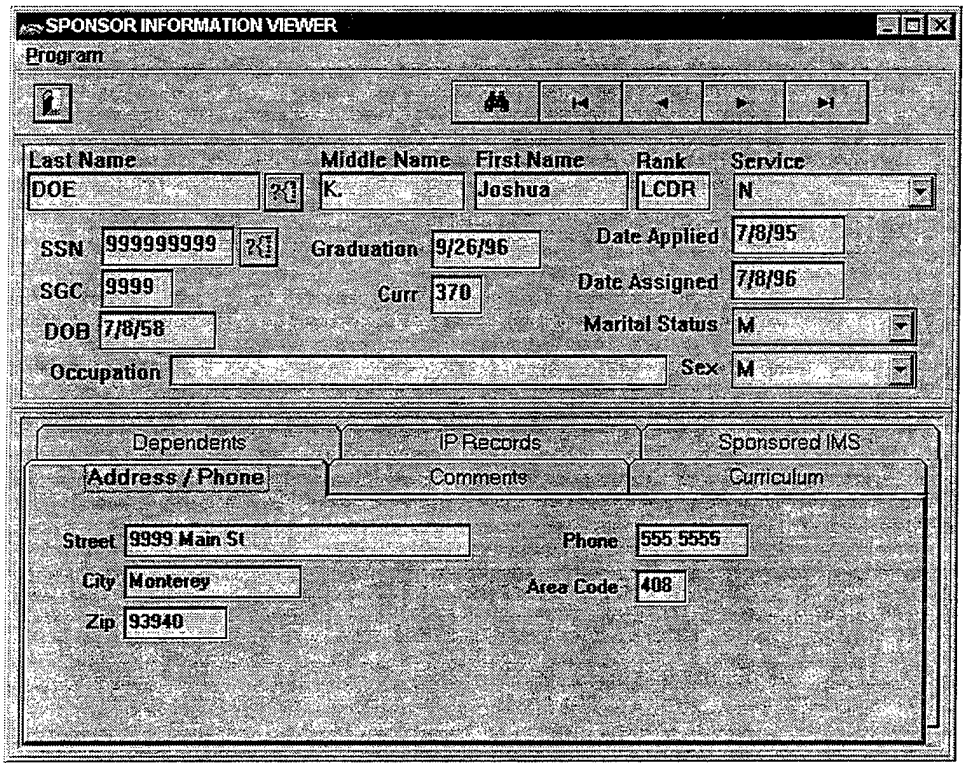


Figure 3.12 Sponsor Information Viewer

This window consists of the:

- Menu bar, across the top of the window, contains Program menu item.
- Speedbar, consists shortcut speed button for exiting from the program and a row of database navigate buttons to navigate on Sponsor table, just below the menu bar.
- Sponsor Information window, displayed below the Speedbar, showing Sponsor currently selected.
- Tabbed notebook, displayed below the Sponsor Information window. This area displays detail information about the sponsor selected.

a. Program Menu

This menu item consists of two commands:

- *About* to see the about window of the program
- *Exit* to exit the program

b. Tabbed Notebook

This tabbed notebook consists of six tabs:

- Address/Phone : enables you to review selected sponsor's address and phone data.
- Comments: enables you to review comments about the selected sponsor.
- Curriculum: enables you to review detail information about the curriculum enrolled by the selected sponsor.
- Dependents: enables you to review information about sponsor's dependents.
- Sponsored IMS: enables you to review IMS(s) sponsored by the selected sponsor.

2. Searching for a Sponsor

See part 1 section "2. Locating Sponsor".

4

INFORMATION PROGRAM MANAGER

This chapter provides detailed information about IPOMIS's Information Program Manager Application.

Information Program Manager is designed to assist Information Program (IP) Coordinator to perform his/her administrative tasks associated with managing and coordinating the IP activities.

Information Program Manager (IPM) enables IP Coordinator to:

- Manage database portion associated with IP activities:
    - Add/Delete activities
    - Add/Delete Student and Escort participation to the activities
    - Add/Delete Vendors providing services to the activities
    - Review previous activities (Archived activities)
- Generate Selection Lists
    - Prepare Selection list
    - Edit Selection list
    - Save Selection list
- Generate IP Reports
    - Prepare report

- Edit report
- Save report

1. Starting IPM and Main Window

Start IPM by clicking the IP Manager icon  in the IPOMIS program group. Database Login window will then open and prompt you to enter your User Name and Password (see Chapter 1 "Common Features of the IPOMIS Applications - How to Login Database").

> ***Important Note:*** Because of security concerns, IPM is designed to be used by IP Coordinator. If you attempt to enter as another user then program will terminate and you will not be able use the program. For further information see Chapter 5 "Administrating Security" and consult with your Database Administrator.

If you provide proper user name and password then program login you to the system's database and the IPM main window will then open:

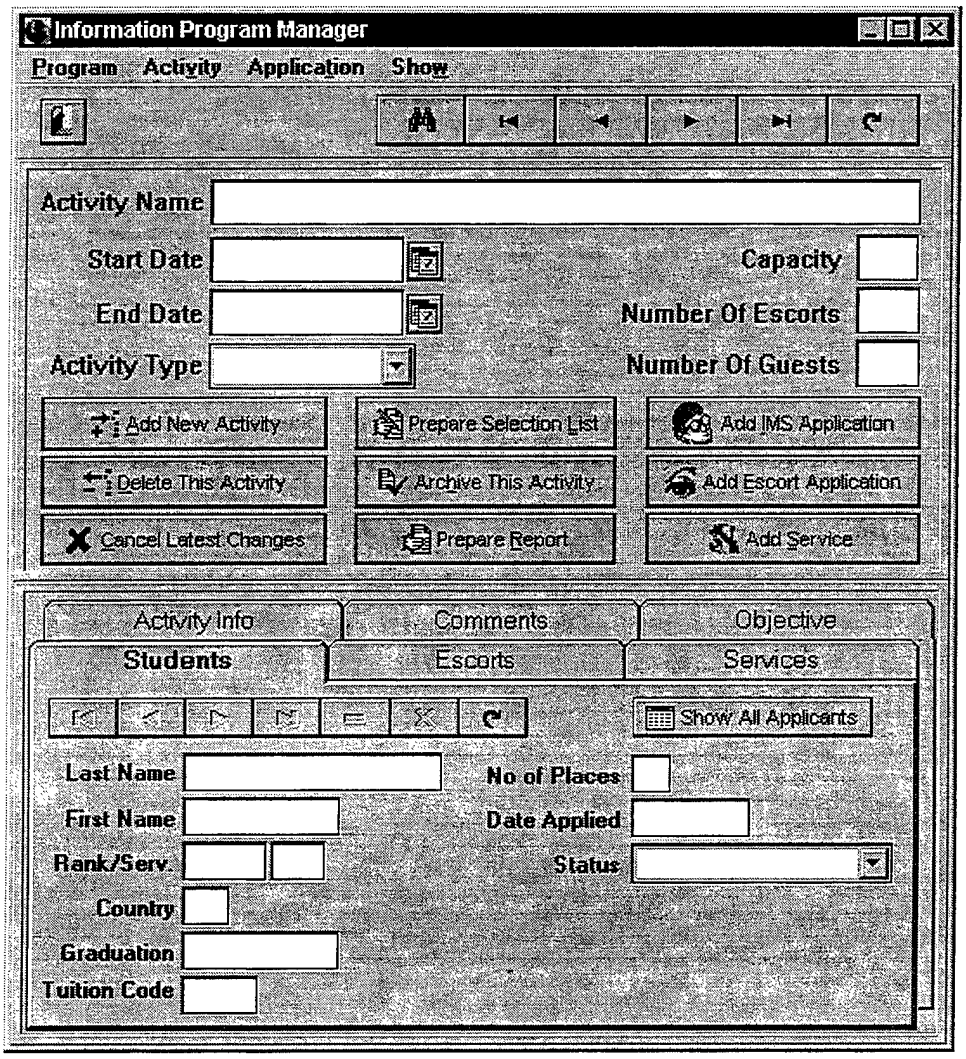


Figure 4.1 Information Program Manager Window

This window consists of the:

- Menu bar, across the top of the window, containing commands you can choose to perform IPM tasks.
- Speedbar, a shortcut speed button for exiting from the IPM and a row of database navigate buttons to navigate on IP activities table, just below the menu bar.

- IP Application window, displayed below the Speedbar, showing IP activity currently selected and the task buttons.
- Tabbed notebook, displayed below the IP Application window. This area displays detail information about the activity selected organized as tabs.

a. IPM Menus

The IPM menus are the basic way to perform tasks with the IPM. There are four pull-down menus:

- Program menu: enables you to exit IPM, set the printer and see about screen.
- Activity menu: enables you to
  - Search an activity,
  - Add, delete, and archive the selected activity,
  - Prepare selection list or activity report.
- Application menu: enables you to add applications and services to the selected activity.
- Show menu: enables you to review applicants list and service providers for the selected activity.

Each pull-down menu item may be activated clicking them or pressing both Alt button and the underlined letter in the menu item. For example: Program menu item will be activated pressing **Alt+P**. Menu items indicate shortcuts (Hot Keys) for each task, such as "Application | Add IMS Alt+I" indicates that if **Alt+I** is pressed then program invokes adding IMS application process.

b. Speedbar

Consist of a shortcut speed button  for exiting from the IPM and a row of database navigate buttons to navigate on IP activities table (see Chapter 1 "Common Features of the IPOMIS Applications - How to Use Database Navigator").

c. Activity Window

This window consist of two parts

- IP activity information: where you can see and edit activity data. You can navigate between edit fields pressing Tab key or click on the field you want to edit.

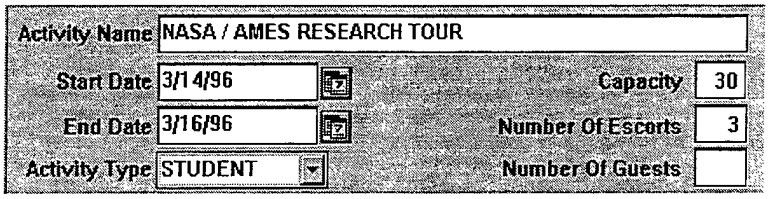


Figure 4.2 IPM Activity Information Window

- Task buttons: enable you to perform IPM tasks. These tasks can also be performed using Menu items.

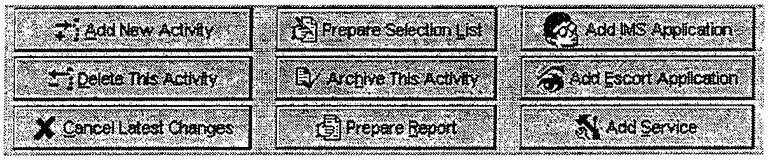


Figure 4.3 IPM Task Buttons

d. Tabbed Notebook

This tabbed notebook consists of six tabs:

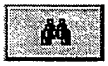
- Activity Info: enables you to review or edit general information about the activity selected.
- Comments: enables you to review or edit comments about the activity selected.
- Objective: enables you to review or edit official IP objective the activity selected.
- Students: enables you to review or edit information about IMSs applied/participated to the activity selected.

- Escorts: enables you to review or edit information about escorts applied/participated to the activity selected.
- Services: enables you to review or edit information about service providers and services used for the activity selected.

2. Locating An Activity

Before performing any task on an activity you must select the activity you want to work on. In order to assist you locating an activity, IPM has "Search by Activity Name" feature.

To locate an activity:

- Press **F3** key or choose "Activity | Search" or click Search by Name button  on the speedbar. Search dialog box will then open and prompt you to enter activity name:

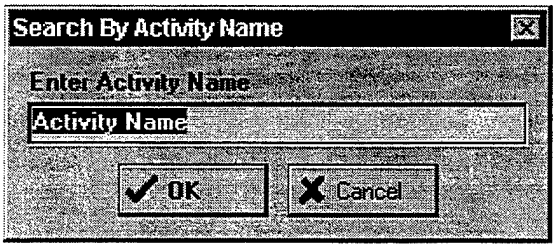


Figure 4.4 IPM Search by Activity Name Dialog

- Enter the activity name in either uppercase or lowercase. You do not have to type full name of the activity. Program will try to locate nearest match.
- Click OK to search or Cancel to abort search.

Note: After search, be sure that the activity shown on the Activity Information window is the activity that you want to work on. Because Activity Name and Start Date attributes are together unique identifiers of an IP activity defined in the IPOMIS database, you might have located an activity with the exact same name but a different start date. In this case use navigate buttons to search forward.

3. Adding New Activity

To add an activity:

Click "Add New Activity" button or press **Alt+A** or choose "Activity | Add". Then all edit fields on the screen will be blank and cursor will jump to the Activity Name field to enable you to start editing. You may use **Tab** to jump next or **Shift+Tab** to previous field.

- In order to fill ***Start Date*** or ***End Date*** fields you may make use of a calendar. Press  button to pick a date from calendar (see Chapter 1 "Common Features of the IPOMIS Applications - How to Use Pick-a-date Calendar").
- The ***Activity Type*** field in one of the most important fields for an activity, therefore you should pick one of the types from the drop down menu.
- You do not have to edit ***Capacity*** or ***Number of Escorts*** fields at this moment. You will be prompted to enter these values when the prepare selection list process initiated.
- Click  on the navigator to refresh the database and actually post your new activity data to the database. Before refreshing/posting, remember at least both Activity Name and Start Date fields must have been edited in order to add a new activity successfully. Otherwise program will warn you and error message will appear on the screen indicating that blank field must have a value.
- Before refreshing your data, you can Cancel/Undo the editing you made: click "Cancel Latest Changes" button or press **Alt+C**.

Now you are ready to add applications and services to your new activity.

4. Deleting an Activity

To delete an activity:

Locate the activity you want to delete and Click "Delete This Activity" button or press **Alt+D** or choose "Activity | Delete". You will be prompted to confirm delete. If you

choose Yes then delete process will initiate and all applications and services associated with this activity will be deleted. If you choose No the activity will not be deleted.

> ***Note:*** Before deleting an activity you must be sure about it. Because when an activity is deleted there is no way to revitalize the data associated with the deleted activity.

5. Adding IMS Application

To add an IMS application:

Click "Add IMS Application" button or press **Alt+I** or choose "Application | Add IMS ". The IMS Application window will then open:

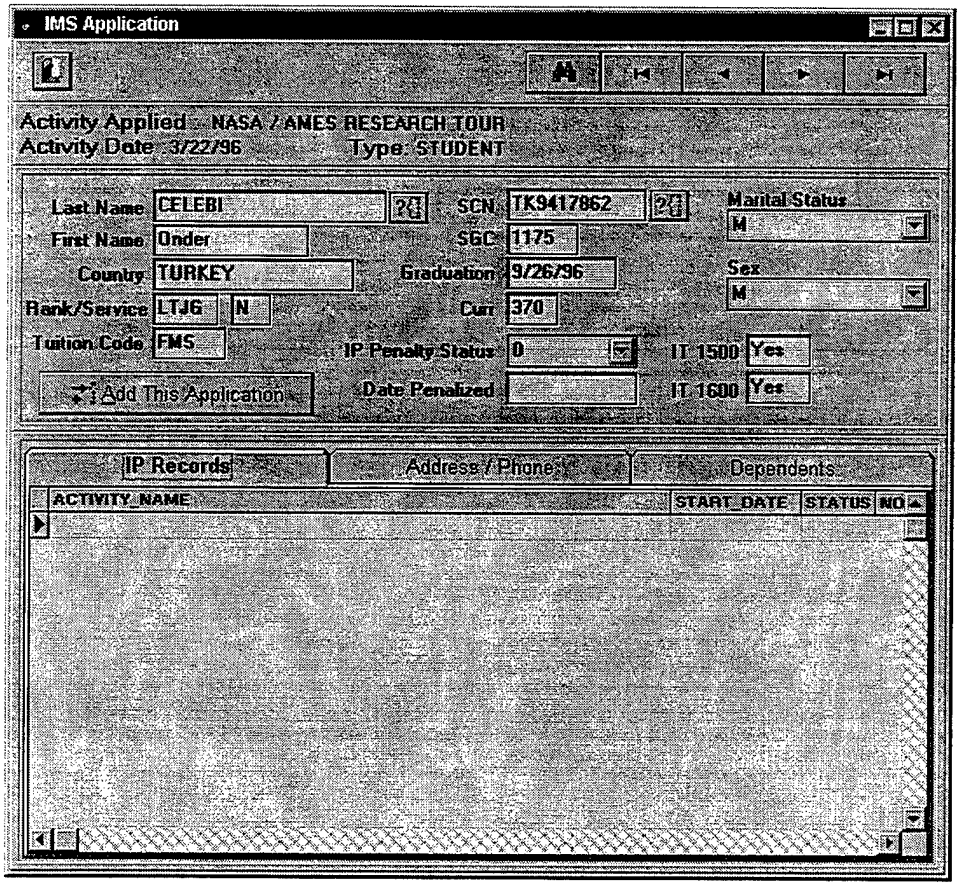


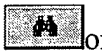
Figure 4.5 IMS Application Window

This window allows you to:

- View IMS Information including
    - Dependents
    - Previously participated activities
- Edit data associated with the Information Program, such as
    - Status about IT1500 or IT1600 classes
    - Penalty status
- Add IMS's application

Before adding an application you should locate the IMS to add his/her application. Therefore each time you open IMS Application window, first "Search By Last Name" dialog box appears:

- Enter the last name in either uppercase or lowercase. You do not have to type full last name. Program will try to locate nearest match. Click OK to search or Cancel to abort search.

    Note: You can locate an IMS anytime searching by his/her last name or SCN. To search by last name: click one of "Search By Last Name" buttons (left of navigator  or next to Last Name field ). To search by SCN: click "Search By SCN" button next to SCN  .

- Click on "Add This Application" button or press **Alt+A** to initiate adding process. "Application Information" window will then open:

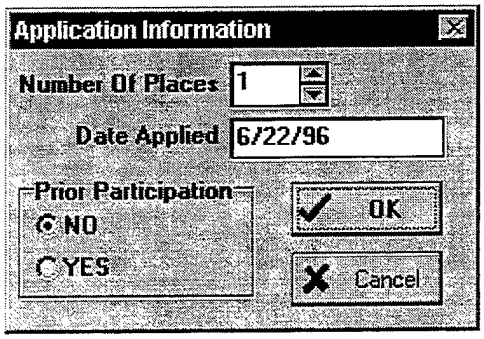


Figure 4.6 Application Information Window

- **Number of places:** allows you to enter the number of places requested by the applicant. You can increment or decrement the number by one pressing up or down buttons attached to the field. If the activity is a STUDENT activity then program does not let you to enter a number different from "1". If the activity is an ADULT activity then program does not let you to enter a number higher than "2". For FAMILY type activities, the only restriction is that you should enter a number greater than zero.

- **Date applied:** allows you to enter/edit the date of application. By default, this value is the day you enter application. The format of the date value is generally MM/DD/YY. Your PC's operating system configuration defines the format of the date. Refer to your PC's operating system manual.

- **Prior Participation:** allows you to enter if the applicant has participated this activity before. If you choose NO radio button then the status field for this application will be left blank. If you choose YES radio button then the status field for this application will be "P"(P indicates prior participation).

- Click the OK button to enter application information you filled out on the Application Information window. Or click CANCEL button to cancel this application. Clicking either button will return you to the IMS Application window so you can enter other applications or view IMS information.

  Note: If the application has already entered and you attempt to enter the same application then program will refuse to accept it and displays an error messages:

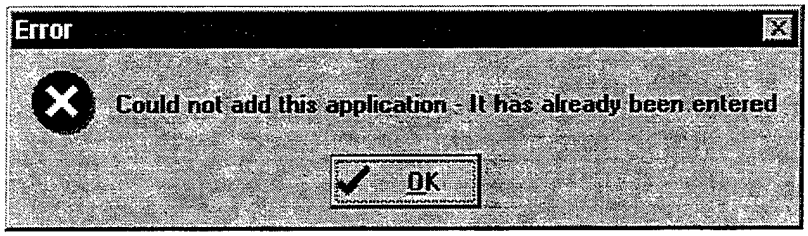


Figure 4.7 Application Enter Error Message

6. Adding Escort Application

To add an Escort application:

Click "Add Escort Application" button or press **Alt+E** or choose "Application | Add Escort ". The Escort Application window will then open:

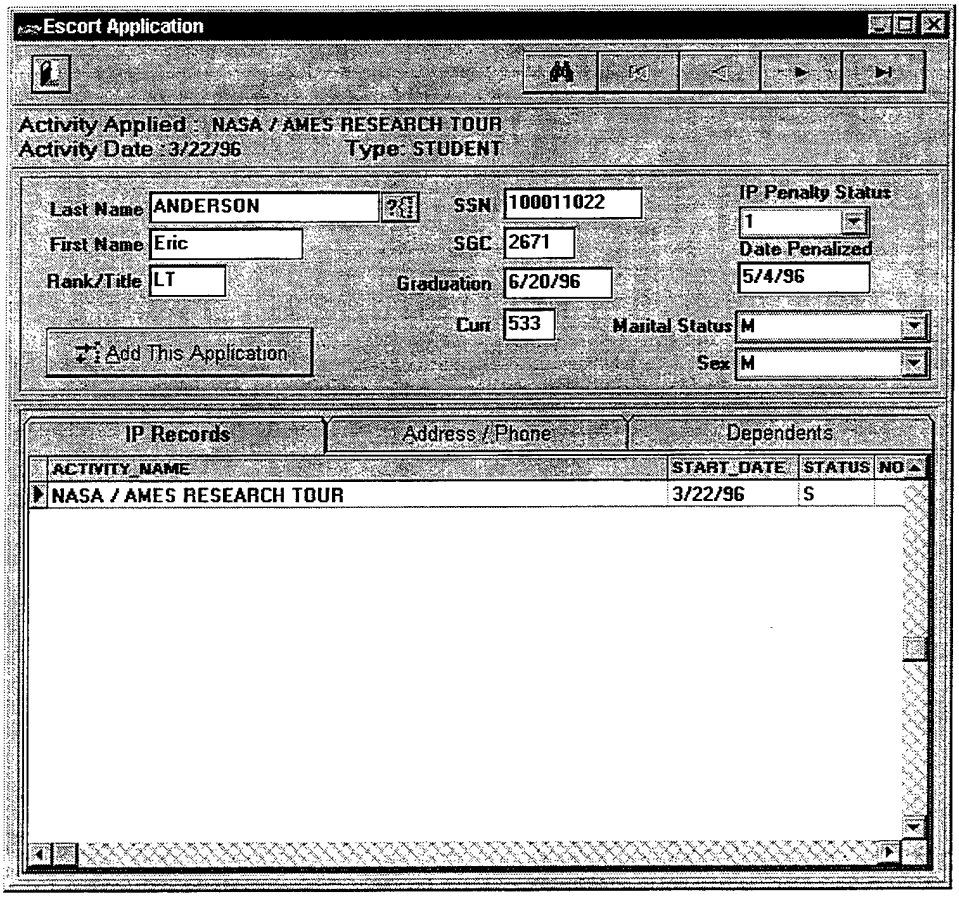


Figure 4.8 Escort Application Window

This window allows you to:

- View Escort Information including
    - Dependents
    - Previously participated activities
- Edit data associated with the Information Program, such as Penalty status
- Add Escorts' application

Before adding an application you should locate the Escort to add his/her application. Therefore each time you open Escort Application window, first "Search By Last Name" dialog box appears:

- Enter the last name in either uppercase or lowercase. You do not have to type full last name. Program will try to locate nearest match. Click OK to search or Cancel to abort search.

    Note: You can locate an Escort anytime searching by his/her last name. To search by last name: click one of "Search By Last Name" buttons (left of navigator  or next to Last Name field ).

- Click on "Add This Application" button or press **Alt+A** to initiate adding process. "Application Information" window will then open (see Figure 4.6). Rest of the process works as in the "Adding IMS Application".

7. Adding Service

To add a Service:

Click "Add Service" button or press **Alt+S** or choose "Application | Add Service ". The Services and Vendor Information window will then open:

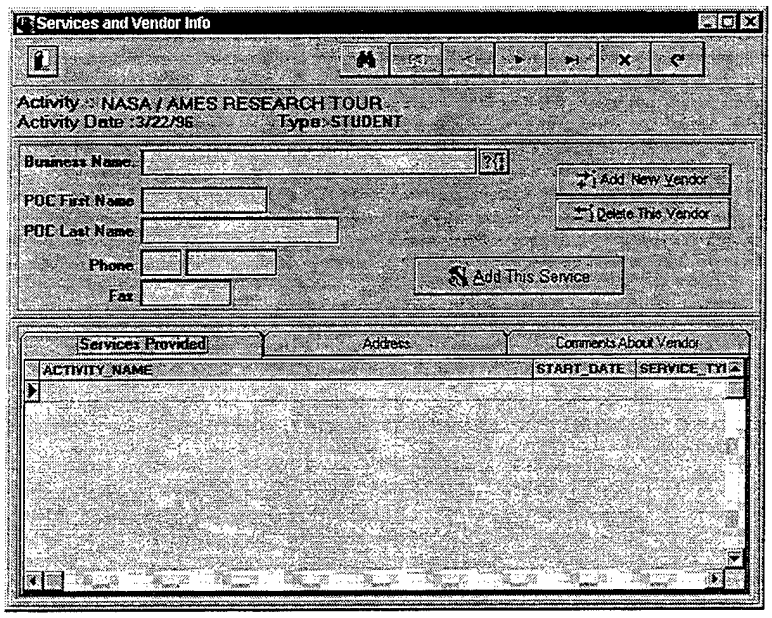


Figure 4.9 Services and Vendor Information Window

This window allows you to:

- Edit and View Vendor Information including
    - POC, Address, and Comments about vendor
    - Services provided to previous activities so far
- Add Service

Before adding a service you should locate the service provider (Vendor).

- To search a vendor by business name click one of "Search By Vendor's Business Name" buttons (left of navigator or next to Business Name field .
- Enter the business name in either uppercase or lowercase. You do not have to type full name. Program will try to locate nearest match. Click OK to search or Cancel to abort search.

- Click "Add This Service" button or press **Alt+A** to initiate adding process. "Service Information" window will then open:

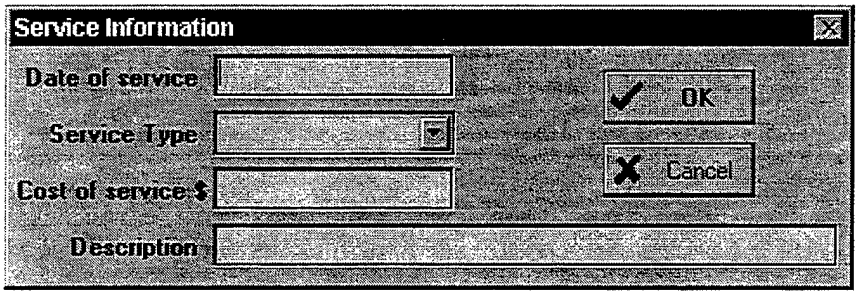


Figure 4.10 Service Information Window

  - **Date of service:** allows you to enter/edit the date of service. By default, this value is the "Start Day" of the activity. The format of the date value is generally MM/DD/YY. Your PC's operating system configuration defines the format of the date. Refer to your PC's operating system manual.
  - **Service Type:** allows you to select the type of the service from a pull down list. The list consist of TRAVEL, LODGING, MEALS, NO COST, and OTHER list items.
  - **Cost of service:** allows you to enter the cost of the service. General format is ...$$$$$$.¢¢
  - **Description:** allows you to enter a short description for the service. Especially "No Cost" and "Other" type of services require a short description. Maximum 50 characters can be entered.

- Click OK to enter service information you filled out on the window. Or click CANCEL to cancel this service. Clicking either button will return you to the main IPM window.

8. Deleting an Application or Service

You can delete an application or a service anytime before or after generating selection list.

- **To delete an IMS application:**
  - Select Students tab on the IPM main window.
  - Locate the IMS using navigator
  - Click Delete record button on the navigator.
  - Confirm the deletion on the dialog box pressing OK.
- **To delete an Escort application:**
  - Select Escorts tab on the IPM main window.
  - Locate the Escort using navigator
  - Click Delete record button on the navigator.
  - Confirm the deletion on the dialog box pressing OK.
- **To delete a Service** :
  - Select Services tab on the IPM main window.
  - Locate the service using navigator
  - Click Delete record button on the navigator.
  - Confirm the deletion on the dialog box pressing OK.

8. Reviewing Applications and Services

You can review IMS or Escort applications and services anytime before or after generating selection list. This reviews provide latest and detailed information about applications and services in the form of a list. You can also print these list to use for various purposes.

- **To review all IMS applications:**
  - Select Students tab on the IPM main window.
  - Click "Show All Applicants" button located on the right side of the tab.
- **To review all Escort applications:**
  - Select Escorts tab on the IPM main window.
  - Click "Show All Applicants" button located on the right side of the tab.

- **To review all Services:**
  - Select Services tab on the IPM main window.
  - Click "Show All Services" button located on the right side of the tab.

The applications or services list will then appear on the screen (see Chapter 1 "Common Features of the IPOMIS Applications - How to Use Report Viewer").Click Exit button after review.

9. Preparing Selection List

To prepare a selection list for the activity selected, click "Prepare Selection List" 

button on the IP Application window or press "**Alt+L**". The Selection List Pad window will then open:

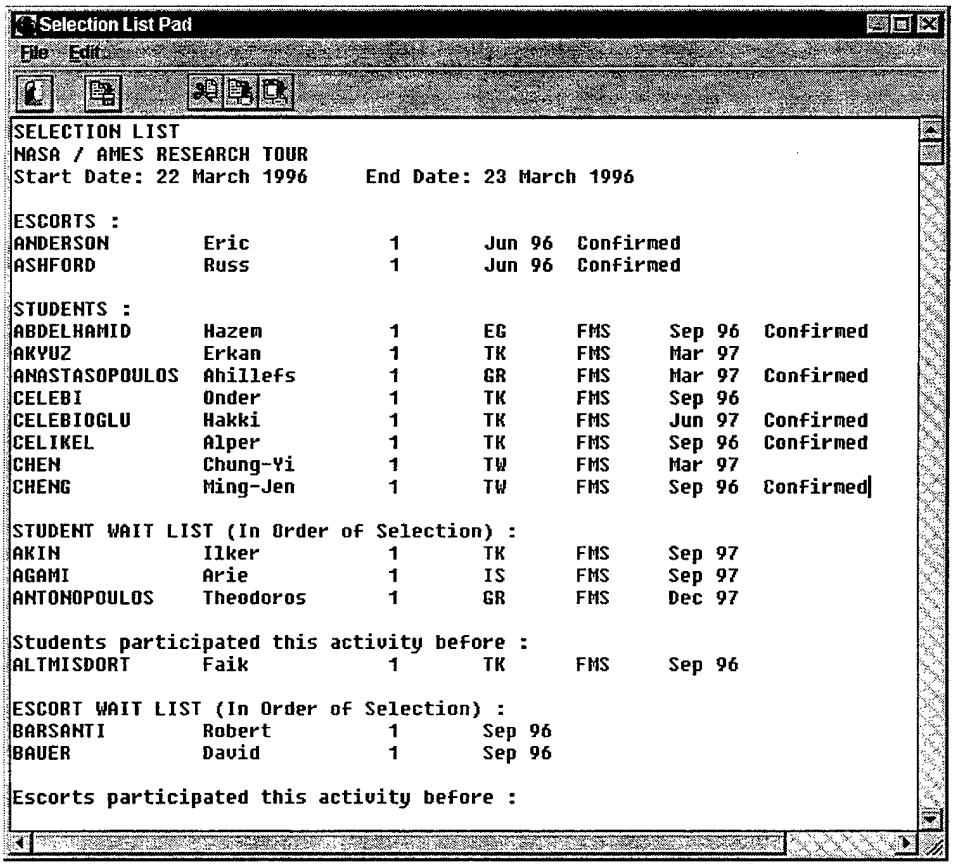


Figure 4.11 Selection List Pad Window

Selection List Pad window allows you to view, edit, and save the selection list generated by the program.

- To view the entire list use vertical and horizontal scroll bars.
- To edit the list use the "Cut / Copy / Paste" features under the edit menu item or use  buttons on the speed bar.
- To save the list click "Save" button  on the speed bar or choose "File | Save" on the menu. Save dialog box will then appear. Default settings on the dialog box:
  - File name: SELECT.TXT (In order to allow you import selection list file to any word processor without conversion problem, file format is selected as ASCII text, standard text format. You do not have to pick the default name. You can give any name to the file.)
  - Path: C:\IPOMIS\APPS\DATA (C drive is the server's main HDD)

10. Preparing Activity Report

To prepare a report for the activity selected, click "Prepare Report" 

button on the IP Application window or press "**Alt+R**" or choose "Activity | Report" from menu. The Report Pad window will then open. The report pad window is identical to the Selection List Pad window (see Figure 4.11)

Report Pad window allows you to view, edit, and save the activity report generated by the program.

- To edit the report use the "Cut / Copy / Paste" features under the edit menu item or use  buttons on the speed bar.
- To save the report click "Save" button  on the speed bar or choose "File | Save" on the menu. Save dialog box will then appear. Default settings on the dialog box:

- File name: REPORT.TXT (In order to allow you import report file to any word processor without conversion problem, file format is selected as ASCII text, standard text format. You do not have to pick the default name. You can give any name to the file.)
- Path: C:\IPOMIS\APPS\DATA (C drive is the server's main HDD)

11. Archiving an Activity

By archiving an activity/event you clean up the unnecessary records. Therefore the database does not have unnecessary records to keep. If you don't archive an activity database keeps all applications even they were not recorded as selected/confirmed. This makes the database larger and may slightly reduce the database performance.

To archive an activity click "Archive This Activity" button 

or press "**Alt+H**" or choose "Activity | Archive" from menu. During archiving process, program deletes the records associated with the activity archived if IMS or escort "status" is W-Waiting, X-Canceled, P-Prior Part. It leaves applicant records with S-Selected, C-Confirmed, and empty status. If status is D-declined or N-No show-up then program penalizes (increments "IP penalty status" by one) the applicant, and deletes this record.

5

SYSTEM UTILITIES

Part 1. Using Server Manager

This part introduces the InterBase Server Manager, a Windows application for monitoring and administering InterBase 4.0 databases and servers. Server Manager runs on a Windows Client, but can manage databases on any server on the local network.

Server Manager enables IPOMIS System Administrator to:

- Manage system's server security.
- Back up and restore system's database.
- Perform database maintenance, including:
    - Validating the integrity of a database.
    - Sweeping a database.
    - Recovering transactions that are "in limbo."

1. The Server Manager Window

Start Server Manager by clicking on the Server Manager icon  in the IPOMIS program group. The Server Manager window will then open:

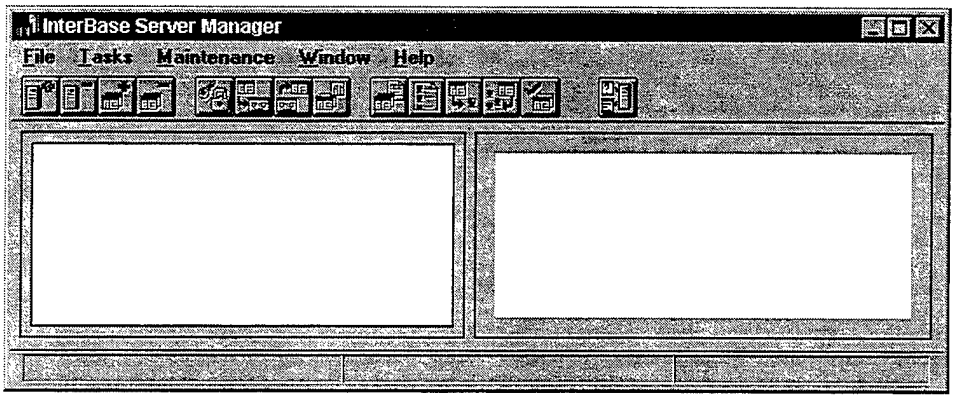


Figure 5.1 Server Manager Window

This window consists of the:

- Menu bar, across the top of the window, containing commands you can choose to perform DBA tasks with Server Manager.
- Speedbar, a row of shortcut buttons for menu commands, just below the menu bar.
- Server/database tree, displayed in the left side of the window below the Speedbar, showing the local server's name and the databases to which Server Manager is currently connected.
- Summary information area, displayed in the right side of the window below the Speedbar. This area displays information about the server or database, depending on which is selected in the server/database tree.
- Status line, that shows the current server and user login and flyover help for menus and the Speedbar.

a. Server Manager Menus

The Server Manager menus are the basic way to perform tasks with Server Manager. There are four pull-down menus:

- File menu: enables you to login to a server and logout, connect to a database, disconnect from a database, and exit Server Manager.

- Tasks menu: enables you to manage database security, perform backup and restoration, validate a database, open the database maintenance window, and start Windows ISQL.
- Window menu: enables you to close or minimize Server Manager windows.
- Help menu: provides online help.

b. Speedbar

The Speedbar is a row of buttons that are shortcuts for menu commands.

The Speedbar buttons are:

 Server login: opens the login dialog box, enabling you to log in to a remote Inter-Base server. The local server is already connected.

 Server logout: logout from the local server, and disconnect from any databases on that server to which you are currently connected.

 Database connect: opens a dialog box, enabling you to connect to a database on the current server.

 Database disconnect: disconnects Server Manager from the current database.

 Configure users: opens the User Configuration dialog box for administering server security.

 Database backup: opens the Database Backup dialog box.

 Database restore: opens the Database Restore dialog box.

 Database validation: opens the Database validation window, which enables you to perform database validations.

 Start ISQL: opens the Interactive SQL Window, and automatically connects to the current database.

c. Server/Database Tree

When the Server Manager window initially opens, the only menu or SpeedBar commands available are Server Login, Windows ISQL, and Help. Once connected to a database, all other commands are enabled.

You can connect to a database by clicking on the Database Connect SpeedBar button or choosing File | Database Connect.... A dialog box will open enabling you to enter the file and directory path of a database.

Once connected to a server, the server name is displayed on the left side of the Server Manager window. This area is called the server/database tree.

If Server Manager is not connected to any database on a server, a small dot will be displayed to the left of the server name. After connecting to a database, a "-" will be displayed instead. Each database to which Server Manager is connected is displayed beneath the server on which it resides in an expandable and collapsible tree.

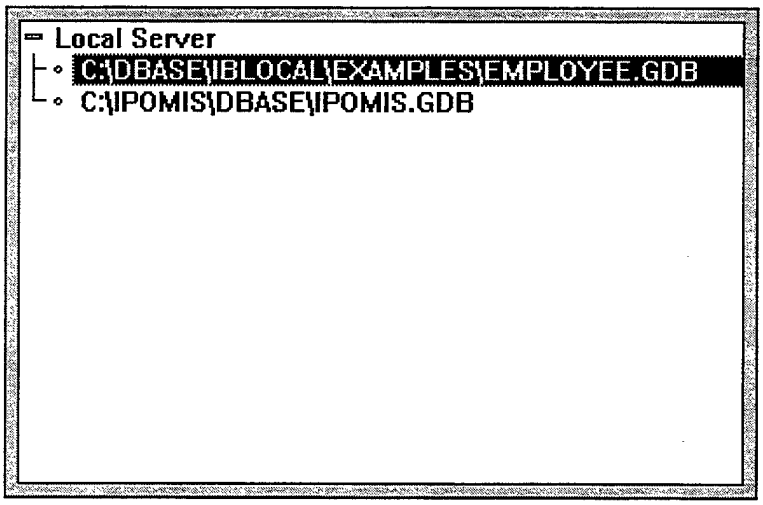


Figure 5.2 Server/Database Tree Window

Click on the "-" next to a server name (or double-click on the server name) to collapse the database tree for the server, and then a "+" will be displayed instead.

Click on the "+" next to a server name (or double-click on the server name) to expand the tree and display the names of all databases on that server to which Server Manager is currently connected. The "+" will become a "-".

In an expanded tree, click on a database name to highlight it. The highlighted database will be the one upon which Server Manager operates, referred to as the current database. When a database is highlighted, the server on which the data-base resides becomes the current server. Any actions of Server Manager then affect that server.

d. Summary Information Area

The summary information area in the right side of the server manger window displays information about the server or database currently selected in the server/database tree.

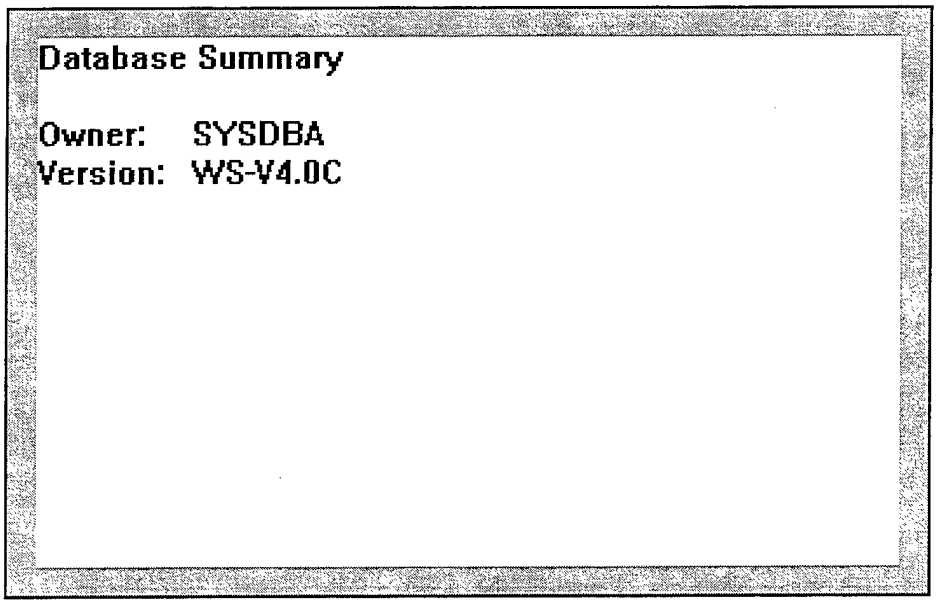


Figure 5.3 Summary Information Area Window

e. Using Online Help

Invoke the online help system by choosing a topic from the Help menu or clicking on a Help button in a dialog box. The help topic appropriate for the current context will

appear. All help topics are accessible through the Help Contents. For instructions on using the online help system, choose Help | Using Help.

2. Accessing IPOMIS Database

Before performing any database administration tasks, you must first connect to a database. Connect to the database by clicking on the Database Connect SpeedBar button or choosing File | Database Connect....

The InterBase Login Window will then appear:

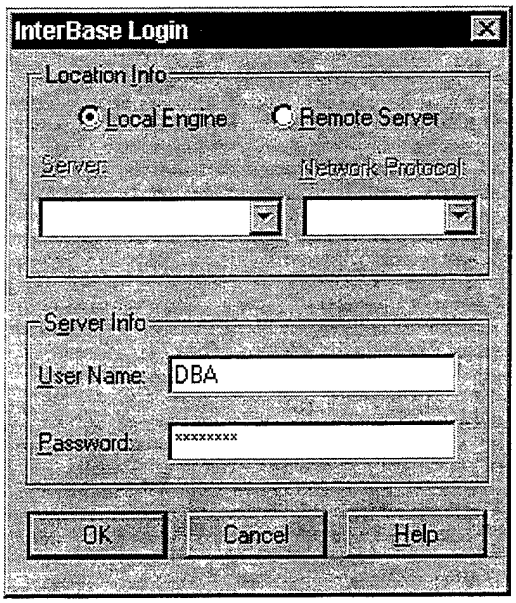


Figure 5.4 InterBase Login Window

Enter your user name and password properly then click OK to connect to the database. After connecting to a database, the Server Manager SpeedBar and menus will be active, and any actions you take will apply to the selected database.

3. Administering System Security

Server Manager enables you to:

- View the list of authorized users for the server.
- Authorize new users.
- Modify user information (user name, password).
- Remove users' authorization.

System Administrator can perform any of these tasks mentioned above. System Administrator must login to the server as SYSDBA with proper password.

Authorized users can only view the list of the authorized users. They cannot modify any part of system security data even their own user profile including password.

Caution: **User SYSDBA (System Administrator) has the highest privileges on system security. In order to avoid any security violation System Administrator must keep his/her password secret.**

After login properly to the server, choose Tasks | Security.... The InterBase Security dialog box will then open:

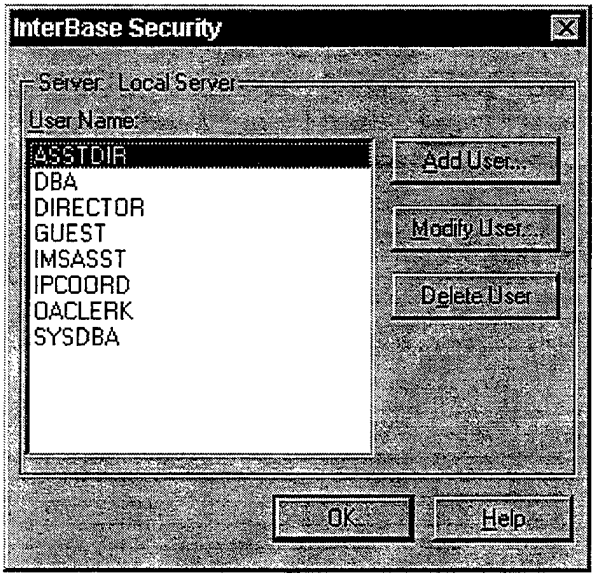


Figure 5.5 Security Dialog Box

- To view details of the user's profile : double click on the User's name you wish to view detailed information (if any) such as name (see Figure 5.5).
- To authorize a new user: click on the button "Add User".
- To modify the user profile: click on the name of the user to select the user to be modified and click on the button "Modify User" or double click on the User's name.

"User Configuration" window will then open:

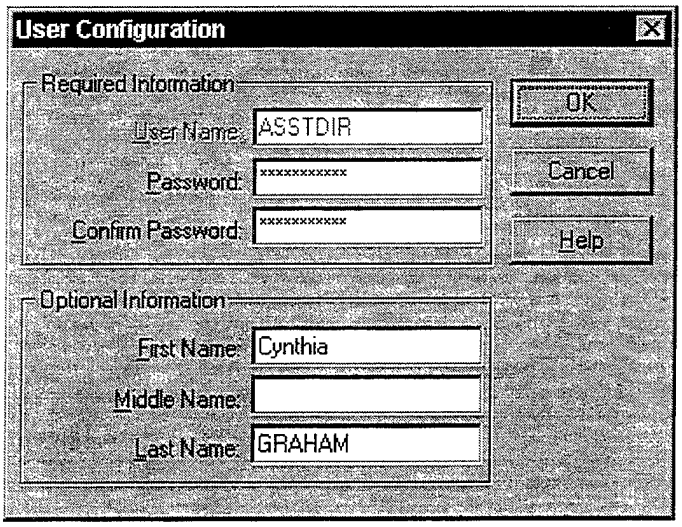


Figure 5.6 User Configuration Window

Any modification on the authorized users list will be updated and available immediately.

4. Maintaining IPOMIS Database

Database maintenance tasks include:

- Configuring database properties.
- Managing transaction recovery.
- Performing a database sweep.
- Validating and repairing a database.

Server Manager must be logged in to IPOMIS server and connected to the database before performing any of these operations. All of these tasks are performed from the Server Manager window.

a. Configuring Database Properties

To view and configure database properties, choose Maintenance | Database Properties... from the menu bar The Database Properties dialog box will then appear:

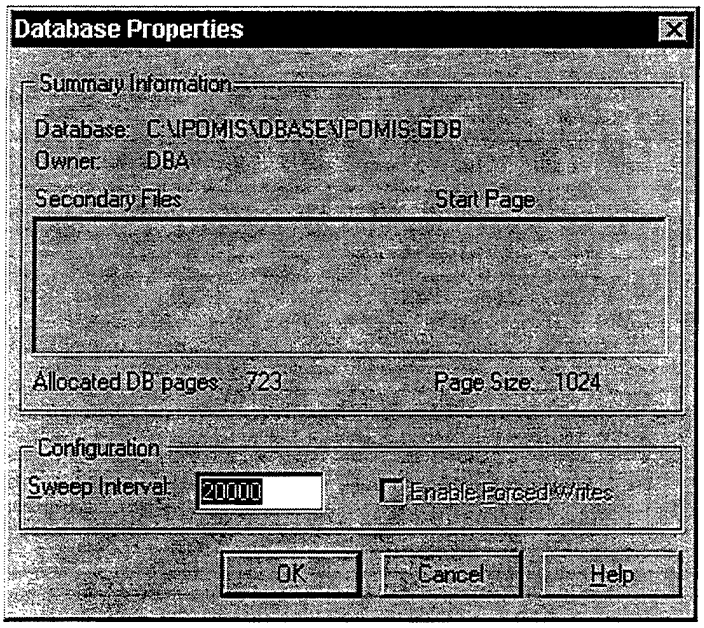


Figure 5.7 Database Properties Dialog Box

This dialog box contains a Summary Information area that displays properties but does not allow modification of them and a Configuration area that does allow modification of the database parameters.

The Summary Information area displays:

- Database name : *Path/* IPOMIS.GDB
- User name of the database owner : DBA
- Database Page Size : 1024
- Number of allocated pages : xxxx (e.g., 723)
- Secondary file names and sizes : N/A

The configuration area displays and allows modification of:

- Sweep interval : 2000
- Enabling of forced writes : Not Active

I. Adjusting Database Sweeping

Sweeping a database is a systematic way of removing outdated records from the database. Periodic sweeping prevents a database from growing too large.

Note to Database Administrator: Periodic sweeping is necessary and recommended. If sweeps are not made, old record versions will take up space and system memory. Regularly backing up and restoring a database can reduce the need to sweep. This enables you to maintain better application performance. For more information about the advantages of backing up and restoring, see "Backing Up and Restoring a Database."

You can sweep a database immediately by using the Maintenance | Database Sweep menu command.

II. Controlling Performance of Forced Writes

If forced writes are not enabled, then even though InterBase performs a write, the data may not be physically written to disk, because operating systems buffer disk writes. If there is a system failure before the data is written to disk, then information can be lost.

Performing forced writes ensures data integrity and safety, but will slow performance. In particular, operations which involve data modification will be slower.

When forced writes are enabled an "X" appears in the box labeled "Enable Forced Writes" in the Database Properties dialog box. To disable forced writes, click on the check box to remove the "X".

Caution **: If forced writes are enabled for a database, then the database will be subject to data loss if there is a hardware or other system failure. In general, it is best to have this feature active.**

b. Two-phase Commit and Transaction Recovery

When committing a transaction that spans multiple databases, InterBase automatically performs a two-phase commit. A two-phase commit guarantees that the transaction updates either all of the databases involved or none of them—data is never partially updated.

In the first phase of a two-phase commit, InterBase prepares each database for the commit by writing the changes from each subtransaction to the database. A subtransaction is the part of a multi-database transaction that involves only one database. In the second phase, InterBase marks each subtransaction as committed in the order that it was prepared.

If a two-phase commit fails during the second phase, some subtransactions will be committed and others will not be. A two-phase commit can fail if a network interruption or disk crash makes one or more databases unavailable. Failure of a two-phase commit causes limbo transactions, transactions that the server does not know whether to commit or roll back.

It is possible that a limbo transaction will make some records in a database inaccessible. To correct this, you must recover the transaction using Server Manager. Recovering a limbo transaction means committing it or rolling it back.

Recovering Transactions

To recover limbo transactions, choose Maintenance | Transaction Recovery... in the Database Maintenance window. A dialog box will then display a list of limbo transactions that can then be operated upon to recover—that is, to commit or roll back.

All the pending transactions in the database are listed in the scrolling area on the left side of the dialog box. Click on the "+" to display all the subtransactions of a transaction.

It is also possible to have a single database transaction that has been prepared and not committed. These transactions are displayed with a bullet to the left of the transaction. You can roll back or commit such transactions.

You can change the path of the database specified by each subtransaction by choosing Connect Path. Enter the directory path of the other database involved in the subtransaction, then choose OK.

The information on the path to the database was stored when the client application attempted the commit. Before attempting to roll back or commit any transaction, confirm the path of all involved databases is correct.

You can choose to either commit or roll back each transaction. To commit or roll back, select the desired transaction ID from the list and choose either Commit or Rollback.

Note: Only entire transactions can be recovered, so the commit and rollback buttons will only be enabled when the main transaction is selected. They will be disabled when a subtransaction is selected.

You can also seek advice by choosing the Advice button. A dialog box will then open and display information on each subtransaction: whether it has been committed, the remote server name, and database name. At the bottom, an action will be recommended: either commit or roll back.

Server Manager analyzes the state of subtransactions by determining when the two-phase commit failed. If the first transactions are in limbo but later transactions are not, Server Manager assumes that the prepare phase did not complete. In this case, you are prompted to do a rollback.

c. Performing an Immediate Database Sweep

To perform a database sweep, choose Maintenance | Database Sweep from the menu bar.

This operation runs an immediate sweep of the database, releasing space held by records which were rolled back and by out-of-date record versions. Sweeps are also done automatically at a specified interval; see "Adjusting Database Sweeping," in this chapter.

Important : Sweeping a database does not require it to be shut down. You can perform sweeping at any time, but it can impact system performance and should be done when it will least affect users.

d. Validating and Repairing a Database

In day-to-day operation, a database is sometimes subjected to events that pose minor problems to database structures. These events include:

- *Abnormal termination of a database application.* This does not affect the integrity of the database. When an application is canceled, committed data is preserved, and uncommitted changes are rolled back. If InterBase has already assigned a data page for the uncommitted changes, the page might be considered an orphan page. Orphan pages are unassigned disk space that should be returned to free space.
- *Write errors in the operating system or hardware.* These usually create a problem with database integrity. Write errors can result in "broken" or "lost" data structures, such as a database page or index. These corrupt data structures can make committed data unrecoverable.

You should validate a database:

- Whenever a database backup is unsuccessful.
- Whenever an application receives a "corrupt database" error.
- Periodically, to monitor for corrupt data structures or misallocated space.
- Any time you suspect data corruption.

To validate a database, choose Maintenance | Database Validation... in the Server Manager window. The following dialog box will open:

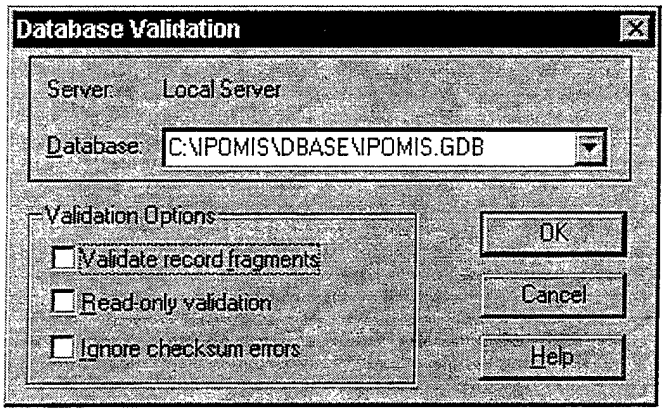


Figure 5.8 Database Validation Dialog Box

The name of the current database is displayed in the Database text field. Because there are some conditions such as a checksum error that will make it impossible to connect to a database, it is not necessary to connect to the database before per-forming a validation. If Server Manager is not connected to the database, you can enter the desired database name in the Database text field or select it from the drop down list by clicking on the arrow to the right of the field.

When Server Manager validates a database it verifies the integrity of data structures. Specifically, it will:

- Report corrupt data structures.
- Report misallocated data pages.
- Return orphan pages to free space.

I. Validation Options

You can select three options with Database Validation:

- Validate record fragments
- Read-only validation
- Ignore checksum errors

By default, database validation reports and releases only page structures. When you select the Validate record fragments option, validation reports and releases record structures as well as page structures.

By default, validating a database updates it, if necessary. To prevent updating, select the Read-only validation option.

II. Handling Checksum Errors

A checksum is a page-by-page analysis of data to verify its integrity. A bad checksum means that a database page has been randomly overwritten (for example, due to a system crash).

Checksum errors indicate data corruption. To repair a database that reports checksum errors, select the Ignore checksum errors option. This option enables Server Manager to ignore checksums when validating a database. Ignoring checksums allows successful validation of a corrupt database, but the affected data may be lost.

Caution **Even if you can restore a mended database that reported checksum errors, the extent of data loss may be difficult to determine. If this is a concern, you may want to locate an earlier backup copy and restore the database from it.**

III. Repairing a Corrupt Database

If a database contains errors, The errors encountered are summarized in the Error Summary area on the dialog box oppened. The repair options you selected in the Database Validation dialog box will be selected in this dialog box also.

To repair the database, choose Repair. This will fix problems that cause records to be corrupt and mark corrupt structures. In subsequent operations (such as backing up), InterBase ignores the marked records.

Note: Some corruption is too serious for Server Manager to correct. These include corruption to certain strategic structures, such as space allocation pages. In addition, Server Manager cannot fix certain checksum errors that are random by nature and not specifically associated with InterBase.

If you suspect you have a corrupt database, perform the following steps:

1. Make a copy of the database using an operating-system command. Do not use the InterBase Backup utility, because it cannot back up a database containing corrupt data.
2. Repair the copy database to mark corrupt structures. If Server Manager reports any checksum errors, validate and repair the database again, choosing the Ignore checksum errors option. It may be necessary to validate a database multiple times to correct all the errors.
3. Validate the database again, with the Read-only validation option selected. Note that free pages are no longer reported, and broken records are marked as damaged. Any records marked during repair are ignored when the database is backed up.
4. Back up the mended database with Server Manager. At this point, any damaged records are lost, because they were not included during the back up. For more information about database backup, see Chapter 14: "Backing Up and Restoring a Database."
5. Restore the database to rebuild indexes and other database structures. The restored database should now be free of corruption.
6. Verify that restoring the database fixed the problem by validating the restored database with the Read-only validation option.

5. Backing Up and Restoring IPOMIS Database

A database *backup* saves a database to a file on a hard disk or other storage medium. To protect a database from power failure, disk crashes, or other potential data loss, you should regularly back up the database. For additional safety, it is recommended to store the backup medium in a different physical location from the database server.

A database *restore* re-creates a database from a backup file.

a. Using the Backup and Restore Utilities

Operating systems usually include facilities to archive database files. Server Manager offers several advantages over such facilities, including:

- Database performance can be improved. Backing up and restoring a database garbage-collects outdated records and balances indexes. The process also frees space occupied by deleted records and packs the remaining data, reducing database size. When you restore, you have the option of changing the database page size or distributing the database among multiple files or disks.
- Backups can run concurrently with other users. You need not shut down the database to run a back up. However, any data changes that occur after the back up begins are not recorded in the backup file. After you create a database backup, you can include it as part of a regular system backup.
- Data can be transferred to another operating system. Different computers have their own database file formats and therefore databases cannot simply be copied to a platform with a different operating system. If desired, you can also make a backup in a generic format called a transportable backup that allows restoration to a server on a different operating system. Making transportable backups is highly recommended in heterogeneous environments.

b. Backing Up Database

The database being backed up is referred to as the source. The file or device to which the database is being backed up is called the destination or target.

To back up a database, choose Tasks | Backup... from the Server Manager window. The Database Backup dialog box appears:

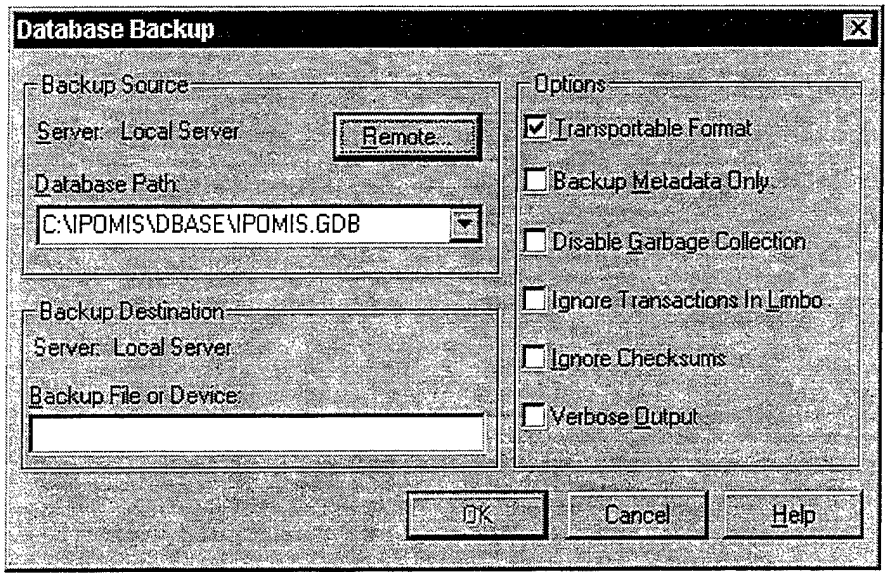


Figure 5.9 Database Backup Dialog Box

This dialog box enables you to back up a database to a file or device. To perform a backup:

1. Type the name of the source database (C:\IPOMIS\DBASE\IPOMIS.GDB) in the Database Path text field in the Backup Source area. By default, the database to which Server Manager is currently connected is displayed.
2. Type the name of the destination file or device in the text field in the lower left of the dialog box.
3. Select the desired backup options, then choose OK to start the backup timer.
4. Server Manager will open a standard text display window to display status and any messages during the backup process.

Note: Database files and backup files can have any name that is legal on the operating system; the .GDB and .GBK file extensions are InterBase conventions only.

When creating a backup file, Server Manager stores the database as one file. A backup file will typically occupy less space than the database because it includes only the cur-rent version of data and incurs less overhead for data storage.

If you specify a backup file that already exists, Server Manager overwrites it. To avoid overwriting, specify a unique name for the backup file.

I. Backup Options

The backup options are indicated by check boxes on the right side of the Database Backup dialog box. If a check box has an "X" inside, then the option is selected. If the box is empty, the option is not selected.

Transportable Format: To move a database to a machine with a different operating system from the machine on which the backup was performed, check the Transportable Format option. This option writes data in a generic format, enabling you to restore to any machine that supports InterBase. To make a transportable backup:

1. Back up the database using transportable format by selecting the Transportable Format option in the Database Backup dialog box.
2. If you backed up to a removable medium, proceed to Step 3. If you created a backup file, use operating-system commands to copy the file to tape, then load the contents of the tape onto another machine. Or copy it across a network to the other machine.
3. On the destination machine, restore the backup file. If restoring from a removable medium, such as tape, specify the device name instead of the backup file.

Back Up Metadata Only: When backing up a database, you can exclude its data, saving only its metadata. You might want to do this to:

- Retain a record of the metadata before it is modified.
- Create an empty copy of the database. The copy will have the same metadata but can be populated with different data.

To back up metadata only, select the Back Up Metadata Only option.

You can also extract a database's metadata using Windows ISQL. ISQL produces an SQL data definition (text) file containing SQL commands. Server Manager creates a backup file containing metadata only.

Disable Garbage Collection: By default, Server Manager performs garbage collection during backup. To pre-vent garbage collection during a backup, select the Disable Garbage Collection option. Garbage collection physically erases old versions of records from disk. Generally, you will want Server Manager to perform garbage collection during backup. You might not want to perform garbage collection during backup if there is data corruption in old record versions and you want to prevent InterBase from visiting those records during a backup.

Ignore Transactions in Limbo: To ignore limbo transactions during backup, select the Ignore Transactions in Limbo option.

When Server Manager ignores limbo transactions during backup, it ignores all record versions created by any limbo transaction, finds the most recently committed version of a record, and backs up that version.

Limbo transactions are usually caused by the failure of a two-phase commit. They can also exist due to system failure or when a single-database transaction is prepared.

Before backing up a database that contains limbo transactions, it is a good idea to perform transaction recovery, by choosing Maintenance | Transaction Recovery...

Ignore Checksums: To ignore checksums during backup, select the Ignore Checksums option.

A checksum is a page-by-page analysis of data to verify its integrity. A bad checksum means that a data page has been randomly overwritten; for example, due to a system crash.

Checksum errors indicate data corruption, and InterBase normally prevents you from backing up a database if bad checksums are detected. Examine the data the next time you restore the database.

Verbose Output: To monitor the backup process as it runs, select the Verbose Output option. This option opens a standard text display window to display status messages on the screen.

By default, the backup window displays the time that the backup process starts, the time it ends, and any error messages.

The standard text display window enables you to search for specific text, save the text to a file, and print the text. For an explanation of how to use the standard text display window, see on-line Help.

c. Restoring Database

To restore a database, choose Tasks | Restore... in the Server Manager window. The Database Restore dialog box will then appear:

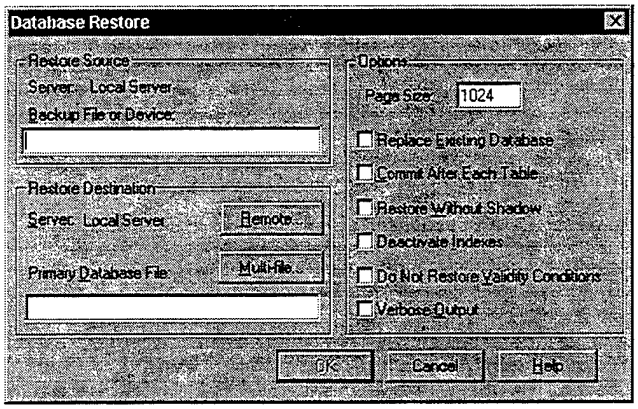


Figure 5.10 Database Restore Dialog Box

This dialog box enables you to restore a database from a previously created backup file on the current server.

The backup file from which the database is being restored is called the source. The database being restored is called the destination or target.

To restore the database:

- Type the name of the source file or device on the current server in the Backup File or Device text field.
- To restore a database to more than one database file, click on the Multi-file button.... For more information about restoring to multiple database files, see "Restoring to Multiple Files," in this section.
- Type the name (including directory path) of the database to restore to in the Primary Database File text field.
- Type the page on which to start the restore in the Start Page field, and the page size, in bytes, in the Page Size text field. Typically, the starting page will be zero (0).
- Select the desired restore options, and choose OK to begin the restore.

Typically, a restored database occupies less disk space than it did before being backed up, but disk space requirements could change if the on-disk structure version changes. For information about the ODS, see "Upgrading to a New On-disk Structure," in this section.

I. Restoring to Multiple Files

You might want to restore a database to multiple files to distribute it among different disks, which provides more flexibility in allocating system resources.

To restore a database to multiple database files, click on the Multi-file button in the Database Restore dialog box. The following dialog box opens:

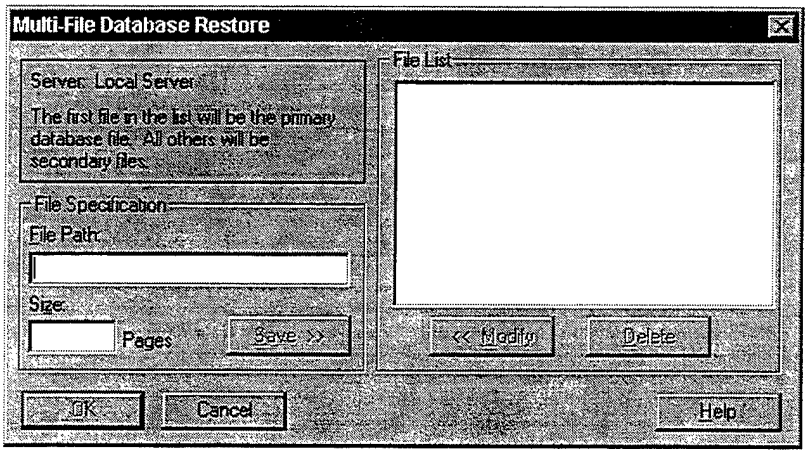


Figure 5.11 Multi-File Database Restore Dialog Box

To specify the database files to restore to, type the file name in the File Path text field and then type the number of pages for that file in the text field below it. The minimum number of pages in a file is 200. Choose Save, and the file name will appear in the File List on the right side of the dialog box.

To modify one of the files in the list, select it and choose Modify. The selected file name will appear in the File Path text field, where you can edit it, and the associated number of pages will appear in the Pages text field. To delete a file, select it in the File List and choose the Delete button.

After entering all the names of the database files to restore to, choose OK to return to the Database Restore dialog box.

II. Restore Options

The restore options are shown in check boxes on the right side of the Database Restore dialog box. If a check box has an "X" inside, then the option is selected. If the box is empty, the option is not selected.

Start Page: The Start Page is the page on which to start the restore. In most cases, this should be left as the default, zero.

Page Size: InterBase supports database page sizes of 1024, 2048, 4096, and 8192 bytes. The default is 1024 bytes. To change page size, back up the database and then restore it, modifying the Page Size field in the Database Restore dialog box.

Changing the page size can improve performance for the following reasons:

1. Storing and retrieving BLOB data is most efficient when the entire BLOB fits on a single database page. If an application stores many BLOBs exceeding 1K, using a larger page size reduces the time for accessing BLOB data.
2. InterBase performs better if rows do not span pages. If a database contains long rows of data, consider increasing the page size.
3. If a database has a large index, increasing the database page size reduces the number of levels in the index hierarchy. Indexes work faster if their depth is kept to a minimum. Choose Tasks | Database Statistics to display index statistics, and consider increasing the page size if index depth is greater than two on any frequently used index.
4. If most transactions involve only a few rows of data, a smaller page size may be appropriate, because less data needs to be passed back and forth and less memory is used by the disk cache.

Replace Existing Database: Server Manager will not overwrite an existing database file unless the Replace Existing Database option is selected. If you attempt to restore to an existing data-base name, and this option is not selected, the restore will fail.

Caution **Replacing an existing database is discouraged. When restoring to an existing file name, a safer approach is to rename the existing database file, restore the database, then drop or archive the old database as needed.**

Commit After Each Table: Normally, Server Manager restores all metadata before restoring any data. If you select the Commit After Each Table option,

Server Manager restores the metadata and data for each table together, committing one table at a time.

This option is useful when you are having trouble restoring a backup file; for example, if the data is corrupt or invalid according to integrity constraints.

If you have a problem backup file, restoring the database one table at a time lets you recover some of the data intact. You can restore only the tables that precede the bad data; restoration fails the moment it encounters bad data.

Deactivate Indexes: Normally, InterBase rebuilds indexes when a database is restored. If the data-base contained duplicate values in a unique index when it was backed up, restoration will fail. Duplicate values can be introduced into a database if indexes were temporarily made inactive (for example, to allow insertion of many records or to rebalance an index).

To enable restoration to succeed in this case, select the Deactivate Indexes option. This makes indexes inactive and prevents them from rebuilding. Then eliminate the duplicate index values, and re-activate indexes through ALTER INDEX in Windows ISQL.

A unique index cannot be activated using the ALTER INDEX statement; a unique index must be dropped and then created again. For more information about activating indexes, see the Language Reference.

Note: The Deactivate Indexes option is also useful for bringing a database online more quickly. Data access will be slower until indexes are rebuilt, but the database is available. After the database is restored, users can access it while indexes are reactivated.

Do Not Restore Validity Conditions: If you redefine validity constraints in a database where data is already entered, your data might no longer satisfy the validity constraints. You might not discover this until you try to restore the database, at which time an error message about invalid data appears.

Caution **Always make a copy of metadata before redefining it; for example, by extracting it using Windows ISQL.**

To restore a database that contains invalid data, select the Do Not Restore Validity Conditions option. This option deletes validity constraints from the metadata. After the database is restored, change the data to make it valid according to the new integrity constraints. Then add back the constraints that were deleted.

This option is also useful if you plan to redefine the validity conditions after restoring the database. If you do so, thoroughly test the data after redefining any validity constraints.

Verbose Output: To monitor the restore process as it runs, select the Verbose Output option. This option will open a standard text display window to display status messages on the screen.

The standard text display window enables you to search for specific text, save the text to a file, and print the text.

Upgrading to a New On-disk Structure: New major releases of the InterBase server often contain changes to the on-disk structure (ODS). If the ODS has changed, and you want to take advantage of any new InterBase features, upgrade your databases to the new ODS.

You need not upgrade databases to use a new version of InterBase. The new versions can still access databases created with a previous version, but cannot take advantage of any new InterBase features.

To upgrade existing databases to a new ODS, perform the following steps:

1. Before installing the new version of InterBase, back up databases using the old version.
2. Install the new version of the InterBase server as described in Installing and Running InterBase for the platform.

3. Once the new version is installed, restore the databases with the new version of InterBase.

The restored databases will be able to use any new InterBase server features.

Part 2. Using Windows ISQL

This section describes how to use Windows ISQL, InterBase's interactive SQL tool. Windows ISQL is part of the Local InterBase Server package that can be used to define, query, and manipulate data on InterBase servers.

1. Starting and Exiting Windows ISQL

To start Windows ISQL, double-click on the Windows ISQL icon  in the IPOMIS program group. The ISQL window will open:

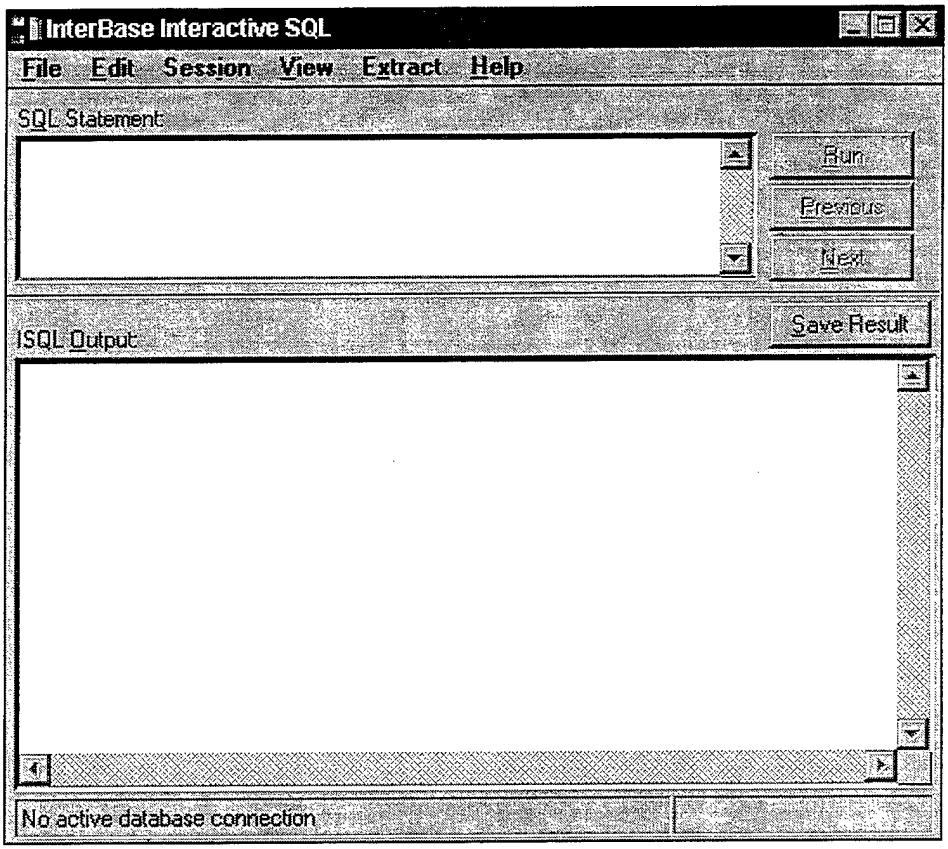


Figure 5.12 Windows ISQL Main Window

The ISQL window can also be opened from the Server Manager by choosing Tasks | Interactive SQL or clicking on the corresponding Speedbar button. Windows ISQL will then be connected to Server Manager's current database (if any).

a. The ISQL Window

The Interactive SQL window consists of a menu bar with pull-down menus, the SQL Statement area, the ISQL Output area, control buttons, and a status bar at the bottom of the window.

The ISQL menus are:

- File menu—contains commands to connect to, create, drop, and disconnect from a database, execute an SQL script file, save results and the session to a file, commit and roll back work, and exit ISQL.
- Session menu—contains statements to set basic and advanced ISQL settings, and display ISQL settings and version.
- View menu—contains a command to view metadata.
- Extract menu—contains commands to extract metadata for databases, tables, and views.
- Help menu—provides on-line help.

The SQL Statement area is where you type an SQL statement to be executed. It scrolls vertically.

The ISQL Output area is where the results of the SQL statements are displayed. It scrolls both vertically and horizontally.

The three buttons to the right of the SQL Statement area, Run, Previous, and Next, are used to execute SQL statements interactively and select statements in the SQL command history. For more information about using these buttons, see "Executing SQL Interactively," in this section. The button above the ISQL Out-put area labeled Save Result opens a dialog box in which you can enter a file name to which to save the results of the last SQL statement executed.

The status bar at the bottom of the ISQL window shows the name of the database to which Windows ISQL is connected or "No active database connection" if it is not connected to a database.

To use Windows ISQL on system's database, you must connect to IPOMIS database.

b. Getting Help

Windows ISQL provides a full online help system. Choose one of the items on the Help menu or click on a Help button in a dialog box to get help.

c. Exiting Windows ISQL

To exit Windows ISQL, choose File | Exit. This will close the connection to the current database (if any) and exit Windows ISQL. Any uncommitted changes to the database will be rolled back.

2. Connecting to a Database

Choose File | Connect to Database... to connect to an existing database. If Windows ISQL is currently connected to a database, the connection will be closed; a dialog box will prompt you to commit changes to it (if there are any). If you choose No, then all database changes since the last commit will be rolled back and the connection will be closed. If you choose Yes, then database changes will be committed. Then the Database Connect dialog box will open:

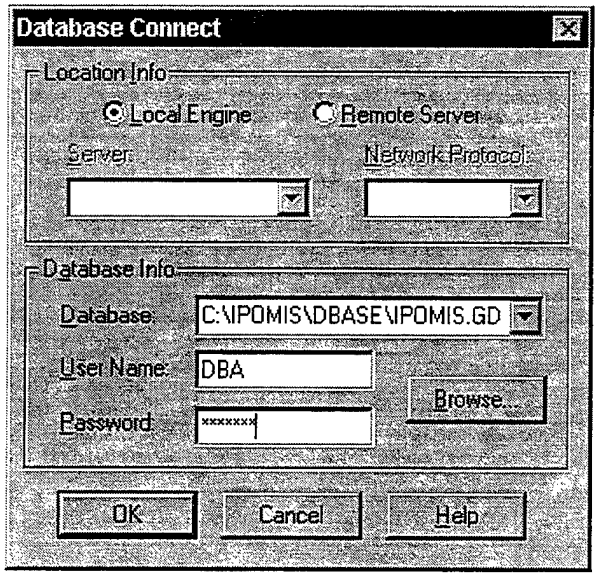


Figure 5.13 Database Connect Dialog

Select the "Local Server" by clicking on the radio button on the screen. In the Database text field, enter the name of the database "C:\IPOMIS\DBASE\IPOMIS.GDB" (including full volume and directory path), or click on the drop-down list and select IPOMIS database from the list. The User Name and Password text fields must be filled out. A null User Name with a null Password is not considered valid for IPOMIS database.

a. Dropping a Database

Dropping a database deletes the database to which ISQL is currently connected, removing both data and metadata.

To drop the current database, choose File | Drop Database.... A dialog box will ask you to confirm that you want to delete the database. A database can be dropped only by its creator (DBA) or the SYSDBA user.

A dropped database is removed from the list of databases maintained in INTERBAS.INI.

Caution **Dropping a database deletes all data and metadata in the database.** ***Do not drop IPOMIS database unless it is really required.***

b. Disconnecting From a Database

To disconnect from the database to which Windows ISQL is connected, choose File | Disconnect from Database.... A dialog box will open to confirm that you want to disconnect. If there are any uncommitted database changes, you will be prompted to commit them before disconnecting.

3. Executing SQL Statements

In Windows ISQL, you can execute SQL statements:

- Interactively, one statement at a time.
- From a file containing an SQL script.

a. Executing SQL Interactively

To execute an SQL statement interactively, type it in the SQL Statement area and choose Run or press **Alt+U**. The statement will be echoed, and up to 32K of the results displayed in the ISQL Output area. Any output beyond 32K will be scrolled out of the ISQL Output Area.

> **Tip** You can copy text from other Windows applications (such as the Notepad text editor) and paste it into the SQL Statement area with **Ctrl+V**. You can also copy statements from the ISQL Output area by highlighting them and pressing **Ctrl+C**. You can then paste them into the SQL Statement area with **Ctrl+V**.

When an SQL statement is executed (whether successfully or not), it becomes part of the ISQL command history, a sequential list of SQL statements entered in the current session. The current statement is the statement displayed in the SQL Statement area.

The three buttons to the right of the SQL Statement area are:

- Run: executes the current statement. The resultant output is displayed in the ISQL Output area. This button is dimmed if there is no active data-base connection.

- Previous: recalls the previous SQL statement in the command history, making it the current statement. When the current statement is the first statement in the command history, this button is dimmed and you may not choose it.
- Next: recalls the next SQL statement in the command history, making it the current statement. When the current statement is the last statement in the command history, this button is dimmed and you may not choose it.

As an alternative to these buttons, use the hot keys **Alt+R, Alt+P,** and **Alt+N,** respectively. The hot key for each button is underlined in its label.

b. Executing an ISQL Script File

To execute a file containing SQL statements, choose File | Run ISQL Script.... The following dialog box will appear:

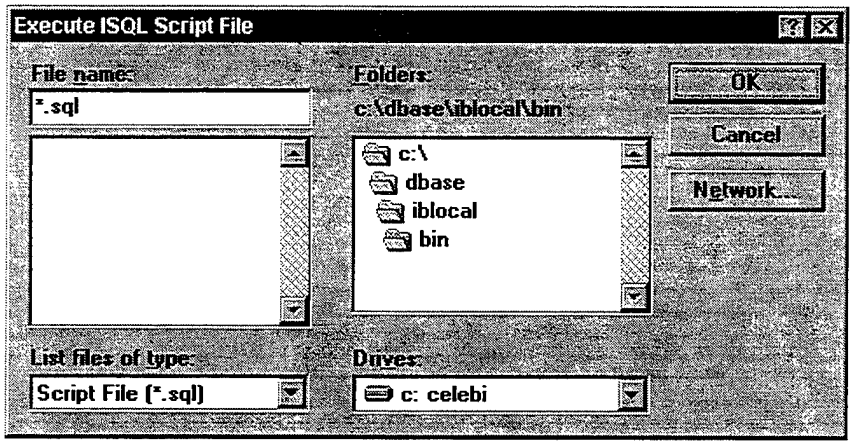


Figure 5.14 SQL Script Dialog

Enter the path and name of the file and choose OK. If you have made uncommitted changes to the database, you will prompted to commit or roll back the work. Then, a dialog box will appear asking "Save Output to a File?" If you choose Yes, then another dialog box will appear enabling you to specify an output file. If you choose No, then the results will then be displayed in the ISQL Output area. If you choose Cancel, then the operation is canceled.

After Windows ISQL finishes executing a script file, a summary dialog will appear indicating if there were any errors. If there were errors, then an error message will appear in the ISQL Output Area (or output file) after each statement that caused the error. Every ISQL script file must begin with either a CREATE DATABASE statement or a CONNECT statement (including user name and password) to specify the database on which the script file operates. For more information, see on-line help about "Using ISQL Script Files."

Statements executed in a script file do not become part of the command history.

c. Committing and Rolling Back Work

Changes to the database from data definition (DDL) statements—for example, CREATE and ALTER statements—are automatically committed by default. To turn off automatic commit of DDL, choose Session | Basic ISQL Settings... and click off the Auto Commit DDL check box.

Changes made to the database by data manipulation (DML) statements—for example INSERT and UPDATE—are not permanent until they are committed. Commit changes by choosing File | Commit Work.

To undo all database changes from DML statements since the last commit, choose File | Rollback Work.

4. Saving Results to a File

Windows ISQL enables you to save to a file:

- The output of the last SQL statement executed.
- SQL statements entered in the current session.

a. Saving ISQL Output

To save to a file the results of the last SQL statement executed, choose File | Save Result to File... or click on the Save Result button in the ISQL window. You can also use the hot key **Alt+R.**

Select the desired directory and file name or type the file name in the text field, and choose OK on the dialog box. The output from the last successful statement and the statement itself will be saved to the named text file.

If you run an SQL script, and then choose File | Save Result to File..., then all the commands in the script file and their results will be saved to the output file. If command display has been turned off in a script with SET ECHO OFF, then SQL statements in the script will not be saved to the file.

b. Saving the Session

To save the SQL statements entered in the current session to a text file, choose File|Save Session to a File....

Select the desired directory and file name or type the file name, and choose OK to save the SQL statements to the file on the dialog box.

Only the SQL statements entered in the current session, not the output, will be saved to the specified file.

5. Extracting Metadata

Windows ISQL enables you to extract metadata for the entire database and for a specific table or view.

a. Extracting Database Metadata

To extract data definition statements (metadata) from a database to a file, choose Extract | SQL Metadata for Database.... Save Output dialog box will open. If you choose Yes, then another dialog box will open, enabling you to enter the name of the file to which to extract the metadata. If you choose No, then the metadata will be displayed to the ISQL Output area only. If you choose Cancel, then the operation will be canceled.

This command does not extract:

- Generators.

- Code of external functions or filters, because that code is not part of the database. The declarations to the database (with DECLARE EXTERNAL FUNCTION and DECLARE FILTER) are extracted.
- System tables, system views, and system triggers.

This command extracts metadata in the following order:

Metadata	Comments
Database	Extracts database with default character set and PAGE_SIZE.
Domains	
Tables	
BLOB data types and known subtypes	
NULL and default values	
PRIMARY KEY constraints	
CHECK constraints	
FOREIGN KEY constraints	Must be added after tables by ALTER TABLE to avoid tables referenced before being created.
Indexes	Only for tables extracted, except triggers from referential or unique constraints.
Views WITH CHECK OPTION	
Stored procedures	In the extracted DDL, stored procedures are shown with no body in CREATE PROCEDURE and then ALTER PROCEDURE to add the text of the procedure body.
Triggers	Does not extract triggers from CHECK constraints.
GRANTs	From RDB$USER_PRIVIEGES table

Table 5-1 Order of Metadata Extraction

b. Extracting Table Metadata

To extract metadata for a single table, choose Extract | SQL Metadata for Table.... The following dialog box will open:

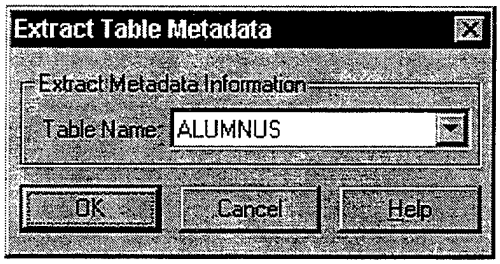


Figure 5.15 Extract Table Metadata Dialog

On the dialog box, click on the arrow to the right of the Table Name field to see a drop-down list of tables in the database. Select a table, then choose OK to extract metadata from that table.

Another dialog box will open, asking whether to save output to a file. Choose Yes to save the metadata to a text file, No to display the metadata to the Output area only, or Cancel to cancel the operation.

If there are no tables in the database, then the menu item will be dimmed, and you cannot select it.

c. Extracting View Metadata

To extract metadata for a single view, choose Extract | SQL Metadata for View.... The following dialog box will open:

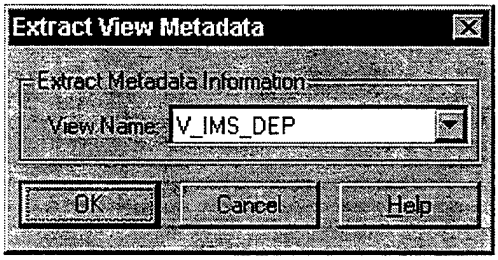


Figure 5.16 Extract View Metadata Dialog

Click on the arrow to the right of the View Name field to see a drop-down list of views in the database. Choose a view, then choose OK to extract metadata from that view.

Another dialog box will open, asking whether to save output to a file. Choose Yes to save the metadata to a text file, No to display the metadata to the Output area only, or Cancel to cancel the operation.

d. Displaying Database Information and Metadata

Choose View | Metadata Information... to display database information and metadata. The following dialog box will open:

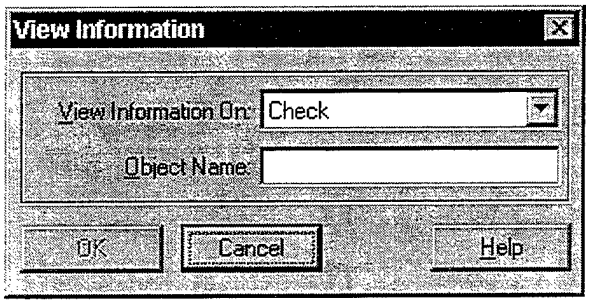


Figure 5.17 View Database Information Dialog

Select the object type for which to display information, supply any required information in the Object Name text field and choose OK. Generally, if you do not supply an Object Name, then ISQL will display the names of all objects of the selected type in the database. If you do supply an Object Name, then ISQL will display information about that object.

The following table summarizes the items that can be displayed.

Item	Displays
Check...	Check constraints for the specified table. Specify table name in the Object Name field.
Database	Current database's file name, page size and allocation, and sweep interval. Do not specify an Object Name.
Domain	Names of all domains in the database (with no Object Name). Name and data type of the domain given as Object Name.
Exception	Names of all exceptions in the database, their associated messages, and the names of triggers and stored procedures which use them (with no Object Name). Name and message of exception given as Object Name, and names of triggers and stored procedures that use it.
Generator	Names and current values of all generators in the database (with no Object Name). Name and current value of the generator given as Object Name.
Grant	Displays permissions for the table or view given as Object Name.
Index	Names of all indexes in the database, their constituent columns, and uniqueness (with no Object Name). Names of all indexes for the table given as Object Name, their constituent columns, and uniqueness. Constituent columns for the index given as Object Name, and the index's uniqueness.
Procedure	Names and dependencies of all stored procedures in the database (with no Object Name). Procedure body, for the procedure given as Object Name, its input parameters, and output parameters.
System	Displays the names of system tables and system views for the current database. Do not specify an Object Name.
Table	Names of all tables in the database (with no Object Name). Columns, data types, PRIMARY KEY, FOREIGN KEY, and CHECK constraints for the table given as Object Name.
Trigger	Names of all triggers in the database and the tables for which they are defined (with no Object Name). Trigger bodies when a table is given as Object Name. Body of the trigger given as Object Name.
View	Names of all views in the database (with no Object Name). Columns, data types, and view source for the view given as Object Name.

Table 5-2 Metadata Information Items

Part 3. Using the BDE configuration utility

The Borland Database Engine configuration utility (BDECFG.EXE) enables you to configure BDE aliases and change the settings reflecting your specific environment in the BDE configuration file, IDAPI.CFG.

To run the BDE Configuration Utility, double-click the BDE configuration utility icon in  the IPOMIS program group. The BDE Configuration Utility opens:

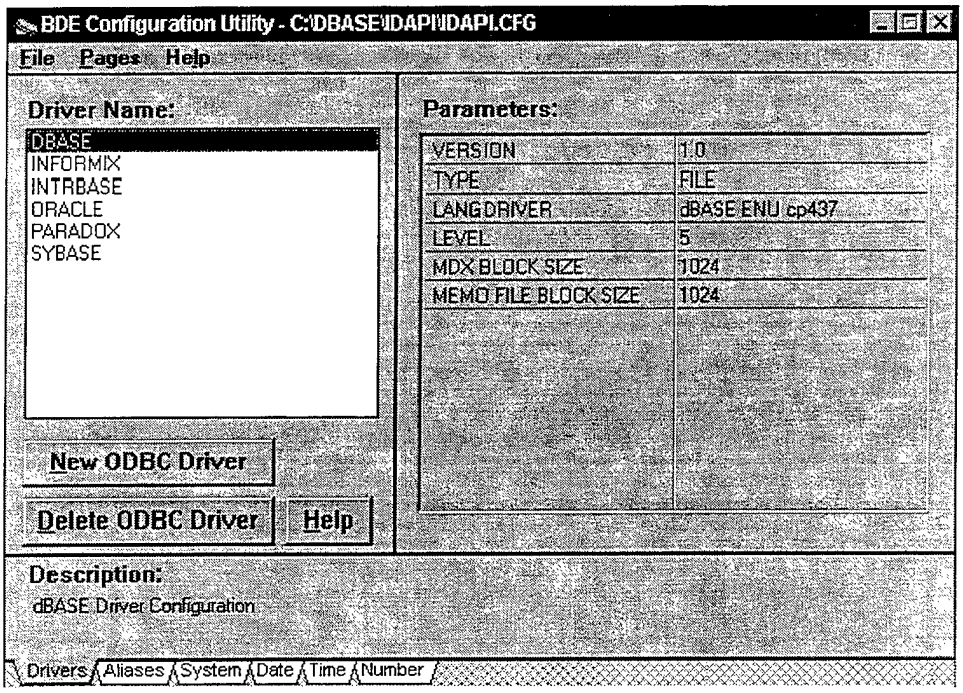


Figure 5.18 BDE Configuration Utility main window

Creating and managing aliases

IPOMIS client applications use alias named IPOMISDB which points system's database, therefore any alias pointing system's database must be named specifically IPOMISDB.

a. Adding a new alias

To add a new alias,

1. Select the Alias Manager (Aliases page) and choose the New Alias button. The Add New Alias dialog box appears (see Figure 5.19) . The type can be STANDARD or SQL-specific. For IPOMIS type must be INTRBASE.

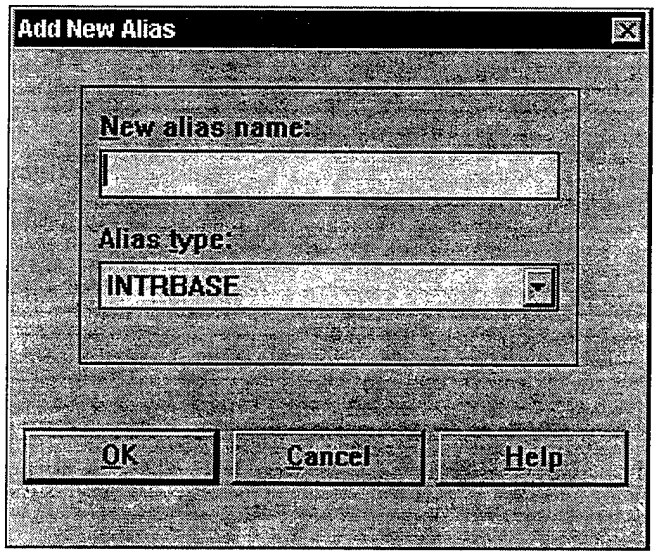


Figure 5.19 Sample Add New Alias dialog box

2. Enter the new alias name and select the SQL-specific alias type (INTRBASE). Then choose OK to begin the setup process. The Alias Manager displays all the configuration parameters you can change to customize the new alias.

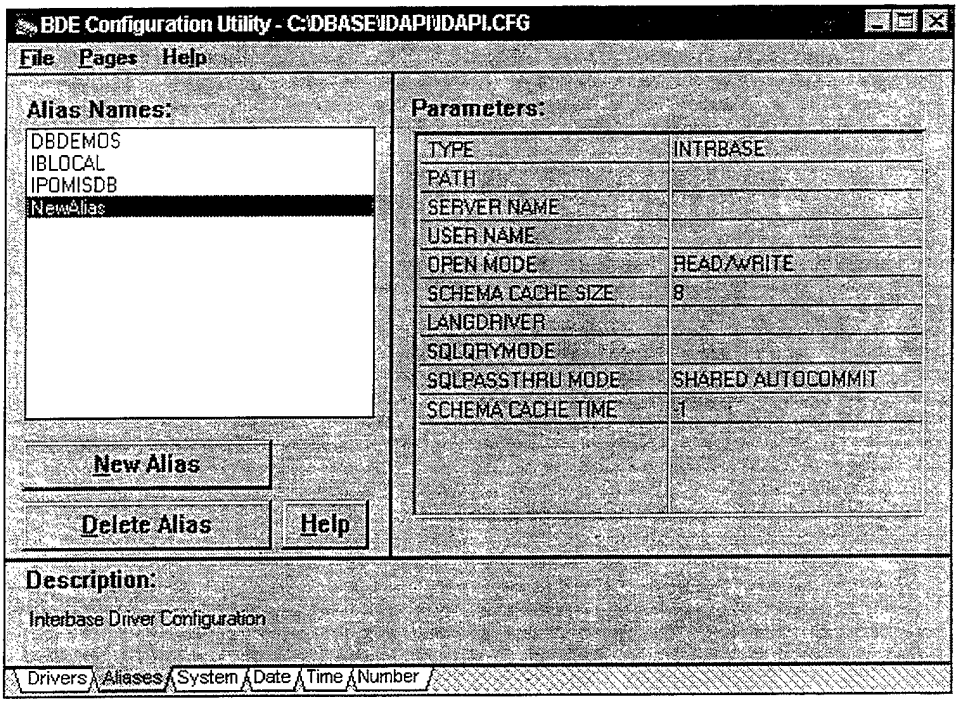


Figure 5.20 Customizing the new alias

3. If desired, edit the settings for the category you selected. If you leave any categories blank, the Alias Manager assumes you want to use the default for driver type.

4. When you are finished, select File|Save to save the new alias in the default configuration file; select File|Save as to save the new alias in a configuration file with a different name.

Note: The other pages contain settings that can also be customized. See online help for specifics.

Page	Settings modified
Driver Manager	Those BDE uses to determine how an application creates, sorts, and handles tables.
System Manager	Those BDE uses to start an application.
Date Manager	Those used to convert string values into date values.
Time Manager	Those used to convert string values into time values.
Number Manager	Those used to convert string values to number values

Table 5.3 BDE Configuration Utility Setting Pages

If you save the new alias in a configuration file with a different name, the BDE Configuration Utility displays:

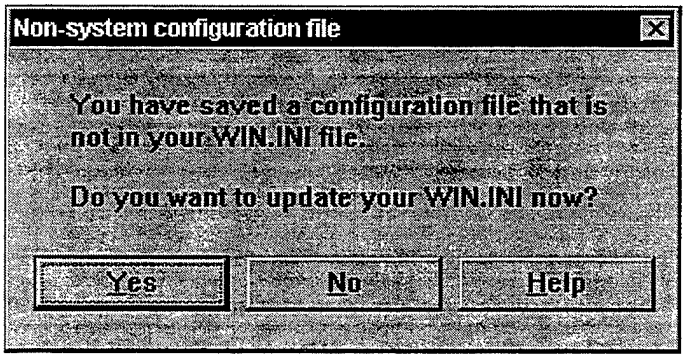


Figure 5.21 BDE non-system configuration dialog box

Choose Yes if you want to activate this configuration file next time you start your application. Choose No if you want to keep using the current default configuration file. Your changes take effect the next time you start your application.

b. Modifying an existing alias

To modify an existing alias,

1. Scan the list of Alias Names available through the current configuration file. If the alias you want to modify was stored in a different configuration file, use File|Open to load that configuration file.
2. Highlight the name of the alias you want to modify. The configuration for that alias appears in the Parameters section of the Alias Manager page.

3. Highlight the configuration parameter you want to change, and enter the desired value. If you leave any categories blank, the Alias Manager assumes you want to use the driver's default value.
4. When you are finished, select File|Save to save the new alias in the default configuration file; select File|Save As to save the new alias in a configuration file with a different name.

When you modify a driver parameter, all aliases that use the default setting for that parameter inherit the new setting.

Your changes take effect the next time you start your application.

c. Deleting an alias

To delete an alias,

1. Scan the list of Alias Names available through the current configuration file. If the alias you want to delete was stored in a different configuration file, use File|Open to load that configuration file.
2. Highlight the name of the alias you want to modify, and select the Delete Alias button.
3. Select File|Save to save your changes in the default configuration file; select File|Save As to save your changes in a different configuration file.

Note: If an IPOMIS application attempts to use its default alias IPOMISDB that has been deleted, applications will raise an exception.

Appendix

Notes to System Developer/Maintainer

The IPOMIS applications introduced in the chapters of this manual are developed using a visual object-oriented application development tool, Borland's Delphi™ 1.0 (16 bit version). The purpose in using a visual tool was to provide the user with a prototype as quickly as possible and make developer/maintainer's job easier.

This type of tool allows the developers to build highly sophisticated applications with very little code writing. Delphi provides the developer with a library of pre-packaged modules that can be visually combined into complete applications. The real power of these tools stems from this library of components that allow an application to be assembled with connections to databases, video, imaging, and messaging.

1. Software requirements

- **Borland's Delphi™ 1.0**
- **Copy of IPOMIS Database:** IPOMIS.GDB file.
- **Source code of the applications:** A copy of the source code is provided with the IPOMIS diskette set. Check the floppy labeled "IPOMIS Source Code".
- **QuickReport for Delphi version 0.95b** Component: The QuickReport package version 0.95b can be freely used in your programs. You can also distribute QuickReport as long as you do not make any changes to the original zip file. See the file MANUAL.WRI for a complete documentation on how to use QuickReport. A copy of this component is provided with the IPOMIS diskette set. Check the floppy labeled "IPOMIS Source Code".

2. Hardware requirements

Minimum requirements for the development environment:

- Intel 386-based PC

- Microsoft Windows 3.1 or later, 100% compatible version
- 6Mb RAM or higher
- At least 30Mb disk space

IPOMIS was developed on an Intel 486(DX2 66Mhz)-based PC running Microsoft's Windows 3.1 and Win95 with 16Mb RAM and plenty of disk space.

3. Documentation

The development documentation was presented as my master's thesis. These are:

- Hard copy source code
- Data model (semantic object model)
- SQL scripts used to create the IPOMIS database
- Process model (Composition and Data flow diagrams)

APPENDIX E. IPOMIS DATABASE DDL SCRIPTS

```
CREATE DATABASE "IPOMIS.GDB" PAGE_SIZE 1024 ;
```

Domain Definitions

```
CREATE DOMAIN  ACTIVITYNAME AS VARCHAR(50) NOT NULL;
CREATE DOMAIN  ACTIVITYSTATUS AS CHAR(1)
       CHECK(VALUE IN ("S","C","W","X","D","P","N") );
CREATE DOMAIN  ACTIVITYTYPE AS VARCHAR(7)
       CHECK ( VALUE IN ("STUDENT","ADULT","FAMILY") );
CREATE DOMAIN  AREACODE AS CHAR(3)  DEFAULT '408';
CREATE DOMAIN  BUSINESSNAME AS VARCHAR(35) NOT NULL;
CREATE DOMAIN  CITY AS VARCHAR(25);
CREATE DOMAIN  COUNTRYCODE AS CHAR(2) NOT NULL;
CREATE DOMAIN  CURRNUMBER AS CHAR(3) NOT NULL;
CREATE DOMAIN  DEPSTATUS AS VARCHAR(3)  DEFAULT 'YES'
       CHECK ( VALUE IN ("YES ","NO","NO ") );
CREATE DOMAIN  EMAIL AS VARCHAR(35);
CREATE DOMAIN  FIRSTNAME AS VARCHAR(20) NOT NULL;
CREATE DOMAIN  IPCLASS AS VARCHAR(3)  DEFAULT 'NO'
       CHECK ( VALUE IN ("YES ","NO","NO ") );
CREATE DOMAIN  IPSTATUS AS CHAR(1)  DEFAULT '0'
       CHECK ( VALUE IN ("0","1","2") );
CREATE DOMAIN  LASTNAME AS VARCHAR(35) NOT NULL;
CREATE DOMAIN  MARITALSTATUS AS CHAR(1)
       CHECK ( VALUE IN ("M","S","G") );
CREATE DOMAIN  MEMO AS VARCHAR(300);
CREATE DOMAIN  MIDNAME AS VARCHAR(20);
CREATE DOMAIN  NOOFPLACES AS SMALLINT DEFAULT 1;
CREATE DOMAIN  PHONENUMBER AS CHAR(8);
CREATE DOMAIN  RELATION AS CHAR(1)
       CHECK ( VALUE IN ("W","H","S","D","O") ) NOT NULL;
CREATE DOMAIN  SCN AS CHAR(9) NOT NULL;
CREATE DOMAIN  SERVICE AS VARCHAR(2)
       CHECK ( VALUE IN ("N","M","A","F","C", "CG", "NG", "N ", "M ", "A ","F ","C ") );
CREATE DOMAIN  SEX AS CHAR(1)  DEFAULT 'M'
       CHECK ( VALUE IN ("M","F") );
CREATE DOMAIN  SGC AS CHAR(4);
```

```
CREATE DOMAIN   SSN AS CHAR(9) NOT NULL;
CREATE DOMAIN   STARTDATE AS DATE NOT NULL;
CREATE DOMAIN   STATE AS CHAR(2)  DEFAULT 'CA';
CREATE DOMAIN   STREET AS VARCHAR(35);
CREATE DOMAIN   TUITIONCODE AS VARCHAR(4)  DEFAULT 'FMS'
        CHECK ( VALUE IN ("FMS ","IMET","FMS") );
CREATE DOMAIN   WCN AS VARCHAR(4);
CREATE DOMAIN   ZIPCODE AS CHAR(5);
```

Table: ALUMNUS

```
CREATE TABLE ALUMNUS (SCN SCN,
    FIRST_NAME FIRSTNAME,
    MID_NAME MIDNAME,
    LAST_NAME LASTNAME,
    HOME_ADDRESS_STREET STREET,
    HOME_ADDRESS_CITY CITY,
    HOME_ADDRESS_STATE VARCHAR(20),
    HOME_ADDRESS_ZIP VARCHAR(10),
    HOME_PHONE_AREACODE VARCHAR(5),
    HOME_PHONE_LOCAL_NUMBER VARCHAR(10),
    SERVICE SERVICE,
    RANK VARCHAR(5),
    DOR VARCHAR(5),
    TUITION_CODE TUITIONCODE,
    DOB DATE,
    ARRIVAL_DATE DATE,
    GRADUATION DATE,
    SEX SEX,
    MARITAL_STATUS MARITALSTATUS,
    WCN WCN,
    COUNTRYCODE COUNTRYCODE,
    CURRNUMBER CURRNUMBER,
    COMMENTS MEMO,
    SPOUSENAME VARCHAR(20),
PRIMARY KEY (SCN));
```

Table: COUNTRY

```
CREATE TABLE COUNTRY (COUNTRYCODE COUNTRYCODE,
    COUNTRYNAME VARCHAR(20) NOT NULL,
    COUNTRYTELCODE VARCHAR(3),
    SENIOR_OFFICER CHAR(9),
PRIMARY KEY (COUNTRYCODE));
```

Table: CURRICULUM

```
CREATE TABLE CURRICULUM (CURRNUMBER CURRNUMBER,
    CURRNAME VARCHAR(50),
    DEGREEOFFERED VARCHAR(60),
    DEPARTMENT VARCHAR(40),
PRIMARY KEY (CURRNUMBER));
```

Table: IMS

```
CREATE TABLE IMS (SCN SCN,
    FIRST_NAME FIRSTNAME,
    MID_NAME MIDNAME,
    LAST_NAME LASTNAME,
    ADDRESS_STREET STREET,
    ADDRESS_CITY CITY,
    ADDRESS_STATE STATE,
    ADDRESS_ZIP ZIPCODE,
    ADDRESS_STATUS VARCHAR(9),
    ADDRESS_AUTHORIZATION VARCHAR(3) DEFAULT 'No',
    PHONE_AREACODE AREACODE,
    PHONE_LOCAL_NUMBER PHONENUMBER,
    HOUSING_AREA VARCHAR(10),
    HOUSING_DATE_IN DATE,
    HOUSING_DATE_OUT DATE,
    HOME_ADDRESS_STREET STREET,
    HOME_ADDRESS_CITY CITY,
    HOME_ADDRESS_STATE VARCHAR(20),
    HOME_ADDRESS_ZIP VARCHAR(10),
    HOME_PHONE_AREACODE VARCHAR(5),
    HOME_PHONE_LOCAL_NUMBER VARCHAR(10),
    SERVICE SERVICE  DEFAULT 'N',
    RANK VARCHAR(5),
    DOR VARCHAR(5),
    TUITION_CODE TUITIONCODE,
    DOB DATE,
    ARRIVAL_DATE DATE,
    DATE_REPORTED DATE,
    GRADUATION DATE,
    SGC SGC,
    IT1500 IPCLASS,
    IT1600 IPCLASS,
    IP_PENALTY_STATUS IPSTATUS,
    IP_PENALTY_DATE_PENALIZED DATE,
    SEX SEX,
    MARITAL_STATUS MARITALSTATUS,
    DLI_ATTENDANCE VARCHAR(3) DEFAULT 'No',
    WCN WCN,
    COUNTRYCODE COUNTRYCODE,
    CURRNUMBER CURRNUMBER,
    SSN CHAR(9),
```

```
    EMAIL EMAIL,
    COMMENTS MEMO,
PRIMARY KEY (SCN));
```

Table: IMS_DEPENDENT

```
CREATE TABLE IMS_DEPENDENT (FIRST_NAME FIRSTNAME,
    LAST_NAME LASTNAME,
    RELATION RELATION,
    DEPSTATUS DEPSTATUS,
    DOB DATE,
    SCN SCN,
    IT1500 IPCLASS,
    IT1600 IPCLASS,
    SEC IPCLASS,
    COMMENTS MEMO,
PRIMARY KEY (FIRST_NAME, LAST_NAME, SCN));
```

Table: IMS_SPONSOR

```
CREATE TABLE IMS_SPONSOR (SCN SCN,
    SSN SSN,
PRIMARY KEY (SCN, SSN));
```

Table: IP_ACTIVITY

```
CREATE TABLE IP_ACTIVITY (ACTIVITY_NAME ACTIVITYNAME,
    START_DATE STARTDATE,
    CAPACITY SMALLINT,
    TYPE ACTIVITYTYPE,
    END_DATE DATE,
    NOOFGUESTS SMALLINT,
    NOOFESCORTS SMALLINT,
    ACTIVITY_INFO MEMO,
    OBJECTIVE MEMO,
    COMMENTS MEMO,
PRIMARY KEY (ACTIVITY_NAME, START_DATE));
```

Table: IP_ACTIVITY_ESCORT_PART

```
CREATE TABLE IP_ACTIVITY_ESCORT_PART (SSN SSN,
    NOOFPLACES NOOFPLACES,
    STATUS ACTIVITYSTATUS,
    ACTIVITY_NAME ACTIVITYNAME,
    START_DATE STARTDATE,
    DATE_APPLIED DATE,
```

```
PRIMARY KEY (SSN, ACTIVITY_NAME, START_DATE));
```

Table: IP_ACTIVITY_IMS_PART

```
CREATE TABLE IP_ACTIVITY_IMS_PART (SCN SCN,
    NOOFPLACES NOOFPLACES,
    STATUS ACTIVITYSTATUS,
    ACTIVITY_NAME ACTIVITYNAME,
    START_DATE STARTDATE,
    DATE_APPLIED DATE,
PRIMARY KEY (SCN, ACTIVITY_NAME, START_DATE));
```

Table: IP_ACTIVITY_SERVICES_USED

```
CREATE TABLE SERVICES_USED (BUSINESS_NAME BUSINESSNAME,
    ACTIVITY_NAME ACTIVITYNAME,
    START_DATE STARTDATE,
    SERVICE_TYPE VARCHAR(15) NOT NULL,
    SERVICE_DATE DATE NOT NULL,
    DESCRIPTION VARCHAR(50),
    COST NUMERIC(9, 2),
PRIMARY KEY (BUSINESS_NAME, ACTIVITY_NAME, START_DATE, SERVICE_TYPE,
SERVICE_DATE));
```

Table: SPONSOR

```
CREATE TABLE SPONSOR (SSN SSN,
    FIRST_NAME FIRSTNAME,
    MID_NAME MIDNAME,
    LAST_NAME LASTNAME,
    ADDRESS_STREET STREET,
    ADDRESS_CITY CITY,
    ADDRESS_STATE STATE,
    ADDRESS_ZIP ZIPCODE,
    PHONE_AREACODE AREACODE,
    PHONE_LOCAL_NUMBER PHONENUMBER,
    SERVICE SERVICE,
    RANK VARCHAR(5),
    OCCUPATION VARCHAR(35),
    SGC SGC,
    DOB DATE,
    MARITAL_STATUS MARITALSTATUS,
    IP_PENALTY_STATUS IPSTATUS,
    IP_PENALTY_DATE_PENALIZED DATE,
    CURRNUMBER CURRNUMBER,
```

```
    DATE_APPLIED DATE,
    DATE_ASSIGNED DATE,
    SEX SEX,
    GRADUATION DATE,
    EMAIL EMAIL,
    PREFERENCES MEMO,
    HOBBIES_INTERESTS MEMO,
    COMMENTS MEMO,
PRIMARY KEY (SSN));
```

Table: SPONSOR_DEPENDENT

```
CREATE TABLE SPONSOR_DEPENDENT (FIRST_NAME FIRSTNAME,
    LAST_NAME LASTNAME,
    RELATION RELATION,
    DOB DATE,
    SSN SSN,
    HOBBIES_INTERESTS MEMO,
    COMMENTS MEMO,
PRIMARY KEY (LAST_NAME, FIRST_NAME, SSN));
```

Table: VENDOR

```
CREATE TABLE VENDOR (BUSINESS_NAME BUSINESSNAME,
    STREET STREET,
    CITY CITY,
    STATE STATE,
    ZIP ZIPCODE,
    PHONE_AREACODE AREACODE,
    LOCAL_NUMBER PHONENUMBER,
    FAX_NUMBER PHONENUMBER,
    POC_FIRSTNAME VARCHAR(20),
    POC_LASTNAME VARCHAR(35),
    COMMENTS MEMO,
PRIMARY KEY (BUSINESS_NAME));
```

Foreign Key Definitions

```
ALTER TABLE IP_ACTIVITY_SERVICES_USED ADD FOREIGN KEY (BUSINESS_NAME)
REFERENCES VENDOR(BUSINESS_NAME);
ALTER TABLE IP_ACTIVITY_SERVICES_USED ADD FOREIGN KEY (ACTIVITY_NAME,
START_DATE) REFERENCES IP_ACTIVITY(ACTIVITY_NAME, START_DATE);
ALTER TABLE IP_ACTIVITY_IMS_PART ADD FOREIGN KEY (SCN)
REFERENCES IMS(SCN);
ALTER TABLE IP_ACTIVITY_IMS_PART ADD FOREIGN KEY (ACTIVITY_NAME,
```

START_DATE) REFERENCES IP_ACTIVITY(ACTIVITY_NAME, START_DATE);
ALTER TABLE IP_ACTIVITY_ESCORT_PART ADD FOREIGN KEY (SSN)
REFERENCES SPONSOR(SSN);
ALTER TABLE IP_ACTIVITY_ESCORT_PART ADD FOREIGN KEY (ACTIVITY_NAME,
START_DATE) REFERENCES IP_ACTIVITY(ACTIVITY_NAME, START_DATE);
ALTER TABLE SPONSOR_DEPENDENT ADD FOREIGN KEY (SSN)
REFERENCES SPONSOR(SSN);
ALTER TABLE IMS_SPONSOR ADD FOREIGN KEY (SSN)
REFERENCES SPONSOR(SSN);
ALTER TABLE IMS_SPONSOR ADD FOREIGN KEY (SCN)
REFERENCES IMS(SCN);
ALTER TABLE IMS_DEPENDENT ADD FOREIGN KEY (SCN)
REFERENCES IMS(SCN);

Granting permissions

GRANT DELETE, INSERT, SELECT, UPDATE ON ALUMNUS TO IMSASST;
GRANT SELECT ON ALUMNUS TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON COUNTRY TO IMSASST;
GRANT SELECT ON COUNTRY TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON CURRICULUM TO IMSASST;
GRANT SELECT ON CURRICULUM TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON IMS TO IMSASST;
GRANT DELETE, INSERT, SELECT, UPDATE ON IMS TO IPCOORD;
GRANT SELECT ON IMS TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON IMS_DEPENDENT TO IMSASST;
GRANT DELETE, INSERT, SELECT, UPDATE ON IMS_DEPENDENT TO IPCOORD;
GRANT SELECT ON IMS_DEPENDENT TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON IMS_SPONSOR TO IMSASST;
GRANT SELECT ON IMS_SPONSOR TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON IP_ACTIVITY TO IPCOORD;
GRANT SELECT ON IP_ACTIVITY TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON IP_ACTIVITY_ESCORT_PART TO IPCOOR
GRANT SELECT ON IP_ACTIVITY_ESCORT_PART TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON IP_ACTIVITY_IMS_PART TO IPCOORD;
GRANT SELECT ON IP_ACTIVITY_IMS_PART TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON IP_ACTIVITY_SERVICES_USED TO IPCO
GRANT SELECT ON IP_ACTIVITY_SERVICES_USED TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON SPONSOR TO IMSASST;
GRANT DELETE, INSERT, SELECT, UPDATE ON SPONSOR TO IPCOORD;
GRANT SELECT ON SPONSOR TO PUBLIC;
GRANT DELETE, INSERT, SELECT, UPDATE ON SPONSOR_DEPENDENT TO IMSASST;
GRANT SELECT ON SPONSOR_DEPENDENT TO PUBLIC;

```
GRANT DELETE, INSERT, SELECT, UPDATE ON VENDOR TO IPCOORD;
GRANT SELECT ON VENDOR TO PUBLIC
```

APPENDIX F. SAMPLE SOURCE CODE

```
program Imsinfo;
uses
  Forms,
  About in 'ABOUT.PAS' {AboutBox},
  Splash in 'SPLASH.PAS' {SplashForm},
  Alumnus in 'ALUMNUS.PAS' {frAlumnus},
  Country in 'COUNTRY.PAS' {frCountry},
  Curricul in 'CURRICUL.PAS' {frCurriculum},
  Dependen in 'DEPENDEN.PAS' {frDependent},
  Currlt in 'CURRLT.PAS' {frCurriculumList},
  Infoshet in 'INFOSHET.PAS' {frIMSinfoSheet},
  Ims in 'IMS.PAS' {frIMS},
  Seniorlt in 'SENIORLT.PAS' {frSeniorOffList},
  Pickdate in 'PICKDATE.PAS' {frPickDate},
  Reportce in 'REPORTCE.PAS' {frReportCenter},
  Imscurr in 'IMSCURR.PAS' {frIMSRosterCurriculum},
  Imsgrad in 'IMSGRAD.PAS' {frIMSRosterGraduation},
  Sproster in 'SPROSTER.PAS' {frSponsorRoster},
  Imshouse in 'IMSHOUSE.PAS' {frIMSHousing},
  Imstcoln in 'IMSTCOLN.PAS' {frIMSRosterTuitionCodeLName},
  Isgclist in 'ISGCLIST.PAS' {frSGCList},
  Sponsgcl in 'SPONSGCL.PAS' {frSponsorSGCList},
  Contotal in 'CONTOTAL.PAS' {frCountryTotals},
  Imsroste in 'IMSROSTE.PAS' {frIMSRoster},
  Seniorsp in 'SENIORSP.PAS' {frSenior},
  Imscotry in 'IMSCOTRY.PAS' {frIMSRosterCountry},
  Imstcode in 'IMSTCODE.PAS' {frIMSRosterTuitionCode},
  Statis in 'STATIS.PAS' {frStatistics};

begin
  Application.Title := 'IMS Info';
  SplashForm := TSplashForm.Create(Application);
  SplashForm.Show;
  SplashForm.Update;
  Application.CreateForm(TfrIMS, frIMS);
  SplashForm.Hide;
  SplashForm.Free;
```

```
  Application.Run;
end.


unit Ims;
interface
uses
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, StdCtrls, Forms,
DBCtrls, DB, DBGrids, DBTables, Grids, Mask, ExtCtrls, DBLookup, TabNotBk,
Buttons,    Dialogs, About, Menus, Curricul, Country, Alumnus, Dependen, InfoShet,
PickDate, Reportce;
type
  TfrIMS = class(TForm)
    ScrollBox: TScrollBox;
    Label2: TLabel;
    EditFIRST_NAME: TDBEdit;
    .
    .
    .
    procedure LNameSearchClick(Sender: TObject);
    procedure About1Click(Sender: TObject);
    procedure Country1Click(Sender: TObject);
    .
    .
    .
  private

  public

  end;

var
  frIMS: TfrIMS;

implementation

procedure TfrIMS.LNameSearchClick(Sender: TObject);
var QueryString : String; {Holds the last name filled by the user}
{Searches for the last name provided by the user through the database and locates the
nearest/exact match}
begin
   QueryString := 'Last Name';
```

```
  if InputQuery ('Search By Last Name','Enter Last Name', QueryString) then
   begin
      QueryString:= AnsiUppercase (QueryString); {Converts the string to uppercase}
      tIMS.SetKey;
      tIMS.FieldByName('LAST_NAME').AsString := QueryString;
      tIMS.GotoNearest; {Goes to the nearest match in the database}
   end;
end;
```

```
procedure TfrIMS.About1Click(Sender: TObject);
{Creates and displays the About screen}
begin
 with AboutBox.Create(Application) do
 begin
  ShowModal;
  Free; {Releases the memory allocated to the About screen}
 end;
end;

procedure TfrIMS.Country1Click(Sender: TObject);
{Creates and shows the country screen to edit/update selected IMS's country
   and senior officer info }
begin
 with TfrCountry.Create(Application) do
 begin
   tCountry.Open;
   tCountry.SetKey;
   tCountry.FieldByName('COUNTRYCODE').AsString :=
       tIMS. FieldByName('COUNTRYCODE').AsString;
   tCountry.GotoKey;
   tSeniorOfficer.Open;
   ShowModal; {Releases the memory allocated to the Country screen}
   Free;
 end;
end;
.
.
.
end. {End of IMS unit}


unit Pickdate;
interface
uses WinTypes, WinProcs, Classes, Graphics, Forms, Controls, Buttons,
 SysUtils, StdCtrls, Grids, Calendar, ExtCtrls;
type
 TfrPickDate = class(TForm)
  Calendar1: TCalendar;
  bnOK: TButton;
  bnCancel: TButton;
```

```
    TitleLabel: TLabel;
    PrevMonthBtn: TSpeedButton;
    NextMonthBtn: TSpeedButton;
    Bevel1: TBevel;
    LastYearbtn: TSpeedButton;
    NextYearBtn: TSpeedButton;
    Year: TLabel;
    Month: TLabel;

    procedure PrevMonthBtnClick(Sender: TObject);
    procedure NextMonthBtnClick(Sender: TObject);
    procedure Calendar1Change(Sender: TObject);
    procedure NextYearBtnClick(Sender: TObject);
    procedure LastYearbtnClick(Sender: TObject);

  private
    procedure SetDate(Date: TDateTime);
    function GetDate: TDateTime;

  public
    property Date: TDateTime read GetDate write SetDate;
  end;

var
  frPickDate: TfrPickDate;

implementation

procedure TfrPickDate.SetDate(Date: TDateTime);
{Sets calendar to the value of  the Date variable passed by the caller}
begin
  Calendar1.CalendarDate := Date;
end;

function TfrPickDate.GetDate: TDateTime;
{Gets and Returns the value of the calendar to the  caller}
begin
  Result := Calendar1.CalendarDate;
end;

procedure TfrPickDate.PrevMonthBtnClick(Sender: TObject);
{Sets the month shown on the calendar to previous month}
```

```
begin
  Calendar1.PrevMonth;
end;

procedure TfrPickDate.NextMonthBtnClick(Sender: TObject);
{Sets the month shown on the calendar to next month}
begin
  Calendar1.NextMonth;
end;

procedure TfrPickDate.Calendar1Change(Sender: TObject);
{Sets the title  shown on the top of  calendar to long month and date}
begin
  TitleLabel.Caption := FormatDateTime('MMMM, YYYY', Calendar1.CalendarDate);
end;

procedure TfrPickDate.NextYearBtnClick(Sender: TObject);
{Sets the year shown on the calendar to next  year}
begin
  Calendar1.NextYear;
end;

procedure TfrPickDate.LastYearbtnClick(Sender: TObject);
{Sets the year shown on the calendar to previous year}
begin
  Calendar1.PrevYear;
end;

end. {End of PickDate Unit}



unit Imsinfo;
interface
uses
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  StdCtrls, Forms,
DBCtrls, DB, DBGrids, DBTables, Grids, Mask, ExtCtrls,  DBLookup, TabNotBk,
Buttons, Dialogs, About, Menus, InfoDlg;
type
  TfrIMSInfo = class(TForm)
    ScrollBox: TScrollBox;
    Label2: TLabel;
```

```
    EditFIRST_NAME: TDBEdit;
    tCountryCOUNTRYNAME: TStringField;
    .
    .
    .
    procedure bnAddClick(Sender: TObject);
    procedure FormActivate(Sender: TObject);

  private
    { private declarations }
  public
    { public declarations }
  end;

var
  frIMSInfo: TfrIMSInfo;

implementation

procedure TfrIMSInfo.bnAddClick(Sender: TObject);
{Adds an IMS application}
begin
 if LabelActivity.Caption <> '' then
 begin
 try
 {Prepares IMS Participation table to add a new record}
  tIMSPart.Open;
  tIMSPart.Insert;
  tIMSPart.FieldByName('ACTIVITY_NAME').AsString := LabelActivity.Caption;
  tIMSPart.FieldByName('START_DATE').AsString := LabelStartDate.Caption;
  tIMSPart.FieldByName('SCN').AsString := tIMS.FieldByName('SCN').AsString;

 {Transfers required data to "Application Information" window and
  Shows the window}
  with TAppInfoDlg.Create(Application) do
  begin
   EditDateApplied.Text := DateToStr(Date);
   ActivityType := LabelType.Caption;
   ShowModal;
   {If settings (data) are confirmed then completes the information on application}
   if Confirmed then
   begin
```

```
   tIMSPart.FieldByName('NOOFPLACES').AsString :=
     IntToStr(EditNumberOfPlaces.Value);
   tIMSPart.FieldByName('DATE_APPLIED').AsString :=
     EditDateApplied.Text;
   if ((not RBnNO.Checked) or (RBnYES.Checked)) then
    tIMSPart.FieldByName('STATUS').AsString := 'P';
  end
  else
  tIMSPart.Cancel;
  Free;
 end;

tIMSPart.Refresh;
bnSearch.SetFocus;
except
 on EDbEngineError do
  if MessageDlg('Could not add this application - It has already been entered',
   mtError, [mbOK], 0) = mrOK then tIMSPart.Cancel;
 end;
end
else
begin
 ShowMessage('There is no activity selected, thus I cannot add this application')
end;
end;

procedure TfrIMSInfo.FormActivate(Sender: TObject);
{Prepares the screen on activation: Sets tab Notebook Page, Opens required tables, and
activates   last name search}
begin
tnIMS.ActivePage := 'IP Records';
try
 if not tIMS.Active then tIMS.Open;
 if not tCountry.Active then tCountry.Open;
 if not tIMSPart.Active then tIMSPart.Open;
tIMSPart.Refresh;
 if not tvIMS_DEP.Active then tvIMS_DEP.Open;
LNameSearchClick(Sender); {Calls the last name search procedure}
 except
  on EDbEngineError do
  begin
   ShowMessage('Error in database, program will be closed');
```

```
  Close
 end;
end;
end;
end.
```

LIST OF REFERENCES

Borland Delphi™ Database Application Developer's Guide, Scotts Valley, California: Borland International, Inc., 1995.

Callaghan, James T. and Holland, Harold B., *Foreign Training Officer's Guide* (Second Edition), Defense Institute of Security Assistance Management (DISAM), Wright-Patterson AFB, Dayton, Ohio, 1984.

Hodges, Nick, *A Visual Development Methodology for the Department of Defense*, Masters Thesis, Naval Postgraduate School, September 1995.

International Committee, *International Committee By-Laws and Organizing Document*, Naval Postgraduate School, 1994.

Kroenke, David M., *Database Processing: Fundamentals, Design, and Implementation*, Fifth Edition, Englewood Cliffs, New Jersey: Prentice Hall, 1995.

SECNAVINST 4950.4: Joint Security Assistance Training (JSAT) Regulation, Headquarters Department of the Army, Washington, DC, 1990.

Whitten, Jeffrey L., Bentley, Lonnie D., and Barlow, Victor M., *System Analysis and Design Methods,* Third Edition, Boston, Massachusetts: Richard D. Irwin, Inc., 1994.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Road., Ste 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, California 93943-5101	2
3.	Personel Egitim Daire Baskanligi Dz.K.K. Karargahi 06100 Bakanliklar / ANKARA TURKEY	2
4.	International Programs Office, Code 035 Naval Postgraduate School Monterey, California 93943-5108	2
5.	Professor James C. Emery, Code 05 Computer and Information Services Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor Magdi N. Kamel, Code SM/Ka Department of Systems Management Naval Postgraduate School Monterey, California 93943-5000	1
7.	Onder Celebi Ugur Mumcu mah. Ozgur Sitesi 136. Sk. No: 32 06370 Batikent/ANKARA TURKEY	1