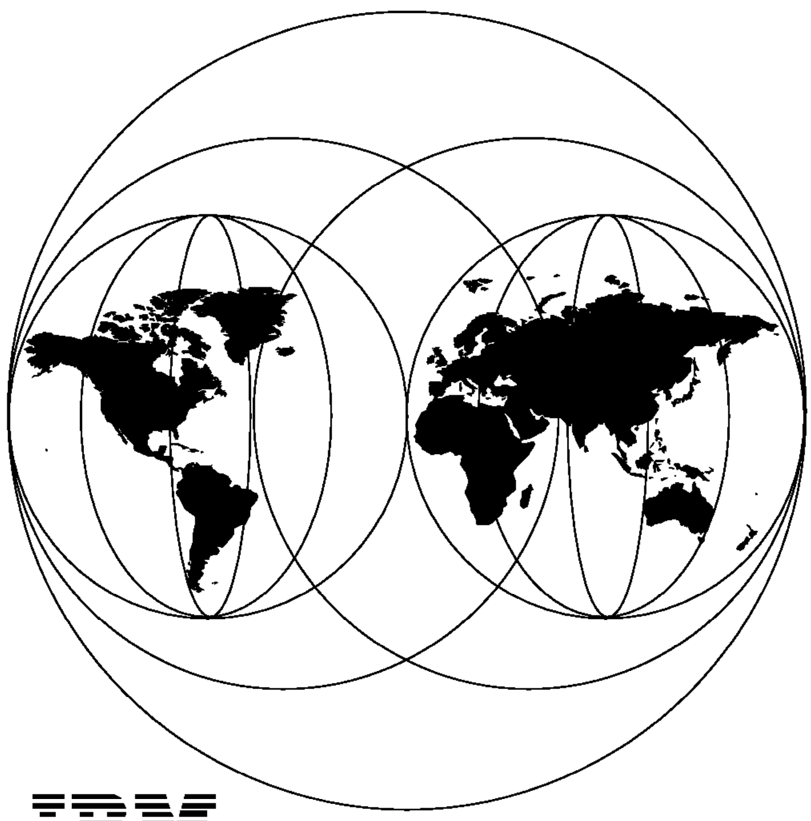
SG24-4846-00

Multi-Language Solutions for Client/Server Database Environment

December 1996



International Technical Support Organization
Boeblingen Center

International Technical Support Organization

SG24-4846-00

Multi-Language Solutions for Client/Server Database Environment

December 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, “Special Notices” on page 255.

First Edition (December 1996)

This edition applies to the DB/2 family of products for use with all related operating systems.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. 3222 Building 71032-02
Postfach 1380
71032 Böblingen, Germany

Contents

Figures

Tables

Preface

This document is unique in its detailed coverage of multi-language solutions in distributed database client/server environments. It provides comprehensive information about implementation of the DRDA connectivity across countries using different national languages, and it shows how the multi-language requirements are supported by the DB2 family of products.

It contains setup and usage information, including all related system definitions for the different platforms involved. These were DB2 for VSE and VM, DB2 for OS/2 and DB2 for AIX and their respective client/server capabilities.

This was a joint project between ITSO and ZPIZ (Institute for Pension and Disability Insurance of Slovenia) and accomplished successfully within 10 weeks of cross country teamwork.

How This Redbook Is Organized

The redbook is organized as follows:

- Chapter 1, "Introduction"

  This chapter gives a brief overview of the project environment, the purpose of the project, the benefits of the project and how the ITSO and ZPIZ environments are set up for this project. It also gives an introduction to DRDA and CDRA.

- Chapter 2, "Multi-Platform Considerations"

  This chapter provides information on how the different protocols are used among different types of connections and how multi-languages are supported by the DB2 family of products on multi-platforms.

- Chapter 3, "VM/ESA Implementation"

  This chapter describes ITSO and ZPIZ software installation and steps to establish DRDA connectivity. The tasks for implementing SQL/DS for VM acting as both DRDA Application Server (AS) and DRDA Application Requester (AR) are described.

- Chapter 4, "VSE/ESA Implementation"

  This chapter describes the most important setup tasks required to enable SQL/DS for VSE/ESA to perform DRDA AS functions. It must be pointed out that SQL/DS for VSE/ESA cannot act as a requester, as SQL/DS in the VM/ESA environment does.

- Chapter 5, "OS/2 Implementation"

  This chapter describes experiences with workstation and host setup in DRDA multi-language environments between ITSO Böblingen and ZPIZ Ljubljana. In order to establish connections, communication controllers 3745 and 3174 were used at the ITSO site and 3745/NCP at the ZPIZ site.

- Chapter 6, "AIX Implementation"

  This chapter describes the detailed steps and tasks to set up an AIX workstation acting as a DRDA AR using DB2 for AIX in a DRDA multi-language environment. DB2 for AIX also can act as a DRDA AS but this feature was not used within our project. How to access DB2 for AIX as a database server via TCP/IP is shown in 2.1.2, “TCP/IP” on page 20.

- Chapter 7, “Multi-Language Solutions”

  This chapter focuses on the customer side, and gives information about the connectivity parameters used at ZPIZ and ITSO, with related installation and customization steps.

- Chapter 8, “REXX SQL Application Enabling and Usage”

  This chapter describes specific OS/2 REXX application capabilities and interfaces to the DB2 family of products. Specifically, the interfaces to remote databases residing on VM/ESA and VSE/ESA were tested.

- Appendix A, “System Tables and Sample Programs”

  This appendix contains the system definition files used on the OS/2, VM and VSE platforms.

- Appendix B, “Customization and Displays”

  This appendix contains the system definition and parameters used on the AIX platform to set up the DRDA connections. You will find here the SNA profile definitions, the DB2 for AIX database directory entries and some information needed when connecting to a DRDA AS on VM/VSE.

- Appendix C, “Sample Code Pages”

  This appendix contains some code page samples.

The Team that Wrote this Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Böblingen Center. The project was designed and managed by:

Manfred Mittelmeier International Technical Support Organization, Böblingen Center

Janko Furman System SW Manager, ZPIZ Ljubljana

Project team members in *Böblingen* were:

Jure Lovsin	IBM Slovenija
Klaus Mairhofer	IBM Germany
Manfred Mittelmeier	ITSO Center Böblingen
Luciano Jose dos Santos	IBM Brazil
Fu Lianhai	IBM China

Project team members in Ljubljana were:

Janko Furman	ZPIZ Ljubljana
Polde Brodt	ZPIZ Business Partner

Thanks to the management of the DP department at ZPIZ Ljubljana for supporting the project:

Franc Pelko	CIO
Anton Dunato	DP Manager

Our sincere thanks also to the colleagues from the ZPIZ system software group and the IBM Laboratory Toronto for their technical support.

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Introduction

1.1 Situation

In the client/server environment, the different functions of information systems (data management, application logic, and presentation services) are separated and placed on the platforms that are regarded as the best place for these functions. There is already a prevalence of database tools that take advantage of the ease of use of the workstation, through GUIs and fourth-generation languages (4GLs), and still utilize the data and system management strength of traditional, host-based databases.

At the same time, the distribution of data itself has given rise to client/server databases that integrate very well with spreadsheets, decision support applications and tools, and word processors. Users, meanwhile, continue to require access to corporate data remaining on the host. This situation has produced various alternatives in accessing different and remote databases with acceptable performance and ease.

Gateway products provide a standard method of integrating data throughout the enterprise. A gateway shields users from the confusion of multiple and usually disparate forms and locations of the data. For some environments, meanwhile, gateways are the only option for data integration as a result of their heterogeneous hardware, software, network platforms, and standards or even languages and code pages.

In this situation, it is more than probable that the users from different locations use different languages which are in different groups. For example, Latin-1 and Latin-2 are in different language groups. However, it must be possible for them to exchange data resulting in the correct representation at both ends. At the same time, customers also wish to pilot their database into the client/server environment, thereby setting up a prototype system for further use. With these goals in mind, the ITSO Böblingen coordinated this joint project using multi-platform and multi-language database client/server solutions with ZPIZ Slovenia, an IBM customer from the eastern part of Europe.

This redbook documents all the technical aspects of this project.

1.2 Objective

To satisfy the criteria for the project, the IBM customer involved had to fulfill certain requirements regarding I/S infrastructure and be in a different language group than the ITSO.

The specific objectives of the project are:

- Connect locations exploiting available networking structure
- Provide DRDA connectivity involving VM/ESA, VSE/ESA, OS/2, AIX using various protocols
- Provide multi-language solutions for proper interpretation of data from client or server which use different character sets and code pages

- Provide techniques for data transformation to fulfill specific requirements, for example to alter the sorting sequence
- Run applications against each other′s data server
- Enable skill transfer
- Access remote data using REXX SQL applications

1.3 The Current Environment

In the Böblingen center, a distributed database environment using DB2 for OS/2, DB2 for AIX, and DB2 for VSE & VM has been built up. They have also established connectivity to remote database servers located in the U.S.A. using DRDA and TCP/IP type protocols crossing multiple networks. On the client side, various types of workstations with different customization and usage are installed.

For this project, the customer requested OS/2 and AIX clients with or without a local database. In addition, a local database server only, optionally combined with DDCS for providing DRDA connectivity was a requirement.

Figure 1 shows the implemented connectivity between the ITSO Böblingen, Germany and the customer, ZPIZ Slovenia.

On the customer side, we used cloned clients based on OS/2, a local database server with DDCS for OS/2 and DB2 for VM & VSE. The VM system was also set up for DRDA application requesters to their own client applications and servers located in Böblingen.

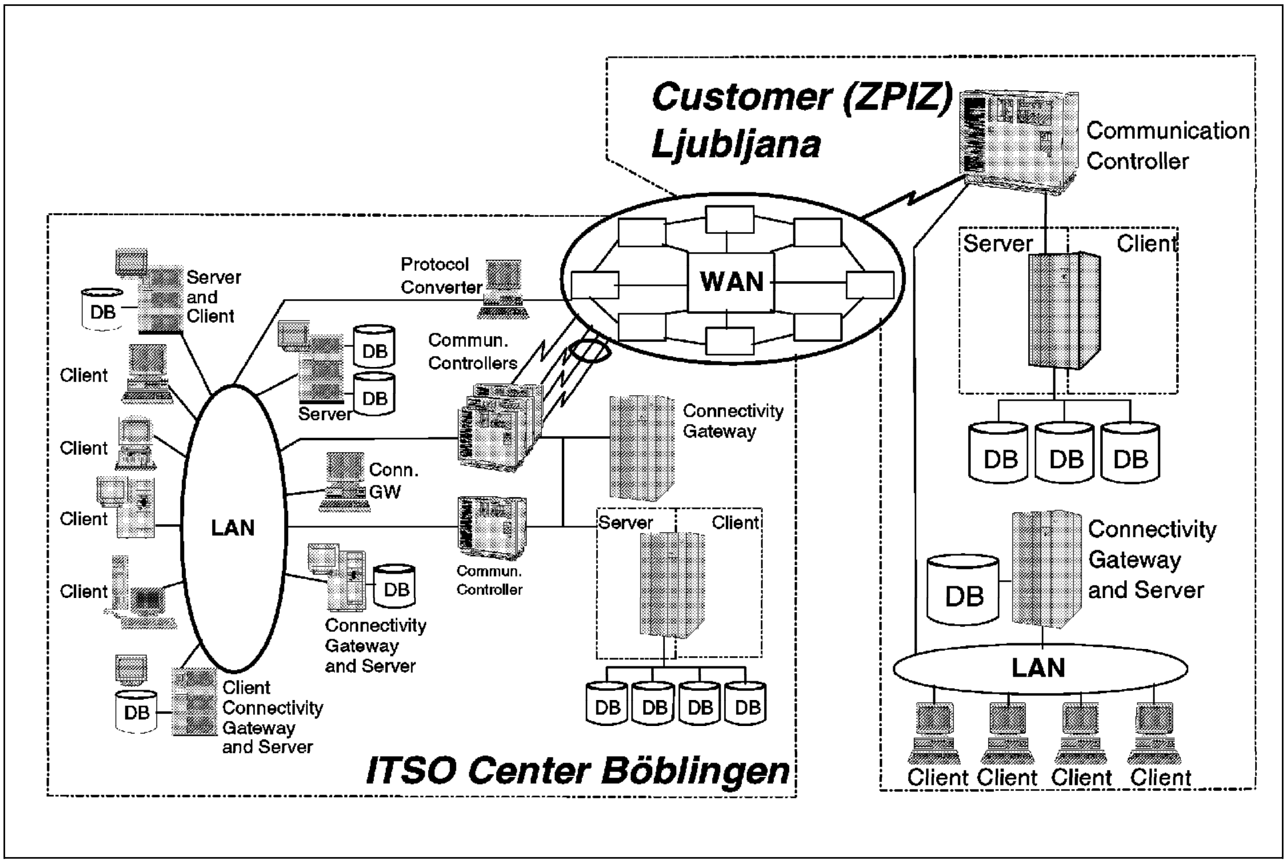


Figure 1. Network Overview

In this client/server environment, the following platforms are involved:

- VM
- VSE
- OS/2
- AIX
- DOS/Windows

Figure 1 includes almost all connectivity possibilities. ZPIZ connects to the remote ITSO Böblingen through the WAN. Locally, both ZPIZ and ITSO use the Token-Ring to set up their own network with multiple protocols, including APPC, TCP/IP and NetBIOS. For communications between DB2 for OS2 and DB2 for VM & VSE, APPC can be used. The DOS client can use NetBIOS or TCP/IP to connect to DB2 for OS2 or DB2 for AIX.

The client can access different databases at the same time, irrespective of whether the database is local or remote or on which platform the database is located. For example, the client at ZPIZ can connect to their own databases or to databases which are in the ITSO system through the WAN. This is one of the great benefits of this client/server environment, especially for application usage. This type of connectivity is transparent to the application. There is no definition required in the application programs to describe this connectivity which, for example, supports application portability.

1.4 Code Page Considerations

The customer ZPIZ requested to use database C/S solutions with countries located in the western part of Europe, belonging to language group Latin-1 and with countries located in the eastern part of Europe, belonging to language group Latin-2.

DRDA (Distributed Relational Database Architecture) provides an architected solution to connect clients and servers on different platforms, perhaps located in different countries. DRDA combines various architectures to provide transparent access to the data.

CDRA (Character Data Representation Architecture) provides automatic code page conversion if client and server are using different code pages. This support can be provided within each language group.

Germany, USA and most of the western countries belong to group 1 (Latin-1), whereas Slovenia belongs to group 1a (Latin-2). The general problem was that at the time of running this project there was **no** automatic code page translation between the different language groups. Figure 2 shows this current situation, and in Appendix C, "Sample Code Pages" on page 245 are copies of the code pages from Latin-1 and Latin-2 countries.

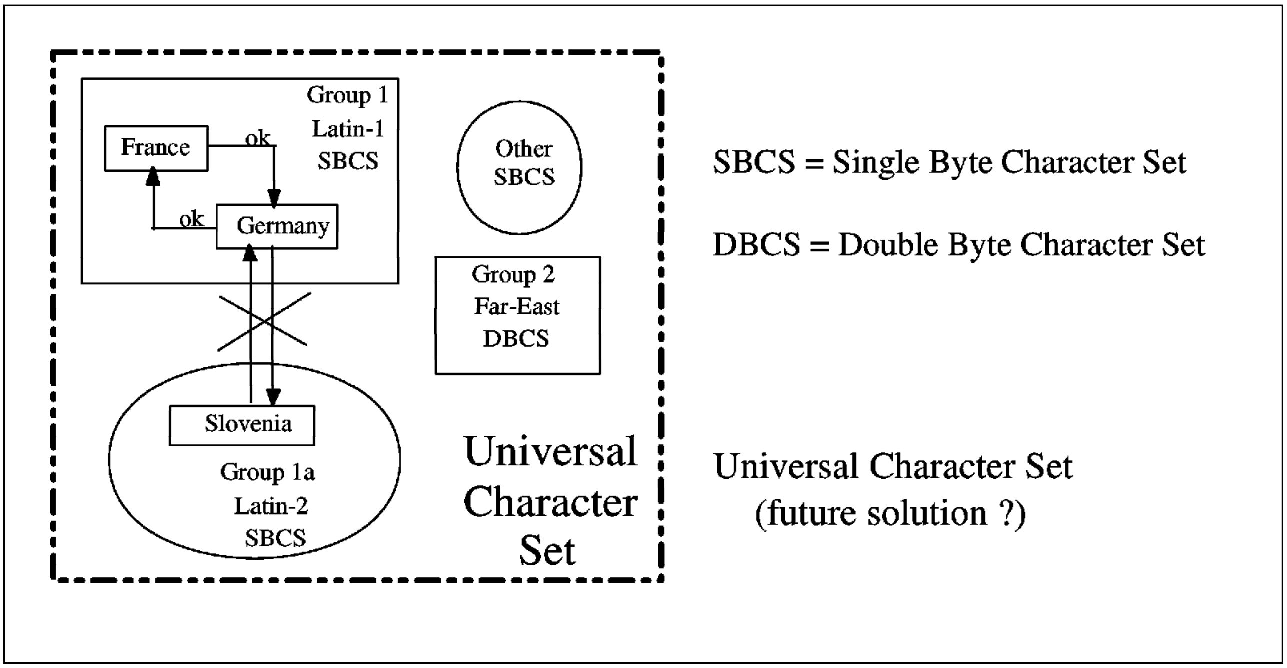


Figure 2. Current Situation

The need for inter-group code page translation is driven by the ongoing implementations of C/S solution especially in countries which are based on Latin-2. There might be also an increasing need to involve other groups, such as Asian countries which are using various code pages based on DBCS (Double Byte Character Set). To combine all groups is a known requirement. The solution for this may be a Universal Character Set! However, this is beyond the scope of this document.

With help from the IBM Lab in Toronto we were able to provide a solution for Latin-1 and Latin-2 code page conversion. This type of solution will be delivered in the next release of DB2 for Common Server V3.

In order to connect to a database with an incompatible code page, the following steps need to be performed. Although this information is correct at this time (Nov '96), you should always check your DB2 manuals first, to see if there is more up-to-date information applicable to the same topic.

1. Make sure you are using DB2 server V2.1.2 or above.
2. Ask your IBM representative to obtain two conversion files for the pair of code pages that you want to support. For example, if you want to support connection between code page 819 and code page 915, you need 08190915.cnv and 09150819.cnv tables.
3. Place these binary files in the *<DB2Path>/sqllib/conv* directory on UNIX platforms such as AIX, and in the *<DB2Path>/sqllib/conv* directory on various Intel platforms such as OS/2 or Windows.
4. If you are only using Common Server DB2 C/S products, you need to place these files only on the server.
5. For connections between Common Server DB2 C/S and mainframe DB2 implementations, such as SQL/DS or DB2/MVS, you need to first ensure that the corresponding DB2 host platform supports the same conversion. You only need the Host-to-Workstation conversion table on the workstation side.
6. Once the connection between incompatible code page environments is established, you will get a warning message:

```
SQL0863W  A successful connection was made, but only single byte
characters should be used.  SQLSTATE=0153.
```

Figure 3. Connection Information

This is normal, and intended to warn you that you should only use the characters that exist in both code pages.

1.5 More Configuration Details

1.5.1 The ITSO Environment

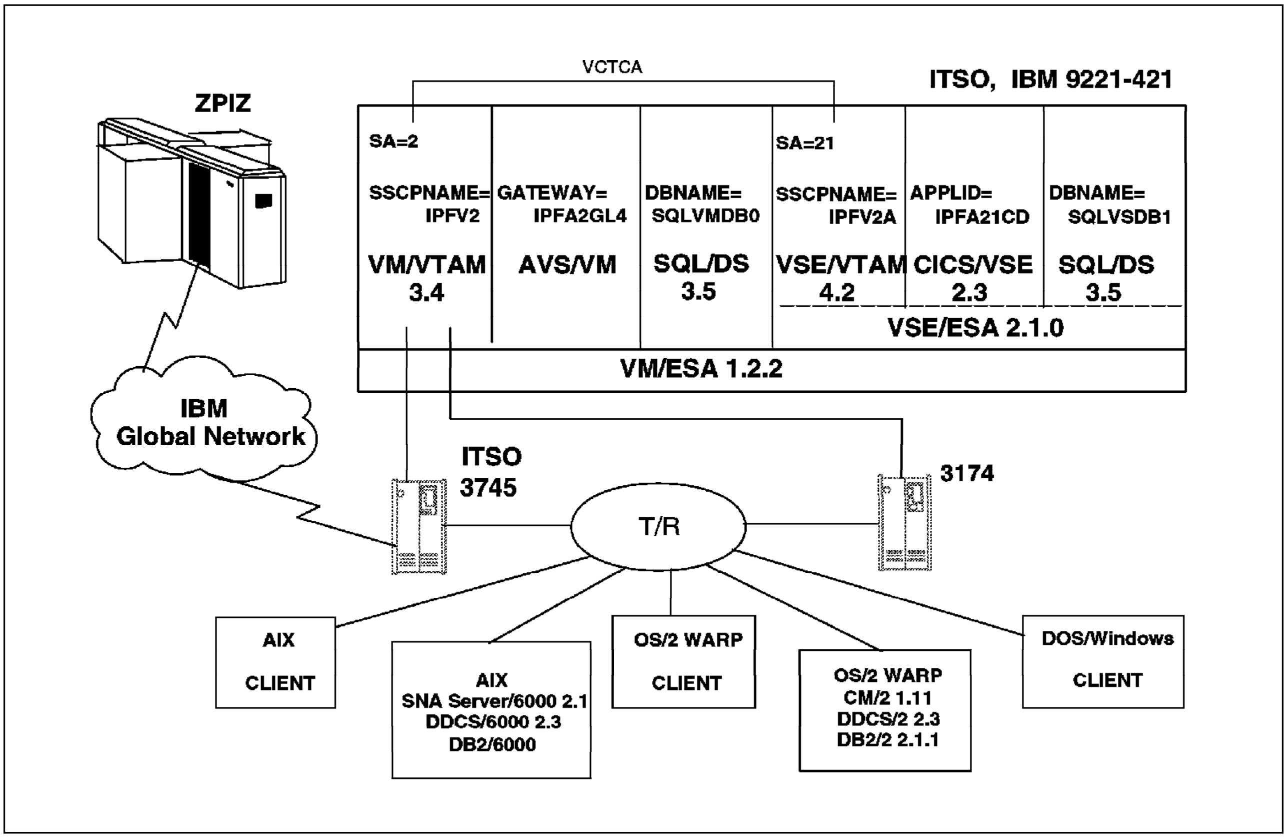


Figure 4. ITSO Connection Layout

Figure 4 shows part of the environment which is setup at the ITSO Böblingen to investigate multi-platform and multi-language DB C/S solutions. In this environment, the VSE system is installed as a guest system under a VM/ESA system. There are two VTAM systems; one installed in the VSE system and the other in the VM/ESA system. The VM/VTAM system is connected to the 3745 NCP, which is linked to the WAN to link finally to the ZPIZ site.

From a networking point of view when crossing multiple networks, (in this example, IBM Germany, IBM Austria, IBM Slovenia), it should be normal procedure to solicit the support of the people whose network you are using. Their help is invaluable with the proper definition of the involved VTAM members (for example, ADJSSCP) or authorization of your application LUs (for example IPFA2GL4), to let your session requests through to the final destination.

From the client point of view, ITSO Böblingen has two communication controllers (3745 and 3174), which are channel attached to the business server. Both controllers are attached to the LAN T/R and can be used alternatively.

1.5.2 The ZPIZ Customer Environment

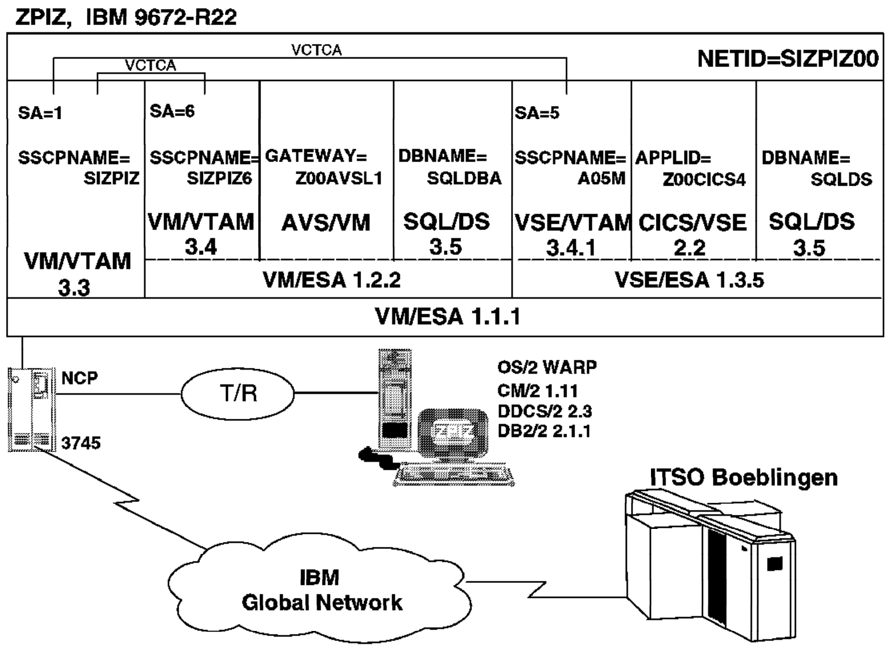


Figure 5. ZPIZ Connection Layout

Figure 5 illustrates the main software and hardware components involved in DRDA connectivity at ZPIZ.

A VM/ESA 1.2.2 system is installed as a second level system and is connected to the first level VM/ESA 1.1.1 via a Virtual Channel-to-Channel Adapter (VCTCA).

SQL/DS 3.5.0 for the VM database runs on a VM/ESA second level system.

SQL/DS 3.5.0 for VSE runs on VSE/ESA 1.3.5 which is connected via VCTCA to the VM/ESA first level system.

VM/VTAM on the VM/ESA first level system is connected via 3745 to Token-Ring and other SNA networks, such as IGN (International Global Network), and via VCTCA to the other VTAM systems.

1.5.3 A Short Description of ZPIZ

The following sections 1.5.3 to 1.5.6 were written by the customer ZPIZ.

The very beginnings of the pension and disability insurance in Slovenia go back to the end of the previous century. These were various social subventions based on mutuality and were paid out by different associations and various private, guild insurances.

Taking into consideration the fact that the Republic of Slovenia has been defined by the constitution as a social and legal state, the constitution represents, besides a general definition of the right to social security, a basis for legal regulation of social security branches. Consequently, the state regulates compulsory health, pension, disability as well as other social insurances (unemployment insurance) by the statute in accordance with a constitutional regulation and also takes care of their implementation.

The bearer and executor of pension and disability insurance in Slovenia is ZPIZ, which has the status of a public institution and covers by compulsory pension and disability insurance the entire territory of the Republic of Slovenia as a uniform risk group. The head of the Institute is in Ljubljana, whereas the service has been brought closer to the insured and pensioners through district offices and branch offices of the Institute.

The Institute also functions as a contact agency with foreign insurance bearers, exercising insurance according to international agreements (conventions).

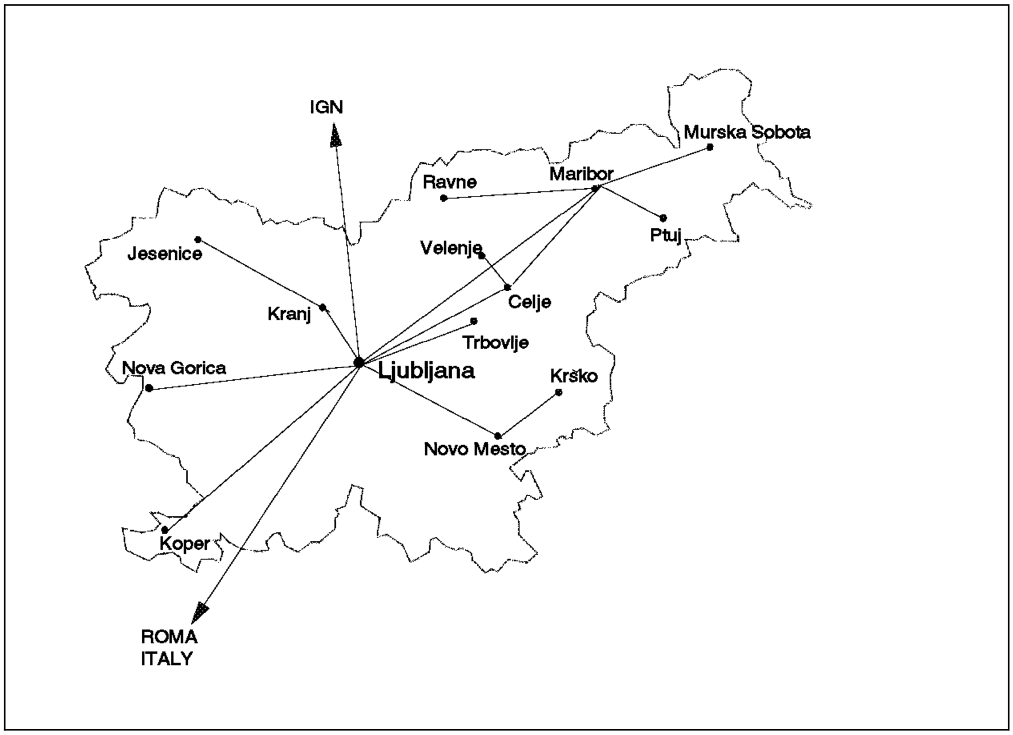


Figure 6. ZPIZ Slovenia

1.5.4 ZPIZ Project Requirement

- Establish a prototyping environment on database C/S using DRDA
- Enable new types of applications using workstation capabilities
- Enable skill transfer, especially for:
  - DRDA implementation
  - Multi-platform and networking
  - Security and administration
  - Project management
- Gain multi-language experience

1.5.5 Project Conclusion

Overall, the joint project was a great benefit to ZPIZ, especially in gaining experience in a comprehensive DB2 DRDA environment using multiple platforms and involving multiple code pages.

The project team can justify much more realistically already settled ZPIZ strategic I/S directions and with confidence suggest new or modified ones. ZPIZ is now aware of the complexities, and very often hidden difficulties, they could face by going in the client/server direction.

The project timeframe was from May 13th, 1996 till July 19th, 1996. It was concluded, as planned, in these 10 weeks.

1.5.6 Benefits

ZPIZ has implemented architected and open solutions, such as CDRA, DRDA and APPC, to gain all the benefits offered by these solutions. These benefits will not be repeated here as they are expressed in other places in this document. The project also offers the following advantages:

- customized and documented DRDA client/server environment
- prototyping system
- enable new types of applications
- network and location transparency
- code page conversions from Latin-1 to Latin-2 and vice versa
- multi-platform and multi-language experience
- enhance skills and expertise
- enhance existing security options
- clear picture of the masked complexities
- management of dispersed and independent geographies
- establish peer contacts with IBM ITSO
- become leading expert in an area
- improve ZPIZ professional image
- higher quality support for end users
- enrich publishing and presentation skills and experiences
- awareness of new skill requirements
- awareness of management issues, such as data, systems and networking
- awareness of security exposures and options

The resulting documentation and established environment will be used as a ***prototyping system*** for establishing a comprehensive client/server database production environment and providing a framework for enabling new types of applications.

The prototyping system will be used as an evolutionary system. This means that after the prototyping system is built and documented, it will be developed further, as business needs demand, until it is transformed into a production environment. The purpose of this prototyping system is to give the basic functions only; it does not go into such details as systems or data management. The prototyping system will be used as a tool for modeling, upon which the users will experiment with different ideas, decisions, database design. They will quickly be able to implement the changes to introduce better quality and suitability for the production distributed database and application environment. This prototyping system is, from the methodology point of view, similar to the simulation. It allows the testing of the complex system without any risk to the production system. Its purpose is to be able to better predict the behavior of the production system and to examine more efficient solutions.

1.6 DRDA Summary

Distributed Relational Database Architecture (DRDA) is a set of protocols that allow transparent connectivity between RDBMSs (Relational Data Base Management Systems) on different platforms, whether they are in different rooms or on different continents. DRDA coordinates communication between systems by deriving what must be exchanged and how it must be exchanged.

DRDA is a robust software architecture that protects data integrity and security. At this time, DRDA describes two levels of distribution: RUW (Remote Unit of Work) and DUW (Distributed Unit of Work).

RUW allows an application to access one database server within one unit of work (transaction), DUW allows an application to access **more** than one database server within one unit of work, that is, the application can switch between database servers before committing the data. This gives an application the ability to involve multiple (distributed) database servers at the same time.

DRDA is transparent to the application programs. This means there are **no** definitions required within the application programs. This is an important advantage, for example providing application portability.

1.6.1 What Architectures are Used by DRDA

- **System Network Architecture (SNA)**

  Describes the rules on which products agree when exchanging data.

- **Distributed Data Management Architecture (DDM)**

  Describes the model of the database environment. It provides command parameters, data objects and reply messages that work as an intermediate language which all systems understand.

- **SNA Management Services Architecture (MSA)**

  Transfers and manages alerts. It defines a method of sending and recording error messages.

- **Formatted Data Object Content Architecture (FD: OCA)**

  Provides a description of the data with information about data types and representation.

- **Character Data Representation Architecture (CDRA)**

  Provides character data integrity when data is transmitted between systems with different code pages and character representation.

1.6.2 DRDA Terminology

In a DRDA environment, there are three different roles: client, application requester and application server. Figure 7 illustrates the relationship between these three roles. Every database platform can play more than one role for different applications at the same time. So it may be that DB2 for AIX is acting as an AS for an OS2 application and, at the same time, as an AR for the AIX application connecting through DRDA to a VSE system.

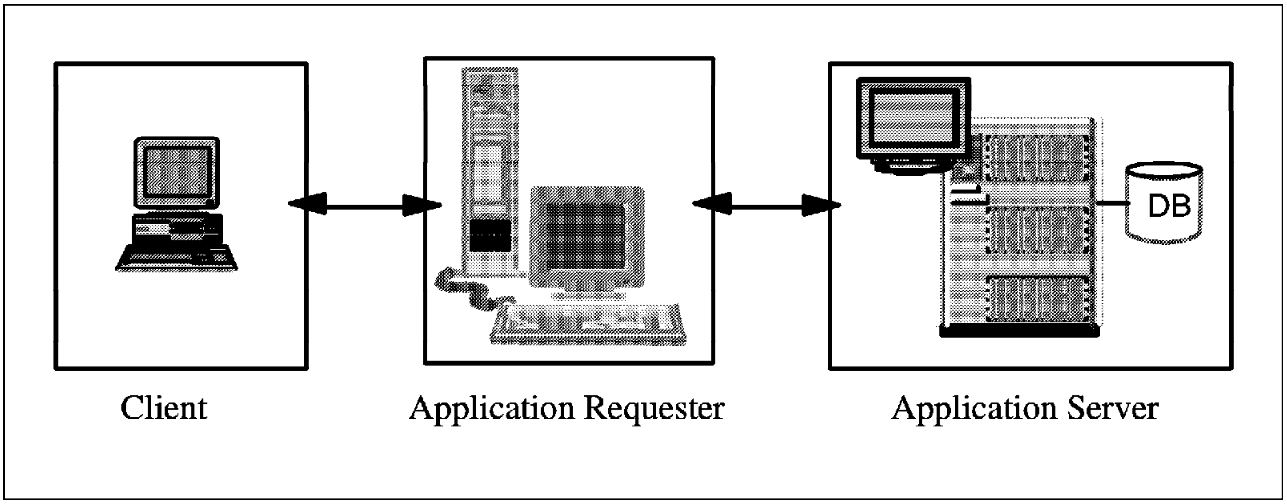


Figure 7. DRDA Database Roles

- **Client** is the component that runs the application program which issues SQL statements to request the data.
- **Application requester** is the code that handles the application end of a distributed connection. For example in a workstation environment, the DDCS product functions as an AR on behalf of application programs.
- **Application server** is the code that handles the database end of the connection. It performs database operations for the client and sends the results back to the requesting client or AR.

1.6.3 Products Required to Support DRDA

This depends on the platform (MVS, VM, VSE, OS/2, AIX and others), whether you are AR or AS and your connectivity and networking requirements. This means database C/S solutions in general can be planned, implemented and used according to indicated customer requirements. More details on these aspects are available in the DRDA library. Please refer also to the publication *Distributed Relational Database Cross Platform Connectivity and Application*, SG24-4311.

1.7 Multi-Language Considerations

The problem situations associated with exchanging data in multi-platform, and especially in combination with multi-language environments, have evolved with the data processing industry. The primary problem starts from the need to handle a variety of different character sets and the different encoding schemes used to represent them. This variety exists for historical reasons.

CDRA provides the foundation to address this need and permits systems and application systems to achieve consistent processing, interchange and presentation of graphic character data.

CDRA defines a consistent method for data identification by:

- describing code page and character sets by a Coded Character Set ID (CCSID).
- defining consistent conversion mapping to manage presentation of graphic character integrity as data is interchanged among any pair of databases.

1.8 CDRA Overview

1.8.1 Background on Data Interchange

People communicate with one another using mutually understood languages, which are generally based on well-defined alphabets. An alphabet can be thought of as a collection of symbols or graphic characters that have well-defined meanings.

Computer networks must also deal with the problem that not all computers have the same repertoire of graphic characters. For example, the Personal Computer handles characters unknown to the host (mainframe), and a host in the US does not recognize some of the characters of a host in Greece.

Figure 8 shows how the $ and ¢ symbols entered in a host in the US may appear on host computer screens in other countries.

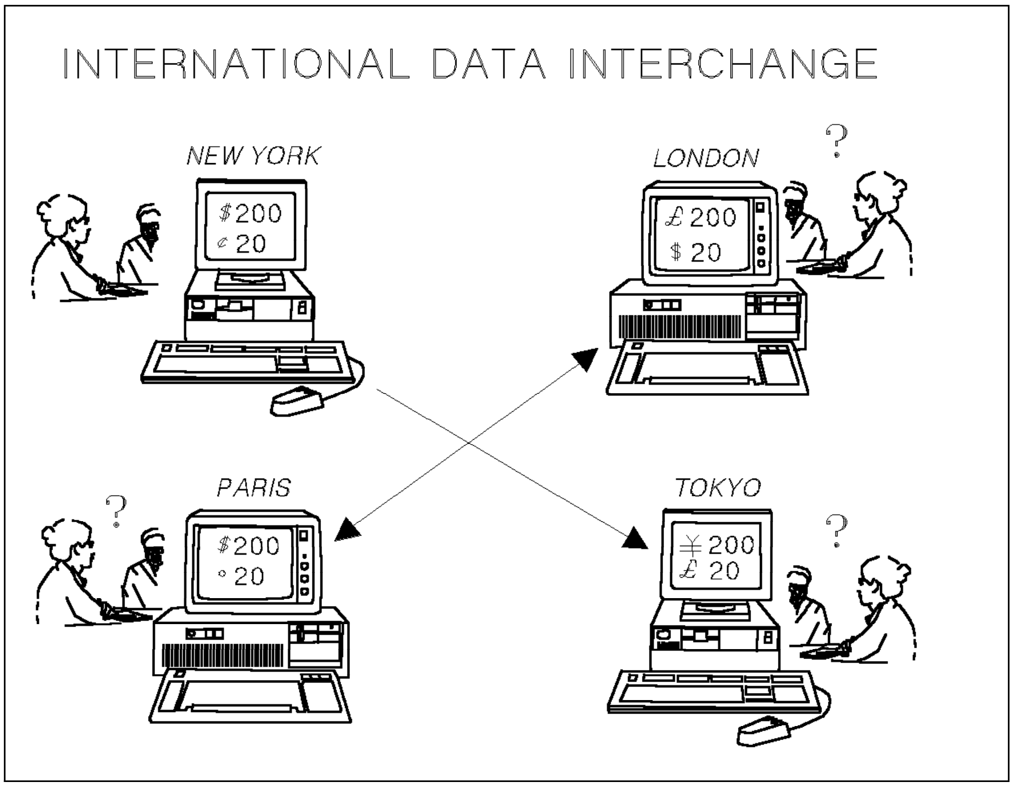


Figure 8. Different Graphic Character Codings

Application or product-specific solutions are not adequate. What is needed is an architecture that can lead to consistent cross-system support for the management of data integrity. Character Data Representation Architecture (CDRA) provides the foundation to address this need.

1.8.2 Requirements Summary

IBM recognizes a common set of requirements concerning the management and processing of coded graphic characters. These are summarized as follows:

- Facilitate the preservation of the meaning of coded graphic characters
- Provide support for a consistent set of coded graphic characters within and across IBM systems
- Reduce the proliferation of coded graphic character sets and code pages
- Provide an identification mechanism for coded graphic characters
- Provide necessary resources in support of the identification mechanism
- Provide for migration and coexistence support

1.8.3 The Solution is Character Data Representation Architecture

CDRA is an IBM architecture that permits system and application software to achieve consistent processing, interchange, and presentation of graphic character data in System Application Architecture (SAA) environments. It is also applicable to other computing environments. CDRA supports the principle of data and application independence, which is a vital requirement of distributed computing environments.

CDRA addresses the requirements stated previously for coded graphic character management, by providing:

- An identification mechanism and associated supporting resources
- Functions such as tagging to associate the identifier with data and applications
- Tables necessary for consistent code point conversion of coded graphic character data
- Recommended strategic coded graphic character sets

Identification Mechanism

The identification mechanism defined by CDRA is used to specify the identity of graphic character data, and to uniquely refer to this data at any place in a system. It can be thought of as providing additional information to eliminate the ambiguity inherent in the binary code point. This identification mechanism has two forms:

- The short form is a fixed-length two-byte identifier, known as the Coded Character Set Identifier (CCSID), which represents the elements of the long-form identifier
- The long form is a variable length identifier, composed of multiple elements:
  - Encoding scheme identifier (ES)
  - Character set identifier (CS)
  - Code page identifier (CP)
  - Further (CS, CP) elements as needed for each additional pair of character set and code page identifiers
  - Additional coding-related information (ACRI)

Tagging

Products implementing CDRA associate the CDRA identifiers with the data objects they manage. This ′tagging′ of data objects is the method used to identify the meaning of the coded graphic characters in the object. This allows a graphic character that has different code points assigned in different machine types to maintain its meaning.

Code Point Conversion

CDRA describes how to manage the representational differences in coded graphic characters, and the criteria to be used for the creation of character conversion tables and methods.

Strategic Coded Graphic Character Sets

CDRA defines the SAA character sets and code pages that can be used to minimize differences in coded graphic character representations and related potential data loss.

Architecture Strategy

CDRA addresses a complex set of coded graphic character problems that are fundamental to computing and are manifested throughout computing systems. The architected solution is segmented in such a way that it can be implemented in manageable steps and can be extended to accommodate future requirements.

The elements of the architected solution are the Architecture Base, Character Set Groups, and Levels.

- **Architecture Base**

  This defines the set of identifiers, services, resources, and processing guidelines to resolve current problems, and can be extended to address future needs.

- **Character Set Groups**

  Character set groups are selected character sets categorized into groups and subgroups according to their common properties (such as countries and languages). A selected few of these are defined as SAA Character Sets for each group or subgroup.

- **Levels**

  The Levels segment the architecture into function sets. Level 1 encompasses the elements of the architecture needed to deal with standard character sets that are currently being used. The primary objective is to achieve character data integrity and consistent processing among SAA environments within a country or group of countries that have a common character set.

  Future levels will extend the architecture to provide a global solution to the character data integrity problem. A global large character set supported with the appropriate identifier, services and resources will be the basis for processing and data interchange across system and country boundaries.

1.8.4 Architecture Benefits

CDRA provides a comprehensive method of data identification that permits unambiguous interpretation of coded graphic characters for correct and consistent data handling.

It defines consistent code point conversion rules and tables to be used during data interchange between different computing environments.

It specifies strategic character sets and code pages that minimize the need for code point conversions, and permit preservation of graphic character integrity.

It also provides an extendable base for migration to a global character data solution.

The benefits of CDRA increase substantially as the number of implementing products grows within the SAA environments. Also, since the architecture is not unique to SAA, similar benefits may be realized through product implementation in other environments.

The following CDRA publications provide technical descriptions and specifications for both Western and Far-East countries:

- *Character Data Representation Architecture - Level 1, Reference, SC09-1390* describes the rationale, strategy, and concepts behind the architecture. It also provides the details pertinent to Level 1 of the architecture.
- *Character Data Representation Architecture - Level 1, Registry, SC09-1391* is the CCSID registry. It also contains CDRA conversion tables and guidelines for their generation.

Chapter 2. Multi-Platform Considerations

2.1 Connectivity

Today, there are many communication protocols in the world. When you set up your own network, which one you should use depends on what platforms and products you have already, and which functions you want to support in the near future. In our scenarios, we use the following communication protocols:

- DRDA
- APPC
- TCP/IP
- NetBIOS

For your general information, currently the DB2 family consists of the following base products:

DRDA Servers	Requesters	Client Application Enablers	Communications Protocols
SQL/DS (VSE)	SQL/DS (VM)	DB2 for AIX	LU 6.2
SQL/DS (VM)	DB2 for MVS	DB2 for OS/2	APPC/APPN
DB2 for MVS	DB2 for OS/400	DB2 for Windows	NetBIOS
DB2 for OS/400		DB2 for DOS	TCP/IP
DB2 for AIX		DB2 for HP UX	IPX/SPX
DB2 for OS/2		DB2 for Solaris	
DB2 for Solaris			
DB2 for HP UX			

The family also includes complementary products such as IBM's Data Replication solutions or Query solutions and the DB2 family continues to grow.

Any requester can be connected to any server with the appropriate communications protocol, and as business needs change, the system implementation can change with little impact to the existing applications. This table shows you the supported communication protocols used between DB2 clients and the related DDCS gateways.

Table 1. Supported Communication Protocols					
DB2 Clients	**Protocols**	**DDCS OS/2**	**DDCS AIX**	**DDCS HP-UX**	**DDCS Sun Solaris**
DOS V1.2	NetBios IPX/SPX TCP/IP APPC	Yes Yes Yes No	No Yes(1) Yes No	No Yes(2) Yes No	No Yes(2) Yes No
OS/2 V2.1	NetBios IPX/SPX TCP/IP APPC	Yes Yes Yes Yes(3)	No Yes Yes Yes	No No Yes No	No No Yes No
Windows V2.1	NetBios IPX/SPX TCP/IP APPC	Yes Yes Yes No	No Yes Yes No	No Yes Yes No	No Yes Yes No
AIX V2.1	NetBios IPX/SPX TCP/IP APPC	No No Yes No	No No Yes No	No No Yes No	No No Yes No
HP-UX V2.1	NetBios IPX/SPX TCP/IP APPC	No No Yes No	No No Yes No	No No Yes No	No No Yes No
Solaris V2.1	NetBios IPX/SPX TCP/IP APPC	No No Yes No	No No Yes No	No No Yes No	No No Yes No

1. Supported natively by DB2 for AIX, and also provided by FireFox, Inc. NOV*IX for NetWare product.
2. Provided by FireFox, Inc. NOV*IX for NetWare product.
3. If the SNA network supports APPN, DB2 Client for OS/2 can also participate in the APPN network.

2.1.1 DRDA

The DB2 family of products implements DRDA. In our environment, we installed the following DB2 products on different platforms:

- SQL/DS for VM and VSE V3.5
- DB2/6000 for AIX V 2.11
- DB2/2 for OS2 V 2.11

Figure 9 illustrates the ITSO DRDA database configuration and the possibilities to access each other′s data.

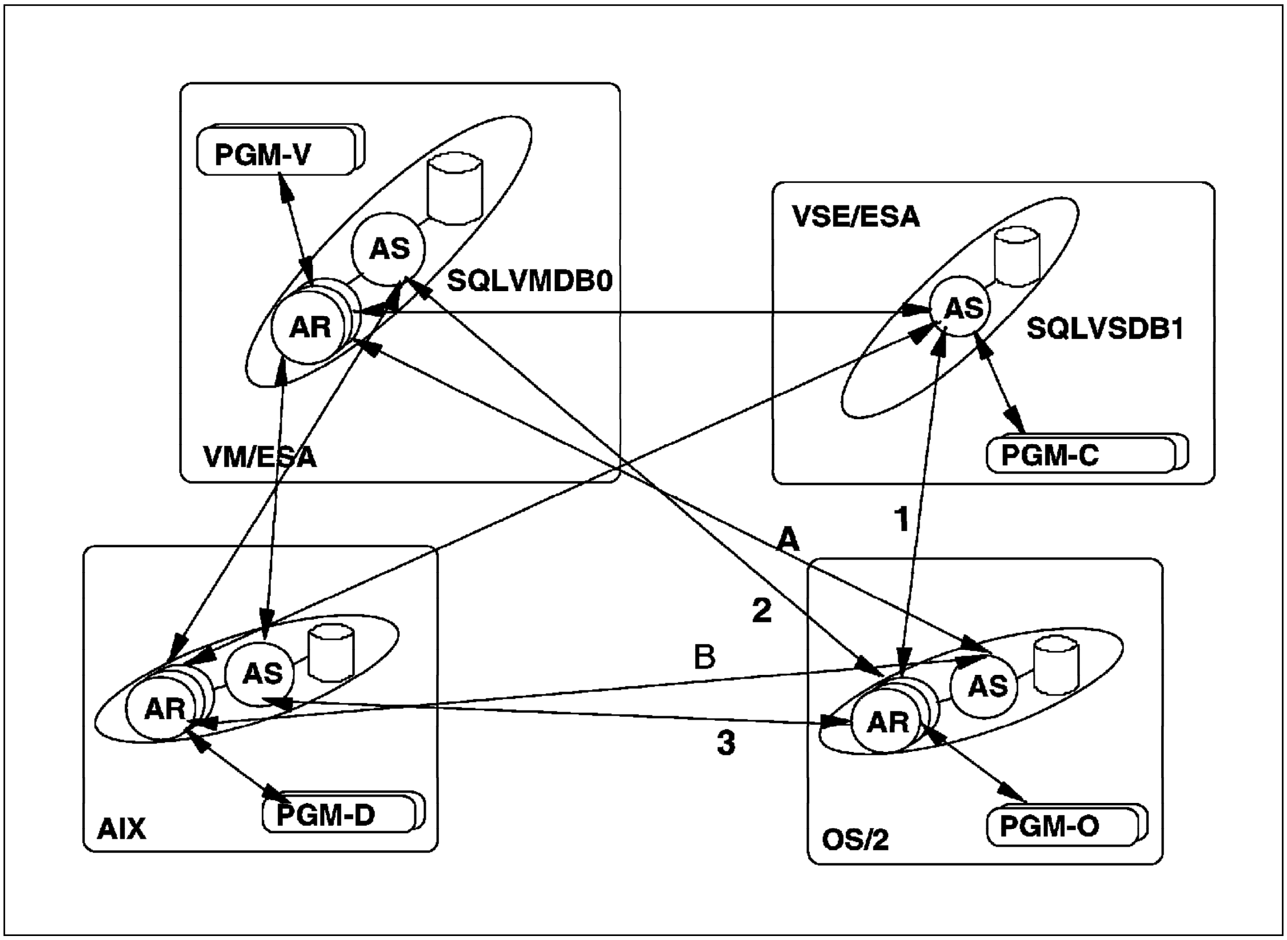


Figure 9. ITSO DRDA Database Configuration

The OS/2 application (PGM-O) can access its own local database, and the other databases located on the different platforms (VM, VSE, AIX) through DRDA connection, that is, through the AR in OS/2 and the AS on the VM, VSE or OS/2 platforms, marked 1, 2 and 3 in Figure 9. At the same time, this DB2/2 on the OS/2 platform can function as an AS handling the requests from other ARs to access its database, shown as A and B in Figure 9.

The one exception is that the DB2 for VSE can only be an AS. Therefore, the VSE application can **only** access its own database, or databases under VM using VM/VSE Guest Sharing which is a not-architected, ″private″ solution.

Figure 10 shows the entire ITSO and ZPIZ DRDA environments for your information.

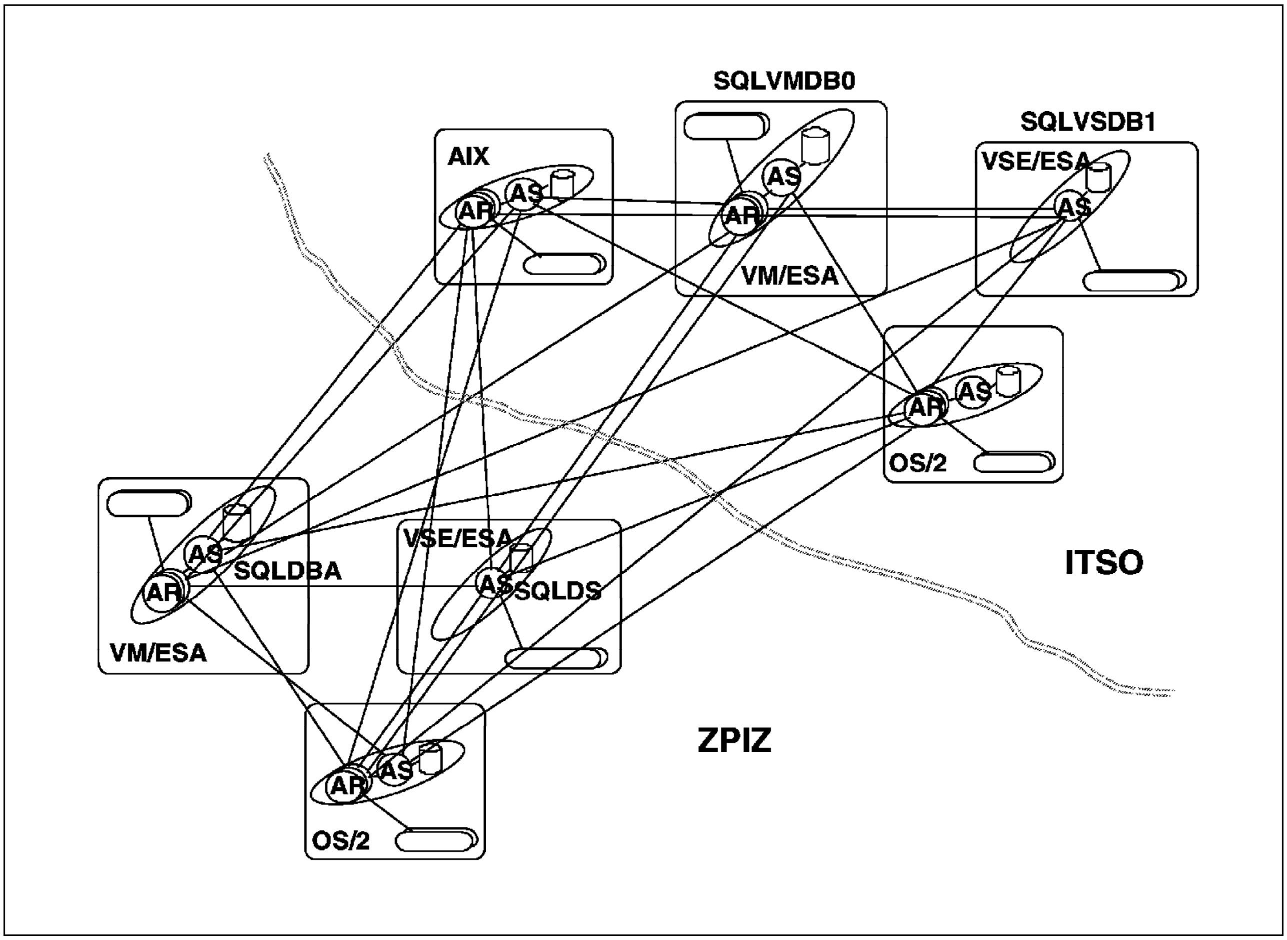


Figure 10. DRDA Database Configuration between ITSO and ZPIZ

2.1.2 TCP/IP

Transmission Control Protocol/Internet Protocol (named TCP/IP after its two main standards) is a set of protocols that can be used to communicate across any set of interconnected networks called ″internets″.

The TCP/IP protocol set consists of four functional layers:

- **The Application Layer**

  Provides for application-to-application cooperation on the same host, network or across a set of interconnected networks. Examples are Telnet (protocol for remote terminal connection), FTP (File Transfer Protocol) and SMTP (Simple Mail Transfer Protocol) for electronic mail.

- **The Transport Layer**

  Provides for reliable end-to-end data transfer. Protocol examples are Transmission Control Protocol (connection-oriented) and User Datagram Protocol (connectionless).

- **The Internet Layer**

  Represents the internetwork that provides for fast connectionless data transfer between two network nodes. The most important protocol in this layer is IP.

- **The Network Interface Layer**

  Interfaces to the actual network hardware. TCP/IP supports many network interfaces, for example IEE802.2 for LANs and X.25 for packet switching networks.

For more information about TCP/IP, please refer to the following manual: *TCP/IP Tutorial and Technical Overview*, GG24-3376.

In our environment, we created two files in directory C: TCPIP\ETC on the client side. These are the HOSTS and SERVICES files.

```
9.164.178.103    itso1
```

Figure 11. HOSTS File

```
DB2AIX     2570/tcp
DB2AIXI    2571/tcp
```

Figure 12. SERVICES File

Figure 13 shows the relation between the parameters of the server (AIX) and the client (OS/2).

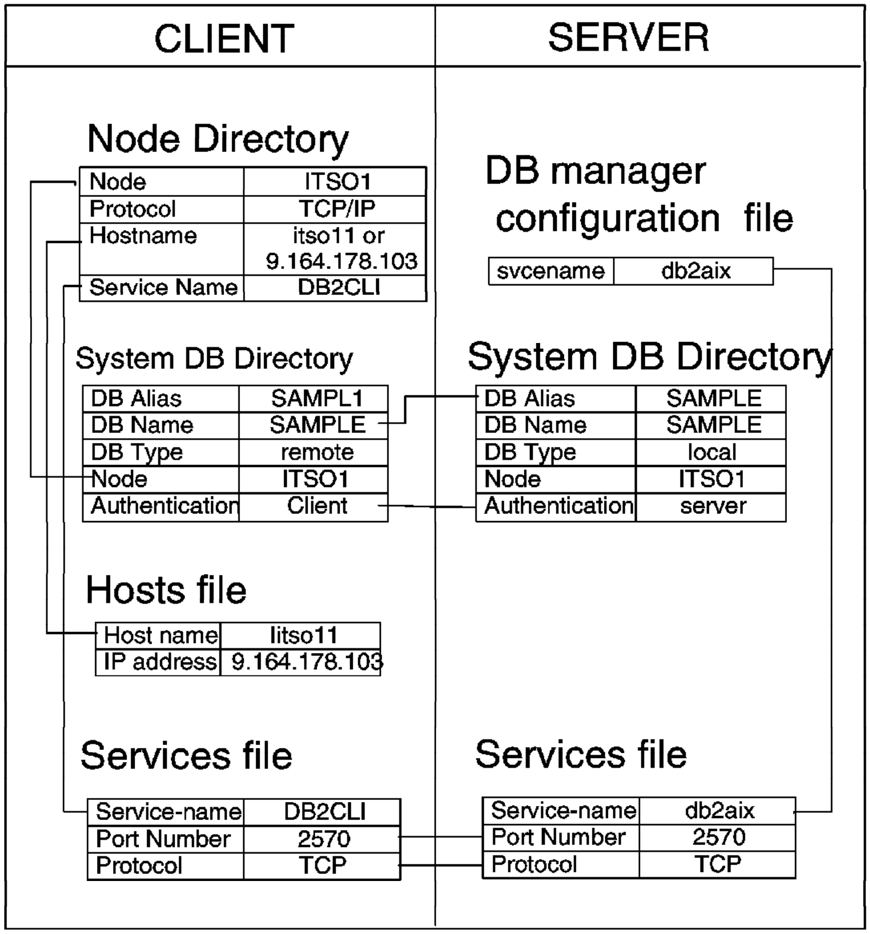


Figure 13. TCP/IP Parameters in ITSO Environment

2.1.3 Others Protocols

The other protocols used in conjunction with DB2/2 are described below:

- **APPC**

  Advance-Program-to-Program Communications is an implementation of SNA LU 6.2 and is used for Peer-to-Peer communications.

- **NetBIOS**

  NetBIOS is an API between a local area network adapter and programs and is the preferred communication protocol for LAN connection because of less memory usage on the client workstation and simplified database communications configuration.

In our environment, all the connections between host (VM/VSE) and workstation (AIX, OS/2 and DOS/Windows) are using APPC protocol.

From the host side, we customized the following products to support APPC connections:

- AVSVM and VTAM in the VM system
- VTAM and CICS in the VSE system

- NCP definition
- 3174 customization

Please see Appendix A, “System Tables and Sample Programs” on page 191 for details.

From the workstation side, we configured the following parts:

- CM/2 in OS/2
- SNA server/6000 in AIX
- MS SNA server in DOS/Windows

There are two paths for a workstation to connect to the host; one is through the 3745, the other is through the 3174. But only one of them can be activated at any one time; if you want to change the path, you have to restart CM/2 in OS/2 or SNA Server/6000 in AIX.

The NetBIOS protocol is used to set up the connection between workstations, such as OS/2 and DOS/Windows. The following parameters are used in our environment.

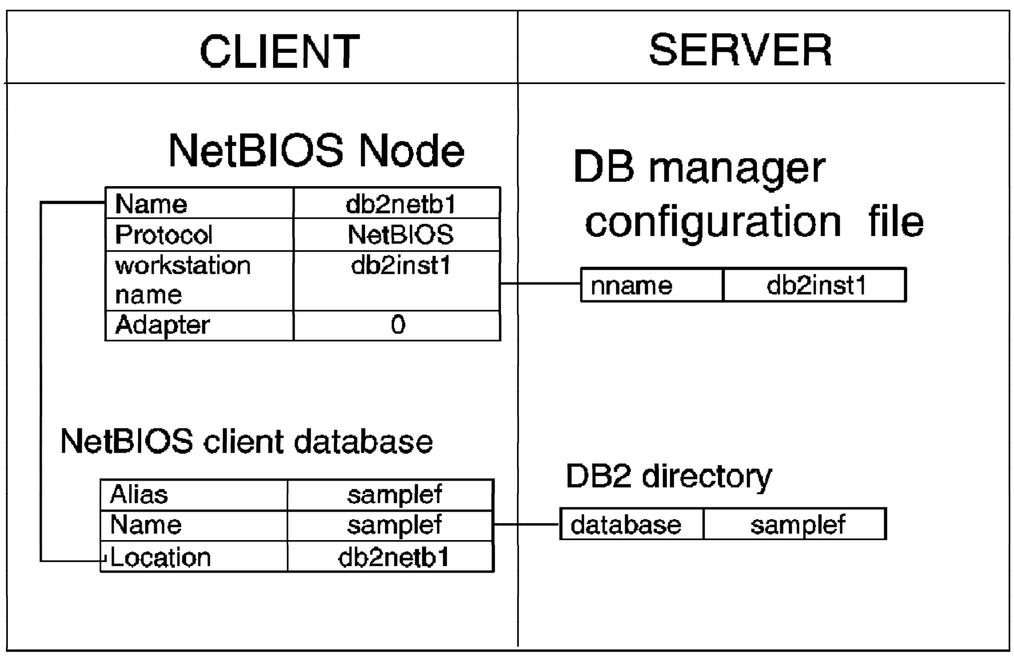


Figure 14. Parameter for NetBIOS Protocol

For more information, please see the manual *Installing and Using DB2 Clients for Windows*.

2.2 Multi-Language Support Requirements

A brief description of appropriate terminology follows:

- **Code page**

  A set of assignment of characters to code points. The code pages have numeric identifiers, for example, 500 is the International code page.

- **Code Character Set**

  A set of characters in which there is a one to one relationship between the characters and their coded representations. These character sets also have numeric identifiers. Several code pages can use the same character set. For example, code pages 37 (English), 273 (German) and 500 (International) are some of the code pages that use character set 697. However, these code pages do not necessarily use the same subset of the character set.

- **CCSID**

  Code Character Set Identifier, a 16-bit number that includes an encoding scheme identifier, character set identifier, code page identifiers, and other information that uniquely identifies the coded character representation used. For single and double byte CCSIDs, the CCSID is generally the same as the code page. For example, the CCSID for the International code page is 500.

2.2.1 Support for Different Code Pages

There are many different code pages due to the different languages. In Table 2 you can see some of the supported code pages and the character set groups to which they belong. It is not a comprehensive list of the code pages supported.

Table 2. Code Page Support

Character Set Group	**Supported Code Pages**
Group 1: Latin-1	37, 273, 277, 278, 280, 284, 297, 500, 437, 819, 850, 863, 1004, 1051, 1252, 1275
Group 1a: Arabic	420, 864, 1046, 1089, 1256, 4960
Group 1a: Cyrillic	855, 866, 915, 1025, 1251, 1283, 4951
Group 1a: Latin-2	852, 870, 912, 1250, 1282, 4948
Group 1a: Greek	813, 869, 875, 1253, 1280
Group 1a: Hebrew	424, 862, 916, 1255
Group 2: Japanese	932, 942, 943, 954, 5035
Group 2: Korean	933, 949, 970
Group 2: Simpl. Chinese	836, 837, 935, 1381, 1383
Group 2: Trad. Chinese	835, 937, 938, 948, 950, 964, 28709
Group 2: Thai	838, 874
Group 1a: Turkish	857, 920, 1254, 1281

When you start a database server or create a new database, you can choose different code pages which are suitable for your environment. This is very useful for those who have specific requirements, for example when a special character such as, ∩, Π or Ÿ is needed.

2.2.2 Conversion Between Different Code Pages

Different systems represent data in different ways with different code pages. When data is moved from one system to another, data conversion is required if they use different code pages. The DB2 family of products supports such data conversion.

Conversion of numeric data depends on how a specific data type is represented on the sending and receiving systems. This kind of conversion is built into the database products and is transparent to the user.

Character conversion is done in accordance with DRDA. The conversion is always done on the receiving system. This is true for DRDA connectivity where the host system and the DDCS product are involved. For Common Server C/S connectivity, for example if OS/2 is the client and AIX is the server or vice versa, the conversions are always done by the server.

Performance should always be a major concern as character conversion can have an adverse affect on it. To avoid or minimize such conversion, choose the same CCSID for the AS and AR, although this is not always possible. Investigate whether you can use the International (CCSID=500) character set when you are supporting different languages. This character set includes many different languages and as such helps minimize character conversion.

2.2.3 Support for Alternate Collating Sequences

The database manager compares character data using a *collating sequence*. This is an ordering for a set of characters that determines whether a particular character sorts higher, lower, or the same order as another.

The country code and the code page are used to determine the default collating sequence that will be used for the database. This default collating sequence may be explicitly overwritten. The DB2 family of products provides related parameters or APIs to overwrite the default collating sequence.

The differences between EBCDIC and ASCII cause differences in sort orders in the various database products, and also affect ORDER BY and GROUP BY statements. One way to minimize these differences is to create a user-defined collating sequence that reflects the EBCDIC sort order. This collating sequence can by specified only when creating a new database.

2.3 DB2 Multi-Language Support for VM/VSE

In the VM/VSE environment, in order to allow DB2 to support multi-languages, the following changes have been made:

- New preprocessing options have been added to support CCSID.
- A CCSID integer parameter has been added to the CREATE TABLE and ALTER TABLE statements.
- The DESCRIBE statement has been enhanced to return information in the SQLDA regarding the CCSID of a column.
- Additional rows have been added to the catalog tables:

  SYSTEM.SYSCOLUMNS
  SYSTEM.SYSKEYCOLS
  SYSTEM.SYSOPTIONS

SYSTEM.SYSCHARSETS
SYSTEM.SYSCCSID
SYSTEM.SYSSTRINGS

2.3.1 Support for Different Code Pages

DB2 for VM & VSE is shipped with different CCSIDs to support different code pages. A system wide default CCSID can be specified. Users can override this system default for their virtual machine, and specify a special CCSID for a particular column. The following table is the list of the CCSIDs available with SQL/DS V3.5, by selecting on the SQL/DS system table SYSTEM.SYSCCSIDs.

CCSID	SUBTYPE	SBCSID	DBCSID	CHARNAME
37	S	0	0	ENGLISH
273	S	0	0	GERMAN
277	S	0	0	DANISH-NORWEGIAN
278	S	0	0	FINNISH-SWEDISH
280	S	0	0	ITALIAN
284	S	0	0	SPANISH
285	S	0	0	UK-ENGLISH
290	S	0	0	290
297	S	0	0	FRENCH
300		0	0	
420	S	0	0	ARABIC
423	S	0	0	GREEK-423
424	S	0	0	HEBREW
500	S	0	0	INTERNATIONAL
833	S	0	0	833
834		0	0	
835		0	0	
836	S	0	0	836
837		0	0	
838	S	0	0	THAI
870	S	0	0	870
871	S	0	0	ICELANDIC
875	S	0	0	GREEK
880	S	0	0	CYRILLIC-880
930	M	290	300	930
933	M	833	834	KOREAN
935	M	836	837	S-CHINESE
937	M	28709	835	T-CHINESE
939	M	1027	300	939
1025	S	0	0	CYRILLIC
1027	S	0	0	1027
4396		0	0	
5026	M	290	4396	KATAKANA
5035	M	1027	4396	JAPANESE-ENGLISH
28709	S	0	0	28709
65535	B	0	0	

Figure 15. CCSIDs and CHARNAME Supplied with SQL/DS

The subtype value can be:

B for bit data
M for a mix of single-byte and graphic data (double-byte)
S for single-byte
BLANK for anything other than non-character

The CCSIDs can be specified in the following places:

1. In the VM system, you can specify the CHARNAME parameter in the SQLSTART statement, to identify the CCSID you want to use. For example:

```
EXEC SQLSTART DBNAME(SQLVMDB0) DCSSID(SQLDBA) PARM(PROTOCOL=AUTO,CHARNAME=INTERNATIONAL)

ARI0015I SYSMODE parameter value is M.
ARI0015I EXTEND parameter value is N.
ARI0015I CHARNAME parameter value is INTERNATIONAL.
ARI0015I DBNAME parameter value is SQLVMDB0.
ARI0015I RESID parameter value is SQLVMDB0.
    .
    .
    .
ARI0282I LUW UNDO is completed.
ARI0281I LUW REDO is completed.
ARI0143I The application server has been initialized
         with the following values:
         CHARNAME = INTERNATIONAL, DBCS = NO, CHARSUB = SBCS,
         CCSIDSBCS = 500, CCSIDMIXED = 0, CCSIDGRAPHIC = 0.
ARI0134I Application server SQLVMDB0 has been
         identified as a global resource.
ARI0060I SQL/DS initialization complete.
ARI0045I Ready for operator communications.
```

Figure 16. SQLSTART Example

In the VSE system, you can specify the CHARNAME parameter in the ARISQLDS statement, to identify the CCSID you want to use. For example:

```
EXEC ARISQLDS,SIZE=AUTO,PARM=¢DBNAME=SQLVSDB1,STARTUP=W,LOGMODE=Y,
               NCUSERS=15,RMTUSERS=5,CHARNAME=INTERNATIONAL¢


  Z1 0045 ARI0282I LUW UNDO is completed.
  Z1 0045 ARI0281I LUW REDO is completed.
  Z1 0045 ARI0143I The application server has been initialized
                   with the following values:
                   CHARNAME = INTERNATIONAL, DBCS = NO, CHARSUB = SBCS,
                   CCSIDSBCS = 500, CCSIDMIXED = 0, CCSIDGRAPHIC = 0.
  Z1 0045 ARI0060I SQL/DS initialization complete.
  Z1 0045 ARI0045I Ready for operator communications.
```

Figure 17. ARISQLDS Example

2. A user can enter the SQLINIT command to specify a CHARNAME, and thereby CCSIDs, for his virtual machine environment. The following command will establish the international character set as the single-byte default for the user machine.

   SQLINIT DB(SQLVSDB1) PROTOCOL(AUTO) CHARNAME(INTERNATIONAL)

3. In the SQLPREP EXEC, you can specify the CCSID(x) parameter as one of the preprocessing options.

4. In the CREATE TABLE and ALTER TABLE command, the CCSID parameter is added to enable users to specify the CCSID on a column level.

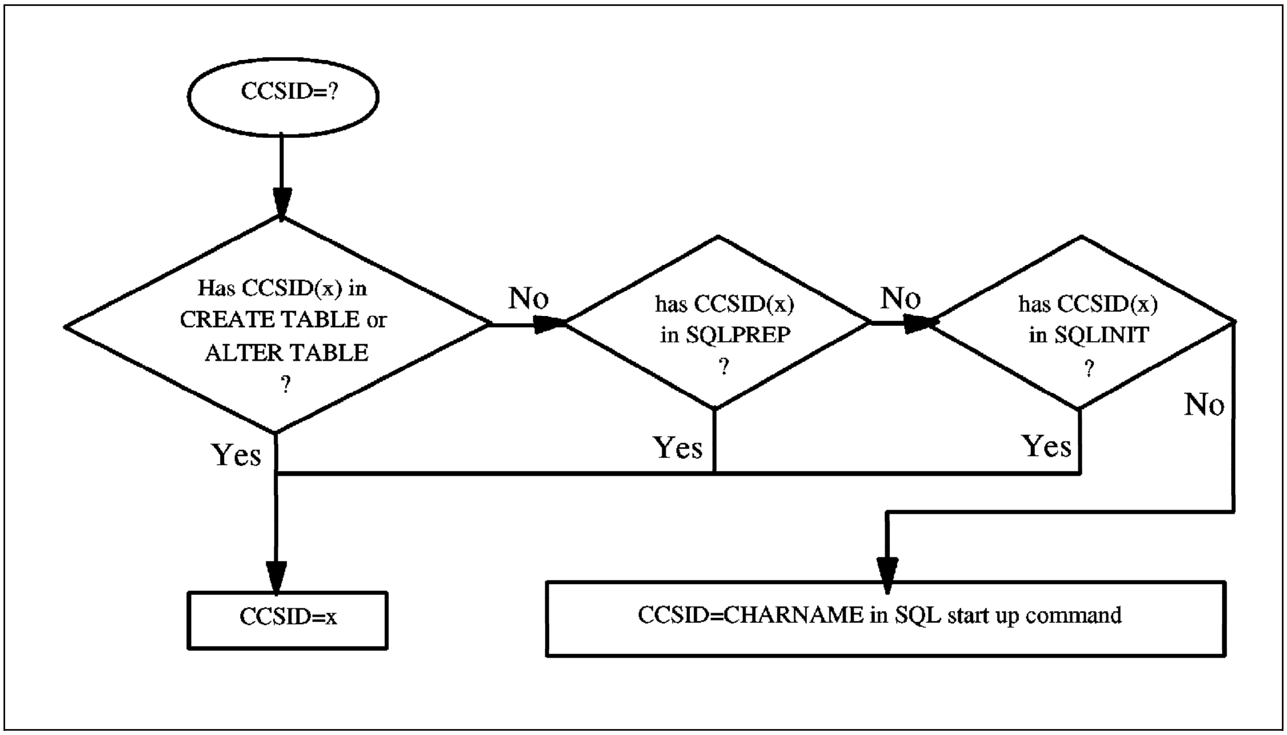


Figure 18. Relationship of the Four Ways to Specify the CCSID

2.3.2 Conversion Between Different Code Pages

When accessing the database using different code pages and/or character sets and/or encoding schemes, in order to get meaningful data, a conversion between the different code pages must be done. SQL/DS and DB2 support such conversion; it is transparent to the user.

The conversion between different code pages is performed based on the information in the SYSTEM.SYSSTRINGS catalog table. This table contains all the conversion tables available for this SQL/DS system. Below is an extract from the SYSSTRINGS table.

```
INCCSID     OUTCCSID   TRANSTYPE  ERRORBYTE  SUBBYTE  TRANSPROC
--------    --------   ---------  ---------  -------  ----------
   273         500        SS          ?         ?
   500         273        SS          ?         ?
```

The conversion must be two ways, for example, if you have the conversion from source CCSID 273 to target CCSID 500, you also have to have the conversion from source CCSID 500 to target CCSID 273, otherwise, the conversion is meaningless.

For each combination, there are:

- Conversion tables and/or routines for the pairs that require conversion
- An indication of whether any code points in the source can result in an error
- An indication of whether any code points in the source can be mapped to a substitution character

The conversion will obey the following rules:

- **CCSID Conversion Rules for Data Value Assignment**

  When assigning a host variable value to a column, the following conversion rules apply:

  – If the CCSID of the value matches the CCSID of the column, the value is assigned without conversion.
  – If the data subtype of either the source or the target is BIT, the value will be assigned without conversion.
  – If the source value is either NULL or an empty string, the value is assigned without conversion.
  – If SYSSTRINGS contains an entry indicating no conversion is needed, the value is assigned without conversion.
  – If there is no entry for the CCSID pair in SYSSTRINGS, the value is not assigned and will result in an error.
  – If SYSSTRINGS contains an entry indicating conversion is needed, the source value is converted to the coded character set of the target column before the assignment is performed.

- **CCSID Conversion Rules for Internal Data Movements**

  For internal data movements, the rules that apply to data input also apply. The rules for concatenation are:

  – Concatenation of mixed and SBCS is mixed
  – Concatenation of mixed and BIT is BIT
  – Concatenation of SBCS and BIT is BIT
  – If any of the operands are “mixed,” the resulting data type is VARCHAR

  When concatenating two or more values with different CCSIDs, the CCSID of the result is determined at bind time. For each pair of CCSIDs, the resulting CCSID is determined by the first match of the following:

  1. If the two CCSIDs are the same, the resulting CCSID will be the same.
  2. If one of the values is BIT, then the final result will be BIT.
  3. If one value is SBCS, and the other is MIXED, the resulting CCSID will be that of the mixed value.
  4. If one value is from a column and the other is not, the resulting CCSID is that of the column.
  5. If one value is a string constant or special register, and the other is a derived string or host variable, the resulting CCSID is that of the string or special register.
  6. If one value is a derived string and the other is a host variable, the resulting CCSID is that of the derived string.
  7. In other cases, the resulting CCSID is that of the first value.

  All the rules for concatenation also apply when two or more columns with differing subtypes or CCSIDs are involved in:

  – UNION within a SELECT
  – UNION all within a SELECT
  – The VALUE scalar function

- **CCSID Conversion Rules for Data Comparison**

  The rules for data comparison are similar to those used for conversion on data value assignment.

1. Comparison is done without conversion if:

   – The CCSIDs of the two values match
   – The subtype of either value is BIT
   – Both values are empty strings
   – Both values are NULL

2. Conversion will be performed before comparison and involves the following steps:

   a. Determine which value is subject to conversion. This choice is based on the following hierarchy:

      – If one value is SBCS and the other is MIXED, the SBCS value is subject to conversion.
      – If one value is from a column and the other is not, the non-column value is subject to conversion.
      – If one value is a string constant or special register and the other is a derived string or host variable, the derived string or host variable is subject to conversion.
      – If one value is a derived string and the other is a host variable, the host variable will be subject to conversion.
      – In other cases, the second operand is subject to conversion.

   b. SQL/DS will search the SYSSTRINGS table using the CCSID of the value subject to conversion as the INCCSID, and the CCSID of the other value as the OUTCCSID. Conversion will be performed based on the row found for the CCSID pair in the SYSSTRINGS table, with the following exceptions:

      – If there is no entry for this CCSID pair in the SYSSTRINGS table, then an error is raised.
      – If the entry indicates that the CCSID pair does not require conversion, then the comparison will be done without conversion.

CCSID conversion is handled by DB2 and is transparent to end users. However, application programmers should be aware of the CCSID conversion rules if the application has the requirement to manipulate CCSIDs with host variables, perform data comparisons, or concatenation for data with differing CCSIDs, since the result may be affected by CCSID conversion.

In SQL/DS, you can define your own character set for special reasons. But we strongly recommend you to use the CCSID shipped with the system if possible as a user defined CCSID or character set will limit the portability of your applications.

For more information, please see 3.3.3, “Defining Slovenian Character Set” on page 62 or *Distributed Relational Database Cross Platform Connectivity and Application* SG24-4311.

2.3.3 Support for Alternate Collating Sequences

In SQL/DS, you can use field procedures to change the collating sequences for a column. A field procedure is a user-written exit routine and is called when a table is created or altered.

To assign a field procedure to a new column, include the FIELDPROC clause on either the CREATE TABLE or ALTER TABLE statement. For the syntax diagrams for the CREATE and ALTER TABLE statements, refer to the *SQL/DS Reference* manual. Figure 19 is the fieldproc-block of those diagrams.

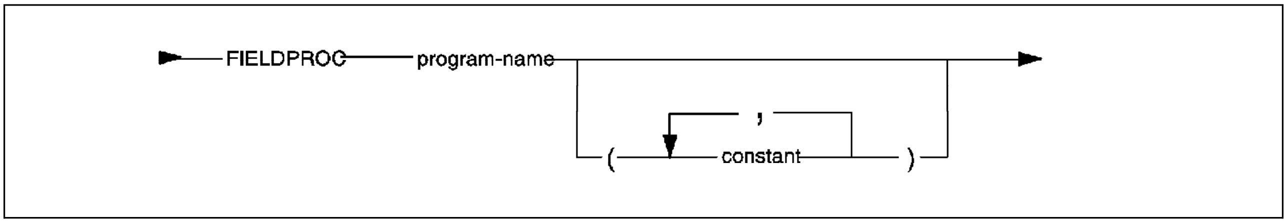


Figure 19. FIELDPROC-block Syntax

If you assign a field procedure to a column, it is called whenever values in that column are changed or are inserted, and it encodes the original value into one value that sorts correctly.

The following rules must be obeyed when writing field procedures:

- It must be written in Assembler.
- Its name must not start with ARI, to avoid conflict with the SQL/DS modules.
- It must not call any SVC services.
- It must store registers in an area pointed to by R13, and restore them before returning.
- It must be serially reusable.
- It must not contain any SQL statements.
- It must reside in the appropriate VSE library(VSE)/minidisk(VM) and be accessible when the database manager is running.
- It must support 31-bit addressing when the VSE system is running in ESA or VM/ESA supervisor mode.

A sample of coding FIELD PROCEDURE SLOSORT is presented in A.3, “Field Procedure Sample” on page 199. For more information about coding field procedures, please see the following books:.

- *System Administration for VSE/VM* GH09-8096/GH09-8084
- *Application Programming for VSE/VM* SH09-8098/SH09-8086

The values in columns with a field procedure are described to the database manager in the following catalog tables:

- SYSTEM.SYSCOLUMNS
- SYSTEM.SYSFIELDS
- SYSTEM.SYSFPARMS
- SYSTEM.SYSKEYCOLS

For more information about catalog tables, see the *SQL/DS Reference* manual.

2.4 DB2 Multi-Language Support for OS/2

DB2/2 also can support multiple CCSIDs, but it's very limited as you cannot add additional CCSIDs or conversion tables in DB2/2. After you install DB2/2, all CCSIDs and conversion tables which can be supported are fixed, this is true for all releases up to and including DB2/2 V2.

2.4.1 Support for Different Code Pages

You can use the command **chcp** to check how many different code pages can be supported in your DB2/2 environment. For example:

```
C: chcp<enter>
```

Figure 20. Check the Code Page

gives you the message

```
Active code page:   850
Prepared system code pages:   437,  850
```

Figure 21. Result of Checking the Code Page

In our example, you can see that there are two code pages supported, 437 and 850. One of them is currently in use. You cannot install an additional code page after you have finished the DB2/2 installation.

You also can use this command to change the active code page. For example:

```
chcp 437<enter>
chcp<enter>
```

Figure 22. Change the Code Page

gives you the message

```
Active code page:   437
Prepared system code pages:   437,  850
```

Figure 23. Result of Changing the Code Page

2.4.2 Conversion Between Different Code Pages

DB2/2 also supports the conversion between different code pages although this support is very limited. Therefore, when you issue the command **CONNECT TO** *database* to establish the connection with the AS, if the conversion is not supported, the connection will not be established. For example:

```
C: chcp
Active code page:   437
Prepared system code pages:   437,  850
```

Figure 24. Current Code Page in DB2/2

```
sqlinit query
ARI0717I Start SQLINIT EXEC: 10/17/96 11:16:37 CET.
SELECTED TABLE IS: SQL/DS
 DBNAME=SQLVMDB0
 DBCS=NO
 SYNCHRONOUS=NO
 DATEFORMAT=ISO
 TIMEFORMAT=ISO
 TRACERA=00
 LDATELEN=0
 LTIMELEN=0
 RELEASE=3.5.0
 WORKUNIT=YES
 QRYBLKSIZE=8
 PROTOCOL=SQLDS
 CHARNAME=870
 CCSIDSBCS=870
 CCSIDMIXED=0
 CCSIDGRAPHIC=0
 TRACEDRRM=0000
 TRACECONV=0
 SSSNAME=ARIQSTAT
ARI0796I End SQLINIT EXEC: 10/17/96 11:16:37 CET
Ready(00001); T=0.12/0.14 11:16:37
```

Figure 25. Current Code Page in SQL/DS for VM

The CCSIDs or code pages in DB2/2 and SQL/DS are not in the same language group. Code Page 437 is in the Latin-1 group, and the CCSID 870 is in the Latin-1a group. So when you want to make the connection between them, you will get the following message:

```
SQL0332N There is no available conversion for the source code page
†870† to the target code page †437†. Reason Code †1†. SQLstate=57017
```

Figure 26. DB2/2 Connect to SQL/DS for VM at ITSO

```
SQL30073N †119C† Parameter value †† is not supported. SQLSTATE=58017
```

Figure 27. DB2/2 Connect to SQL/DS for VSE at ITSO

They must be put into the same language group. There are two ways to do this:

- On the PC side, find the correct one in the prepared system code pages and use the command **CHCP xxx** to active it.
- On the host side, you can restart the database using the correct CCSID. For example:

```
SQLSTART DBNAME(SQLVMDB0) DCSSID(SQLDBA) PARM(PROTOCOL=AUTO,CHARNAME=INTERNATIONAL)
```

Figure 28. Change the CCSID in SQL/DS

After this, you can issue the command CONNECT TO database again, which should be successful. Following is the message shown.

```
    Database Connection Information

  Database product       = SQL/DS VSE3.5.0
  SQL authorization ID   = DEMO
  Local database alias   = IPFVS1A4
```

Figure 29. OS/2 Connect to VSE Database at ITSO

All conversion tables are carried internally and are not subject to change.

Generally, the character conversions will occur during the following situation:

- A client has a code page that is different from the code page of the database. Most of this conversion is done at the server. For Common Server C/S connectivity, for example if the client is OS/2 and the server is AIX or vice versa, the conversions are always done by the server. For DRDA C/S connectivity, for example if the client is on OS/2 or VM CMS and the server is on OS/390 or VSE/ESA, the conversions are always done on the receiving system, which can be AR or AS.
- An application has a code page that is different from the code page of the database. We recommend you to specify the code page that the application uses during pre-compile and bind.
- Data is retrieved from a DRDA server. In this case the data is converted at the DDCS server.

Note that the file names and BIT data, such as BLOB data, are not converted.

2.4.3 Support for Different Collating Sequences

You can specify the collating sequence which you want to use when you create the new database. The syntax is as following:

CREATE DATABASE database-name [USING CODESET codeset TERRITORY territory] [COLLATE USING { SYSTEM | IDENTITY }]

IDENTITY Identity collating sequence, in which strings are compared byte for byte

SYSTEM Collating sequence based on the current territory

You cannot change the collating sequences after the database has been created.

There are some sample user-defined collating sequences in the host language include file. They are available with DB2 for OS/2, just as for DB2 for AIX.

- sqle819a
- sqle819b
- sqle850a
- sqle850b
- sqle932a
- sqle932b

For detailed definitions, please see 2.5.4.3, “Samples of User-Defined Collating Sequences” on page 52.

2.5 DB2 Multi-Language Support for AIX

This section contains information about the National Language Support (NLS) provided by the DB2 database products. We have to consider the differences if we speak of NLS in terms of the AIX environment, in terms of the DB2 for AIX environment or in terms of a DRDA environment.

Different systems represent data in different ways. When data is moved from one system to another, data conversion sometimes must be performed. Products supporting DRDA will automatically perform any necessary conversions at the receiving system. With numeric data, the information needed to perform the conversion is the data type of the data and how the data type is represented by the sending system. With character data, additional information is needed to convert character strings. String conversion depends on both the coded character set of the data and the operation that is to be performed with that data. Character conversions are performed in accordance with the IBM Character Data Representation Architecture (CDRA).

We will see that in our environment we have several players that will deal with the problem of data representation and data conversion:

- The AIX operating system
- The DB2 for AIX database server or the clients
- The DRDA products based on CDRA
- The application running on a specific platform

2.5.1 AIX Locale

A locale defines conventions for a specified culture, such as time formatting and character classification, conversion, and collation.

2.5.1.1 Locale Overview for System Management

In an application you do not want to have built-in assumptions or dependencies on code set, character classification, character comparison rules, character collating order, monetary formatting, numeric punctuation, date and time formatting, or of text messages. A locale is defined by these language and cultural conventions. All of the information pertaining to culture and language is obtained at process run time. All locale information must be accessible to programs at run time so that data is processed and displayed correctly for your language and culture.

2.5.1.2 Understanding Locale

A locale is made up of the language, territory, and code set combination used to identify a set of language conventions. The LC environment variables and the LANG environment variable can be used in specifying the desired locale. The installation default locale refers to the locale selected at installation time by the system based on information given by the user. With this Information, the system sets the value of the default locale, specified by the LANG environment variable.

```
itso2(root) $ echo $LANG
De_DE
```

Figure 30. Show the Content of LANG Environment Variable

Every process uses this locale unless the LC or LANG environment variables are modified.

2.5.1.3 Changing Locale

The LC environment variables and the LANG environment variable can be used to specify the desired locale for your application.

```
itso2(root) $ export LANG=En_US

itso2(root) $ locale
LANG=En_US
LC_COLLATE=†En_US†
LC_CTYPE=†En_US†
LC_MONETARY=†En_US†
LC_NUMERIC=†En_US†
LC_TIME=†En_US†
LC_MESSAGES=†En_US†
LC_ALL=
```

Figure 31. Set Locale to En_US and Show New Locale

2.5.1.4 Locale Definition Files

System-defined locale definition files are provided with AIX. The /usr/lib/nls/loc directory contains the locale definition files for system-defined locales. With the command *locale -a* you can check which locale definition files are already installed on your system. These files are for example:

Locale	Language	Country	Code Set
de_DE	German	Germany	ISO8859-1
De_DE	German	Germany	IBM-850
de_CH	German	Switzerland	ISO8859-1
De_CH	German	Switzerland	IBM-850
en_US	English	United States	ISO8859-1
En_US	English	United States	IBM-850
el_GR	Greek	Greece	ISO8859-7
Fr_CA	French	Canada	IBM-850
tr_TR	Turkish	Turkey	ISO8859-9

Figure 32. Examples of Locales in AIX

2.5.1.5 AIX Locale Interpretation

An AIX locale consists of two parts: part1_part2.

- Part1 specifies the LANGUAGE. De or de stand for German. En or en stand for English. De with an upper case D specifies an IBM code set. De with a lower case d specifies an ISO code set.
- Part 2 specifies the TERRITORY. The territory part is always written in upper case letters. DE stands for Germany, CH for Switzerland.

So the locale De_DE specifies the German language spoken in Germany using the IBM code set 850. The locale de_CH specifies the German language spoken in Switzerland using ISO code set 8859-1.

For information about locales refer to ″Understanding Locale″ and ″Changing Your Locale″ in *AIX System Management Guide:Operating System and Devices*.

2.5.2 DB2 for AIX Environment

In this part we consider DB2 for AIX as a database server with database clients not connected via DRDA products. The clients are using CAE or DB2 on workstation platforms and are connected via TCP/IP, APPC, NETBIOS or IPX/SPX to the database server.

2.5.2.1 DB2 for AIX and AIX Locale

All language specific messages, help, panels and application defaults are stored independently of the programs and are translated in several languages and encoded in several code sets as for example:

```
German in Germany  IBM-850   (De_DE)
German in Germany  ISO8859-1 (de_DE)
English in US      IBM-850   (En_US)
Japanese in Japan  IBM-932   (Ja_JA)
```

See *DB2 for AIX Installation and Operation Guide* for more information.

You specify the language to be used in your DB2 environment by installing the message option for the desired language and then setting the LANG environment variable to the desired locale. If you want to use the Japanese messages for example, you must have the Ja_JA message option installed, the LANG environment variable set to LANG=Ja_JA and the LANG variable made known to the environment by the command: export LANG. Setting and exporting the LANG variable can be done within the profile of the user for the entire session or within a single window for only this process.

The selected DB2 message catalog options, as well as the messages for the installation scripts, are placed in the /usr/lpp/db2_02_01/msg/%L directories on the target workstation, where %L is equal to the locale name of the message catalog. Selected messages and resources of the DB2 Database Director are placed in the /usr/lpp/db2_02_01/msg/%L and the /usr/lpp/db2_02_01/dba/res/%L directories.

2.5.2.2 Setting up Application and Database NLS Environment

DB2 for AIX NLS support is based on specific code set and territory settings established when a database is created or when an application executes. The code set value is mapped to an IBM-defined code page and the territory is mapped to an IBM-defined country code.

Table 3 on page 38 shows only some of the countries and code sets supported by DB2 for AIX and how these values are mapped to country code and code page values which are used by the database manager.

Table 3. Supported Locales, Code Sets, Code Pages					
AIX 3.2 Locale	**AIX 4.1 Locale**	**Code Set**	**Code Page**	**Country Code**	**Country (Language)**
ar_AR	ar_AA	ISO8859-6	1089	785	Arabic Countries
Ar_AR	Ar_AA	IBM-850	850	785	Arabic Countries
C	C	IBM-850	850	001	Australia (English)
	pt_BR	ISO8859-1	819	055	Brazil (Portuguese)
de_DE	de_DE	ISO8859-1	819	049	Germany
De_DE	De_DE	IBM-850	850	049	Germany
ja_JP	ja_JP	IBM-eucJP	954	081	Japan (Japanese)
Ja_JP	Ja_JP	IBM-932	932	081	Japan (Japanese)
sp_YU	sr_SP	ISO8859-5	915	381	Serbia/ Montenegro (Cyrillic)
sh_YU	sh_SP	ISO8859-2	912	381	Serbia/ Montenegro (Latin)
si_SI	sl_SI	ISO8859-2	912	386	Slovenia
tr_TR	tr_TR	ISO8859-9	920	090	Turkey
en_US	en_US	ISO8859-1	819	001	U.S.A. (English)
En_US	En_US	IBM-850	850	001	U.S.A. (English)
Note: Refer to ″Country Code and Code Page Support″ in *DB2 for AIX Installation and Operation Guide* to see all the supported values.					

The code set (for example IBM-850) is a character string identifier for the code page and the territory (for example US) is a character string identifier for the country code. Each code page is identified by a numeric identifier (here 850). Each country is also represented by a numeric identifier (here 001). A code page is a mapping of each character from a character set, such as Latin alphabet, to a numeric representation. For example, code set IBM-850 code page 850 represents character ″A″ as hexadecimal 41. The country code setting establishes any national conventions that are supported by the database manager.

This information is used by the database manager during database creation and during database access.

During database creation, the database manager uses the territory and code set of the application which invokes the CREATE DATABASE command. If, within the CREATE DATABASE command the USING CODESET and TERRITORY option is specified, the database manager will use the code set and territory specified in the command. For internal processing, the database manager maps the territory value to a country code and the code set to an IBM-defined code page value. Country code and code page are also used to determine the default collating sequence that will be used for that database. (See 2.5.4, “Support for Alternate Collating Sequences” on page 51 for more information.)

When creating databases you need to either provide proper territory and code set specifications or ensure that the operating environment is set to meet your country′s application requirements.

All this NLS information (code page, country code, collating sequence) is stored in the database configuration file and cannot be changed.

While accessing the database, the application code page is compared to the code page of the database to determine if character data conversion is required or how the database manager has to transform the date and time presentation from the internal storage format to the appropriate country external format of the application.

2.5.2.3 Deriving Code Page Values

The application code page is derived from the active environment when the database connection is made.

The database code page is derived from the value specified (explicitly or by default) at the time the database is created.

In AIX-based environments, the active environment is determined from the locale environment variables, which include information about language, territory and code set.

The database manager determines code page attributes for all character strings when an application is bound to a database. The potential code page attributes are:

The Database Code Page
The database code page stored in the database configuration file. This code page value is determined when the database is created and cannot be altered.

The Application Code Page
The code page under which the application is executed. Note that this is not necessarily the same code page under which the application was bound.

Code Page 0
This represents a string that is derived from an expression that contains a FOR BIT DATA or BLOB value.

Character string code page attributes are as follows:

- Columns may be in the database code page or code page 0 (if defined as character FOR BIT DATA or BLOB).
- Constants and special registers (for example, USER, CURRENT SERVER) are in the database code page. Note that constants are converted to the database code page when an SQL statement is bound to the database.
- Input host variables are in the application code page.

A set of rules is used to determine the code page attributes for operations that combine string objects, such as the results of scalar operations, concatenation, or set operations. At execution time, code page attributes are used to determine any requirements for code page conversions of strings.

For more details on these set of conversion rules see the *DB2 SQL Reference* manual.

2.5.2.4 Deriving Locales

Both the database application and the database manager have to derive the locale they have to use during runtime. When we want to know how an application derives the locale we have to consider that there are two locales:

- The environment locale allows you to specify the language, currency symbol, and so on, that you want to use.
- The program locale contains the current language, currency symbol, and so on, of a program that is executing.

When a program is started, it gets a default C locale. It does **not** get a copy of the environment locale. Your program has a few choices to specify the program locale:

- Ignore the environment locale. The program could set the locale with the *setlocal()* function and so hardcode the value of the locale in the program.
- Copy the environment locale to the program locale.
- Ignore the environment locale and use whatever defaults it gets from the operating system.

Also DB2 has to derive a locale. The active locale used by DB2 is determined from the LC_CTYPE portion of the locale.

- If LC_CTYPE of the program locale has a value other than that of ″C″, DB2 will use this value to determine the application code page by mapping it to its corresponding code page.
- If LC_CTYPE of the program locale has the value of ″C″ (the C locale), DB2 will set the program locale according to the environment locale using the *setlocale()* function.
- If LC_CTYPE still has the value of ″C″, DB2 will use its default locale for that platform. See *DB2 SDK Building Your Application* for your platform for information on the default locale for that platform.
- If LC_CTYPE is no longer ″C″, its new value will be used to map to a corresponding code page.

It is strongly recommended that applications be pre-compiled, bound, compiled, and executed using the same code page. This is because data conversions by the server can occur in both the bind and the execution phases. Ensure that the same conversion tables are used by binding and executing with the same active code page. Any external data obtained by the application will be assumed to be in the application code page. This includes data obtained from a file or from user input. Make sure that data from sources outside the application uses the same code page as the application.

For more detailed considerations see *DB2 Application Programming Guide*, *DB2 SQL Reference* and *DB2 Installation and Operation Guide*.

2.5.2.5 Support for Different Code Pages

DB2 for AIX supports a lot of different code pages. Your DB2 for AIX database server or database client can use these code pages to store data and to run applications.

In Table 4 on page 41 you can see the code pages supported by DB2 Common Server and the character set groups to which they belong.

Table 4. Code Page Support

Character Set Group	Supported Code Pages
Group 1: Latin-1	437, 819, 850, 863, 1004, 1051, 1252, 1275
Group 1a: Arabic	864, 1046, 1089, 1256
Group 1a: Cyrillic	855, 866, 915, 1251, 1283
Group 1a: Latin-2	852, 912, 1250, 1282
Group 1a: Greek	813, 869, 1253, 1280
Group 1a: Hebrew	862, 916, 1255
Group 2: Japanese	932, 942, 943, 954
Group 2: Korean	949, 970
Group 2: Simpl. Chinese	1381, 1383
Group 2: Trad. Chinese	938, 948, 950, 964
Group 2: Thai	874
Group 1a: Turkish	857, 920, 1254, 1281

2.5.2.6 Conversion Between Different Code Pages

Ideally the application should always use the same code page as the database. However, this is not always practical or possible. The DB2 products provide support for character conversion that allows your application and database to use different code pages. Characters from one code page must be mapped to the other code page in order to maintain the meaning of the data.

Character conversion can occur in the following situations:

- When a client or application accessing a database is running in a code page that is different from the code page of the database. This conversion will occur on the database server machine for both conversions from the application code page to the database code page and from the database code page to the application code page.
- When a client or application importing a PC/IXF file runs in a code page that is different from the file being imported. This data conversion will occur on the database client machine before the client accesses the database server. Additional data conversion may take place if this application is running in a code page different from the code page of the database server.
- When DDCS for AIX is used to access data on a DRDA AS. In this case character conversion occurs on the receiver of the data, as defined by the DRDA rule. Refer to 2.5.3, “DB2 for AIX and DRDA Environment” on page 48.

Character conversion will **not** occur for:

- File names.
- Data that is targeted for or comes from a column assigned the FOR BIT DATA attribute, or data used in an SQL operation whose result is FOR BIT or BLOB data.
- Access to a DB2 for OS/2 Version 1.0 or Version 1.2 database server.
- A DB2 product or platform that does not support, or that does not have support installed, for the desired combination of code pages. In this case you will get the SQL0332N error message.

When your application converts from one code page to another, it is possible that one or more characters are not represented in the target code page. In this case, DB2 inserts a *substitution* character into the target string in place of the character that has no representation. The replaced character is then considered a valid part of the string. If that occurs, the SQLWARN10 indicator in the SQLCA is set to ″W″. This flag is not set after a substitution using the WCHARTYPE CONVERT precompiler option.

2.5.2.7 Supported Character Conversion

When data conversion occurs, conversion will take place from a source code page to a target code page. The source code page is determined from the source of the data:

- Data from the application has a source code page equal to the application code page.
- Data from the database has a source code page equal to the database code page.

The target code page is determined from where the data is to be placed. But also rules for intermediate operations are considered.

- If the data is moved directly from an application into a database without intervening operations, the target code page is the database code page.
- If the data is being imported into a database from a PC/IXF file, there are two character conversion steps.
  - From the PC/IXF file code page to the application code page.
  - From the application code page to the database code page.

  Exercise caution in situations where two conversion steps might occur. Make sure you follow the supported character conversions listed in Table 5 on page 43 to avoid a possible loss of character data.
- If the data is derived from operations performed on character data, data conversion is based on a set of rules. The source may be any of the application code page, the database code page, FOR BIT DATA, or for BLOB data. Some or all of the data items may have to be converted to an intermediate result, before the final target code page can be determined.

  See the *DB2 SQL Reference* manual for a summary of these rules and for specific applications with individual operators and predicates.

Table 5 on page 43 shows the supported code page conversions.

Table 5. Supported Code Page Conversions	
Code Pages	**Countries**
437, 819, 850, 863, 1004, 1051, 1252, 1275	Austria, Australia, Belgium, Brazil, Canada, Denmark, Finland, France ,Germany, Iceland, Italy, Latin America, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, UK, USA
813, 869, 1253, 1280	Greece
852, 912, 1250, 1282	Croatia, Czech Republic, Hungary, Poland, Romania, Serbia/ Montenegro (Latin), Slovakia, Slovenia
855, 915, 1251, 1283	Bulgaria, FYR Macedonia, Serbia/Montenegro (Cyrillic)
857, 920, 1254, 1281	Turkey
862, 916, 1255	Israel
864, 1046, 1089, 1256	Arabic countries
866, 915, 1251, 1283	Russia
932, 942, 943, 954	Japan
938, 948, 950, 964	Taiwan
949, 970	Korea
1381, 1383	People's Republic of China
Note: For special information about these code pages and languages refer to *DB2 for AIX Installation and Operation Guide*	

These conversions are supported by DB2 for AIX if the database client or database application is also running on a workstation platform.

Any code page can be converted to any other code page that is listed in the same row of the table.

If all the code pages of both the DB2 for AIX database server and of all (remote) database clients or database applications are in the same group, the DB2 for AIX products support the code page conversion between all of them if required.

For example: If the database server and all clients are using code page 850, no conversion is required.

If the database server uses code page 850 and a database client uses code page 437, the code page conversion is done by the database manager.

If the database server uses code page 850 and a database client uses code page 852, then there is no code page conversion available at this time and you will get SQL error SQL0332N when connecting.

2.5.2.8 DB2 for AIX Example

In the following example you can see how AIX locales and DB2 for AIX are working together and how to influence the dependencies.

In this example we use the locale tr_TR for Turkish, because Turkish is not a member of the character set group Latin-1. Therefore, DB2 for AIX does not do an automatic code page conversion from code page 850 to code page 920 when accessing the database but has to be set to the correct environment.

To see the possibilities and problems within this constellation just go step by step.

Installed AIX Locales

```
 itso1(root) $ locale -a
en_US.ISO8859-1
En_US.IBM-850
de_DE.ISO8859-1
De_DE.IBM-850
Ja_JP.IBM-932
En_JP.IBM-850
pt_PT.ISO8859-1
Pt_PT.IBM-850
tr_TR.ISO8859-9
```

Figure 33. Show which Locales are Installed

Active AIX Locale

```
itso1(root) $ echo $LANG
En_US
```

Figure 34. Show the Active Locale

Create Database enus

```
itso1(db2inst1) $  db2 †create database enus†
```

Figure 35. Create Database with Active Locale

Create Database trtr

```
itso1(db2inst1) $
  db2 †create database trtr using codeset ISO8859-9 territory TR†
```

Figure 36. Create Database with Locale tr_TR

List Database Directory

```
itso1(db2inst1) $ db2 †list database directory†

 System Database Directory

 Number of entries in the directory = 2

Database 1 entry:

 Database alias                    = ENUS
 Database name                     = ENUS
 Local database directory          = /home/db2inst1
 Database release level            = 6.00
 Comment                           =

Database 2 entry:

 Database alias                    = TRTR
 Database name                     = TRTR
 Local database directory          = /home/db2inst1
 Database release level            = 6.00
 Comment                           =
 Directory entry type              = Indirect
```

Figure 37. List Database System Directory

Database Configuration Files

```
itso1(db2inst1) $ db2 †get database configuration for enus†


      Database Configuration for Database enus

 Database configuration release level          = 0x0600
 Database release level                        = 0x0600

 Database territory                            = En_US
 Database code page                            = 850
 Database code set                             = IBM-850
 Database country code                         = 1
 ........
```

Figure 38. Database Configuration File for enus Database

```
itso1(db2inst1) $ db2 †get database configuration for trtr†


     Database Configuration for Database trtr

 Database configuration release level         = 0x0600
 Database release level                       = 0x0600

 Database territory                           = tr_TR
 Database code page                           = 920
 Database code set                            = ISO8859-9
 Database country code                        = 90
 ........
```

Figure 39. Database Configuration File for trtr Database

Show Active Locale Variables

```
itso1(db2inst1) $ locale
LANG=En_US
LC_COLLATE=†En_US†
LC_CTYPE=†En_US†
LC_MONETARY=†En_US†
LC_NUMERIC=†En_US†
LC_TIME=†En_US†
LC_MESSAGES=†En_US†
LC_ALL=
```

Figure 40. Show the Currently Used Locale

Connect to the Databases

```
itso1(db2inst1) $ db2 †connect to enus†

Database Connection Information

Database product       = DB2/6000 2.1.0
SQL authorization ID   = DB2INST1
Local database alias   = ENUS
```

Figure 41. Connection Information

```
itso1(db2inst1) $ db2 †connect to trtr†

SQL0332N  There is no available conversion
for the source code page †850† to
the target code page †920†.  Reason Code †1†.
SQLSTATE=57017
```

Figure 42. Connection information

Change the Active Locale

Now change the active locale to the equivalent for database trtr.

```
itso1(db2inst1) $ export LANG=tr_TR

itso1(db2inst1) $ locale
LANG=tr_TR
LC_COLLATE=†tr_TR†
LC_CTYPE=†tr_TR†
LC_MONETARY=†tr_TR†
LC_NUMERIC=†tr_TR†
LC_TIME=†tr_TR†
LC_MESSAGES=†tr_TR†
LC_ALL=
```

Figure 43. Change and Show Active Locale

Connect to the Databases Again

```
itso1(db2inst1) $ db2 †connect to enus†

Database Connection Information

Database product       = DB2/6000 2.1.0
SQL authorization ID   = DB2INST1
Local database alias   = ENUS
```

Figure 44. Connection Information

```
itso1(db2inst1) $ db2 †connect to trtr†

SQL0332N  There is no available conversion
for the source code page †850† to
the target code page †920†.  Reason Code †1†.
SQLSTATE=57017
```

Figure 45. Connection Information

Terminate DB2 CLP

Even though we changed the locale to the appropriate one we still get SQL error SQL0332N when connecting to database trtr. The reason is that we are using DB2 CLP as the application to connect to the databases. But DB2 CLP is still running on code page IBM-850. To make the new locale available to DB2 CLP we have to terminate CLP and start again.

```
itso1(db2inst1) $ db2 †terminate†
```

Figure 46. Terminate CLP

Connect to the Databases Again

```
itso1(db2inst1) $ db2 † connect to enus†

 SQL0332N  There is no available conversion
 for the source code page †920† to
 the target code page †850†.  Reason Code †1†.
 SQLSTATE=57017
```

Figure 47. Connection Information

```
itso1(db2inst1) $ db2 † connect to trtr†

   Database Connection Information

  Database product       = DB2/6000 2.1.0
  SQL authorization ID   = DB2INST1
  Local database alias   = TRTR
```

Figure 48. Connection Information

Now DB2 CLP is running on code page 920. So connecting to database trtr is successful and connecting to database enus fails.

2.5.3 DB2 for AIX and DRDA Environment

Now consider the case when DB2 for AIX is a DRDA application requester using DDCS for AIX to connect to a DRDA application server. The DRDA AR and the DRDA AS might be running on different code pages or CCSIDs. In this case data conversion has to take place. Within a DRDA environment the receiver of data is always responsible for the correct data conversion.

DB2 for AIX also can act as a DRDA AS. In that case DB2 for AIX might deliver data to a DRDA AR on MVS.

2.5.3.1 Conversion of CCSIDs with DRDA

Different systems or products represent data in different ways. When data is moved from one system to another, data must sometimes be converted. Products supporting DRDA automatically perform any necessary conversions at the receiving system.

Both numeric data and character data might require conversion. All numeric conversions are built into the database products. However, because of their large number not all character conversions are supported by the database products.

Character sets, code pages, and encoding schemes are identified by coded character set identifiers (CCSIDs). Each system or product has an associated CCSID value.

The communication network delivers the DRDA messages sent by the AR. The DRDA messages describe the SQL operations to be performed, and they are transmitted using the DRDA application support protocols. Each data object included in the DRDA messages contains a descriptor indicating the internal representation of the data. DRDA specifies that the system sending a data object is allowed to transmit data using its own internal data representation. If the data object is in a format that is not acceptable to the receiver, the receiver must perform any required conversions.

For example, DRDA allows machines using ASCII character representation to send messages to machines using EBCDIC character representations. When this occurs, the system receiving the messages is responsible for performing the correct data conversion. When the DRDA messages are interpreted, the application server performs the requested SQL operations at the local database and sends the results back to the application requester. As a receiver of data, the DRDA AR machine may also have to perform the correct data conversions for the results sent by the DRDA AS.

2.5.3.2 Supported Conversions with DRDA

Table 6 shows the conversions that are supported within a DRDA environment. It shows the available conversions between the code pages on the workstation and the CCSIDs on the hosts.

Table 6. Workstation Code Page to Host CCSID Conversion

Character Set Group	EBCDIC CCSID	ASCII Code Page
Group 1: Latin-1	037, 273, 277, 278, 280, 284, 285, 297, 500, 871	437, 819, 850, 1051, 1252, 1275
Group 1a: Arabic	420	864, 1046, 1089, 1256
Group 1a: Cyrillic	1025	855, 866, 915, 1251, 1283
Group 1a: Latin-2	870	852, 912, 1250, 1282
Group 1a: Greek	423, 875	813, 869, 1253, 1280
Group 1a: Hebrew	424	862, 916, 1255
Group 2: Japanese	930, 931, 939, 5026, 5035	932, 942, 943, 954
Group 2: Korean	933	949, 970
Group 2: Simpl. Chinese	935	1381, 1383
Group 2: Trad. Chinese	937	938, 948, 950, 964
Group 2: Thai	838	874
Group 1a: Turkish	1026	857, 920, 1254, 1281

Note: For further information about CCSID support for DB2 for AIX refer to *DB2 for AIX Installation and Operation Guide* or *DDCS Installation and Configuration Guide*.

Using DRDA products you have conversion support between the EBCDIC CCSID and the corresponding ASCII code page in the same line. But there is no support between EBCDIC CCSID and ASCII code pages in different lines.

For example: If the DRDA AR on the workstation uses ASCII code page 850 and the DRDA AS on MVS uses EBCDIC code page 500 then the required conversion will be done by the DRDA products. The conversion is always done on the receiver of the data.

If the DRDA AR on the workstation uses ASCII code page 850 and the DRDA AS on MVS uses EBCDIC code page 870 then the required conversion cannot be done by the DRDA products at this time. You may get the SQL error SQL30073N with parameter-identifier ″119C″ specifying that the target server does not support the CCSID requested by the application requester.

You should ensure that the CCSIDs used by the application server and the AR are compatible (that is, each system can convert from the other′s CCSID into its own CCSID). If the CCSIDs are not compatible, then the two systems will not connect successfully.

In general, data can be converted from code page to CCSID and back again with no change. The following are the only exceptions to this rule:

- In double-byte character set (DBCS) code pages, some data containing user-defined characters may be lost.
- For single-byte code pages defined within mixed-byte code pages, some characters may be mapped to substitution characters and they are lost when the data is converted back to the original code page.

In order to connect to a database with an incompatible code page, the following steps need to be performed. Although this information is correct at this time (Nov ′96), you should always check your DB2 manuals first, to see if there is more up-to-date information applicable to the same topic.

1 Make sure you are using DB2 server V2.1.2 or above.

2 Ask your IBM representative to obtain two conversion files for the pair of code pages that you want to support. For example, if you want to support connection between code page 819 and code page 915, you need 08190915.cnv and 09150819.cnv tables.

3 Place these binary files in the *<DB2Path>/sqllib/conv* directory on UNIX platforms such as AIX, and in the *<DB2Path>/sqllib/conv* directory on various Intel platforms such as OS/2 or Windows.

4 If you are only using Common Server DB2 C/S products, you need to place these files only on the server.

5 For connections between Common Server DB2 C/S and mainframe DB2 implementations, such as SQL/DS or DB2/MVS, you need to first ensure that the corresponding DB2 host platform supports the same conversion. You only need the Host-to-Workstation conversion table on the workstation side.

6 Once the connection between incompatible code page environments is established, you will get a warning message:

```
SQL0863W  A successful connection was made, but only single byte
characters should be used.  SQLSTATE=0153.
```

Figure 49. Connection Information

This is normal, and intended to warn you that you should only use the characters that exist in both code pages.

When data is transferred between a DDCS workstation and a DRDA server, it is usually converted from a workstation code page to a host CCSID, and vice versa.

If the two systems use different code pages or CCSIDs, code points are mapped from one code page or CCSID to the other. This conversation is based on the underlying CDRA and is always performed at the receiver of the data.

For example, if DDCS for AIX is used to access data from a DB2 for MVS DRDA server, the following happens:

- DDCS sends SQL statements and input data from AIX to MVS
- DB2 for MVS DRDA AS converts the data to an EBCDIC CCSID and processes it
- DB2 for MVS sends the resulting data back to DDCS for AIX
- DDCS for AIX converts the resulting data to an ASCII or ISO code page and returns it to the user/application

2.5.4 Support for Alternate Collating Sequences

The database manager compares character data using a *collating sequence*. This is an ordering for a set of characters that determines whether a particular character sorts higher, lower, or the same order. Note: Character string data defined with the FOR BIT DATA attribute, or BLOB data, is sorted using the binary sort sequence.

The country code and the code page are used to determine the default collating sequence that will be used for the database. This default collating sequence may be explicitly overwritten. For more information refer to the CREATE DATABASE command or **sqlecrea** API below, about how to overwrite the collating sequence.

All this NLS information (code page, country code, collating sequence) is stored in the database configuration file and cannot be changed since it is used for additional NLS processing. This includes, for example to generate and process indexes on character data type columns, to process character data string comparisons or to process SORT BY and GROUP BY clauses in SQL statements.

2.5.4.1 Defining Collating Sequence with API sqlecrea

You can use the API *sqlecrea* to create a database specifying at creation time the code set, territory and collating sequence information. Therefore, you can use the API parameters:

pCountryInfo

Input parameter. May be NULL.

A pointer to the *sqledbcountryinfo* structure, containing the locale and the code set for the database.

For more information about this structure refer to the *DB2 API Reference* manual.

pDBDescriptor

Input parameter. May be NULL.

A pointer to the database descriptor block used when creating the database. The database description block may be used to supply values that are permanently stored in the configuration file of the database, such as collating sequence.

For more information about the database description block structure SQLEDBDESC refer to the *DB2 API Reference* manual.

The CREATE DATABASE operation assigns the code set and the territory defined by *pCountryInfo* in the creating application as the database code set and territory, and stores these, as well as the collating sequence from *pDBDescriptor*. If no code set or territory is specified, the default action is to use the locale of the application which is making the **sqlecrea** call to determine the code set or territory. A flag is set in the database configuration file if the collating sequence consists of unique weights, or if it is the identity sequence.

2.5.4.2 Defining Collating Sequence with CREATE DATABASE Command

You can specify the CREATE DATABASE command to create a new database defining the code set, territory and collating sequence information at creation time.

To specify the collating sequence information you have two possibilities:

CREATE DATABASE *databasename* USING CODESET *codeset* TERRITORY *territory* COLLATE USING SYSTEM

CREATE DATABASE *databasename* USING CODESET *codeset* TERRITORY *territory* COLLATE USING IDENTITY

COLLATE USING identifies the type of collating sequence to be used for the database. Once the database has been created the collating sequence cannot be changed.

The types you can choose are:

IDENTITY Identity collating sequence, in which strings are compared byte for byte

SYSTEM Collating sequence based on the current territory

2.5.4.3 Samples of User-Defined Collating Sequences

The differences between EBCDIC and ASCII cause differences in sort orders in the various database products, and also affect ORDER BY and GROUP BY clauses. One way to minimize these differences is to create a user-defined collating sequence that mimics the EBCDIC sort order. You can specify a collating sequence only when you create a new database. For more information see the *DB2 API Reference* and the *DB2 Command Reference* manuals.

The following sample user-defined collating sequences are available with DB2 for AIX in the host language include files:

sqle819a If the code page of the database is 819 (ISO Latin1), this sequence will cause sorting to be performed according to the host CCSID 500 (EBCDIC International).

sqle819b If the code page of the database is 819 (ISO Latin1), this sequence will cause sorting to be performed according to the host CCSID 037 (EBCDIC English).

sqle850a If the code page of the database is 850 (ISO Latin1), this sequence will cause sorting to be performed according to the host CCSID 500 (EBCDIC International).

sqle850b If the code page of the database is 850 (ISO Latin1), this sequence will cause sorting to be performed according to the host CCSID 037 (EBCDIC English).

sqle932a If the code page of the database is 932 (ASCII Japanese), this sequence will cause sorting to be performed according to the host CCSID 5035 (EBCDIC Japanese).

sqle932b If the code page of the database is 932 (ASCII Japanese), this sequence will cause sorting to be performed according to the host CCSID 5026 (EBCDIC Japanese).

These sample collating sequences are provided (as include files) to facilitate database creation using EBCDIC collating sequences instead of the default workstation collating sequence.

They can also be used as models for the construction of other user-defined collating sequences.

For more information about creating your own collating sequence see the *DB2 Application Programming*, the *DB2 API Reference* and the *DB2 SQL Reference* manuals.

Chapter 3. VM/ESA Implementation

3.1 General Consideration

The software installation for both ITSO and ZPIZ and the steps to establish DRDA connectivity are described in detail in the following chapter. The tasks to implement SQL/DS for VM as both a DRDA application server (AS) and a DRDA application requester (AR) are described. These tasks do not change regardless of the related client/server partner.

The operating system was VM/ESA V1.2.2 with the required software for DRDA implementation, that is SQL/DS 3.5, VTAM V3.4 and AVS/VM.

The steps undertaken at the ITSO and ZPIZ sites at this point were:

1. Install SQL/DS V3.5 for VM on the ITSO BOEVMIS2 system, and generate databases SQLVMDB0, SQLVMDB1, SQLVMDB6.

   Install SQL/DS V3.5 for VM on the ZPIZ VM122 system, and generate database SQLDBA.

2. Install SQL/DS V3.5 for VSE on the ITSO VSEANL13 system, and generate database SQLVSDB1.

   Install SQL/DS V3.5 for VSE on the ZPIZ VSE3 system, and generate database SQLDS.

3. Use a previously customized CICS partition for ITSO.
4. Install a new CICS partition on the VSE3 system for ZPIZ.
5. Check connectivity between both SQL/DS database environments using AVS/VM, VTAM and CICS.
6. Check connectivity from ITSO SQLUSER AR to the VSEANL13 system.
7. Check connectivity from ZPIZ SQLUSER AR to the VSE3 system.
8. Create the ISQL package on VSEANL13 for use from ITSO SQL/DS AR.
9. Create the ISQL package on VSE3 for use from ZPIZ SQL/DS AR.
10. Check connectivity from OS/2 DB2/2 to SQL/DS for VM and SQL/DS for VSE.

ZPIZ used the default database names and sizes as provided with the SQL/DS installation. The SQL/DS database for VM was defined as a global resource to VM. The major difference between ZPIZ and ITSO was, that at the ZPIZ site Latin-2 CCSID (870) was implemented as opposed to the default CCSID of 500 on the ITSO system.

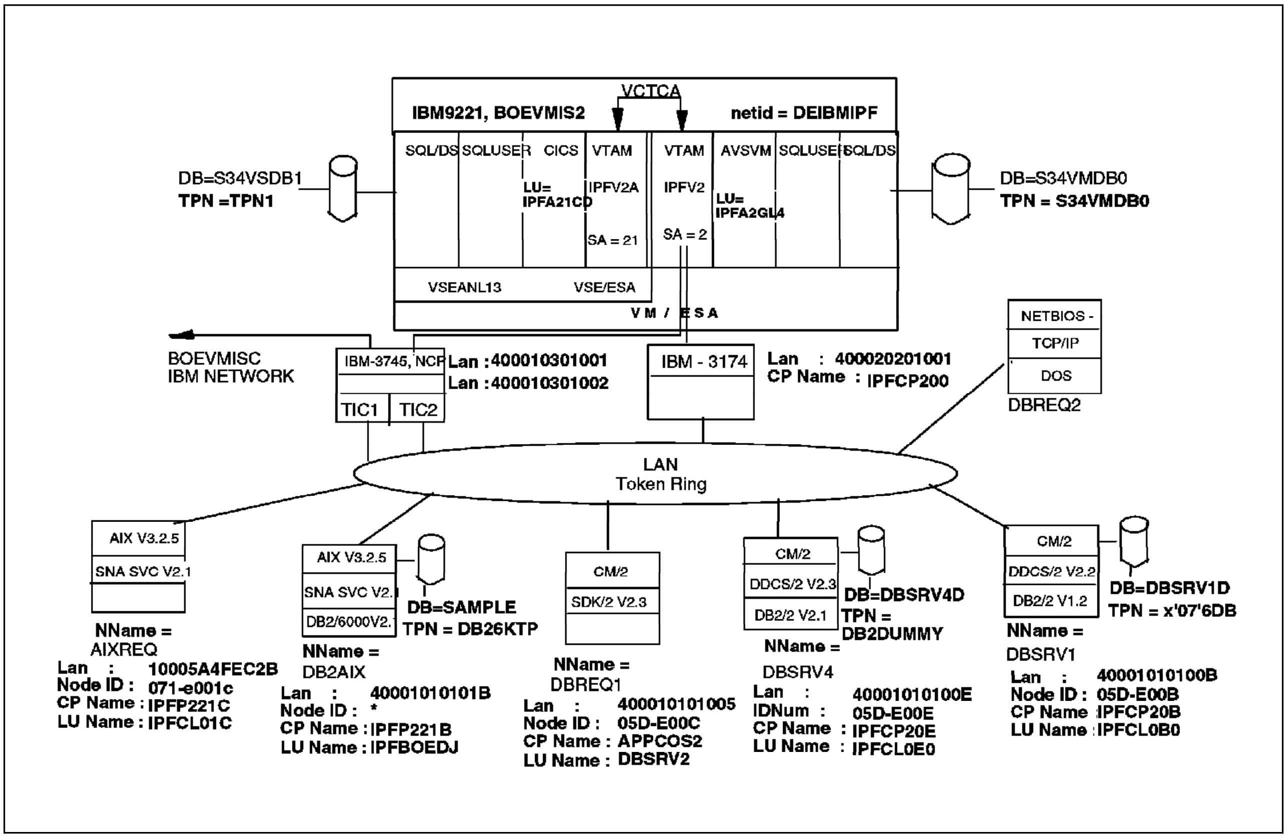


Figure 50. Configuration - ITSO

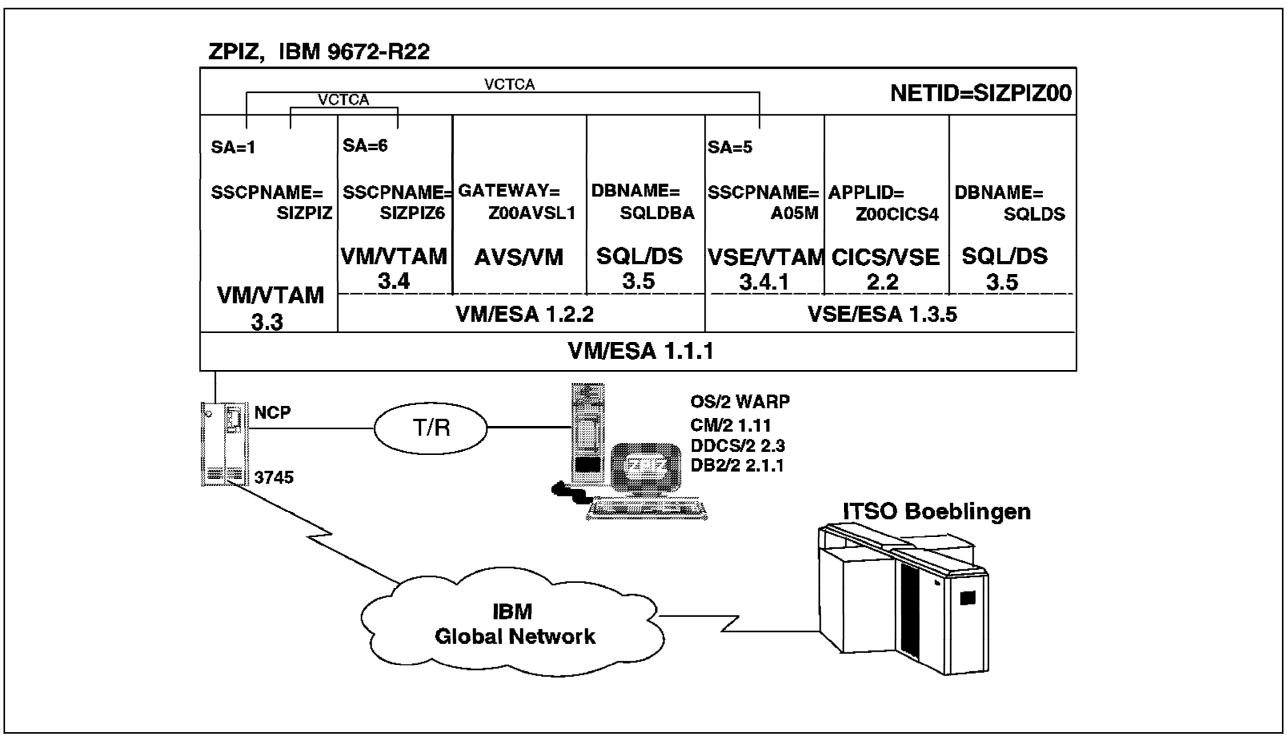


Figure 51. Configuration - ZPIZ

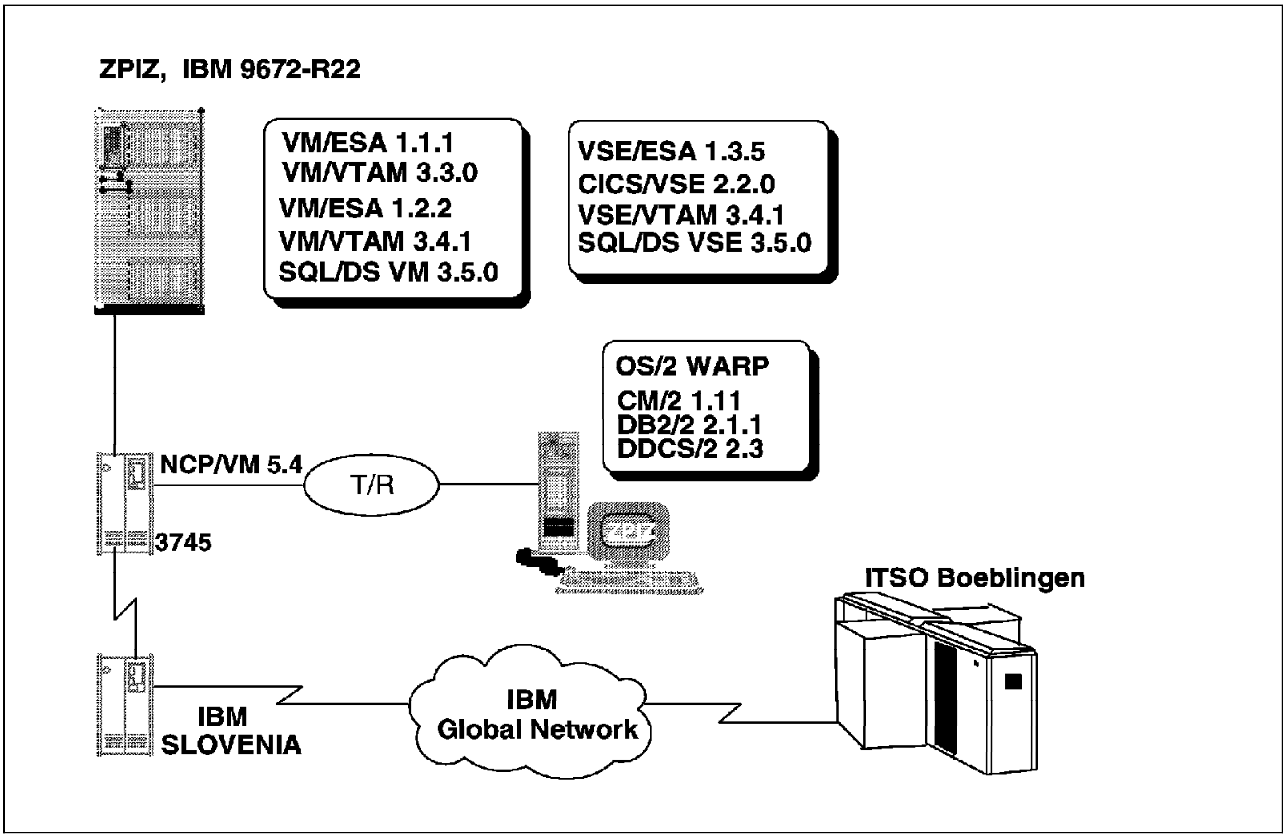


Figure 52. Software - ZPIZ

3.2 ITSO and ZPIZ System Setup

How the VM systems at ITSO and ZPIZ were set up for the SQL/DS database machines will be shown on the next pages.

3.2.1 Database Machine SQLUSER (DRDA AR)

The SQLUSER entry used on the ITSO or ZPIZ machine provides the AR function of DRDA to other systems (that is, like or unlike database systems).

3.2.2 UCOMDIR NAMES File

This is the resource directory, residing in the requester. It lists the resources as they are known to the requester and the information about how to get to them. There are two levels of communication directories. The SCOMDIR NAMES file contains system-wide information, which can be supplemented or overridden by the UCOMDIR NAMES file for an individual virtual machine.

Requesting the use of a remote resource means telling the system which resource is needed, which gateways to use to access it, what security information needs to be provided, and what type of session is required. This information is provided by a COMDIR file.

A COMDIR file contains one entry for each resource (such as a database) that will be accessed. Both local and remote resources may have COMDIR entries.

```
:NICK.SQLVSDB1
              :TPN.TPN1
              :LUNAME.IPFA2GL4 IPFA21CD
              :MODENAME.IBMRDBM
              :SECURITY.PGM
              :USERID.CICSUS1
              :PASSWORD.CICSUS1
              :DBNAME.SQLVSDB1
:NICK.SQLDBA
              :TPN.SQLDBA
              :LUNAME.IPFA2GL4 Z00AVSL1
              :MODENAME.IBMRDB
              :SECURITY.PGM
              :USERID.Z00VMU1
              :PASSWORD.RGNCSQ
              :DBNAME.SQLDBA
:NICK.SQLDS
              :TPN.TPN1
              :LUNAME.IPFA2GL4 Z00CICS4
              :MODENAME.IBMRDB
              :SECURITY.PGM
              :USERID.Z00VSEU1
              :PASSWORD.RGNCSQ
              :DBNAME.SQLDS
```

Figure 53. UCOMDIR NAMES File for the AR SQLUSER at ITSO

```
:NICK.SQLDS
              :TPN.TPN2
              :LUNAME.Z00AVSL2 Z00CICS4
              :MODENAME.IBMRDB
              :SECURITY.PGM
              :USERID.Z00VSEU1
              :PASSWORD.RGNCSQ
              :DBNAME.SQLDS
:NICK.SQLVSDB1
              :TPN.TPN1
              :LUNAME.Z00AVSL1 IPFA21CD
              :MODENAME.IBMRDB
              :SECURITY.PGM
              :USERID.CICSUS1
              :PASSWORD.CICSUS1
              :DBNAME.SQLVSDB1
:NICK.SQLVMDB0
              :TPN.SQLVMDB0
              :LUNAME.Z00AVSL1 IPFA2GL4
              :MODENAME.IBMRDBM
              :SECURITY.PGM
              :USERID.DRDAX1
              :PASSWORD.SQL4VSE
              :DBNAME.SQLVMDB0
```

Figure 54. UCOMDIR NAMES File for the AR SQLUSER at ZPIZ

Note: After the changes are made to a COMDIR, a SET COMDIR RELOAD command must be issued, or the SQLUSER must log off and log on, to activate those changes.

For example:

```
SET COMDIR RELOAD USER
SET COMDIR RELOAD SYSTEM
```

Figure 55. SET COMDIR RELOAD

3.2.3 Database Machine SQLVMDB0 and SQLMACH (DRDA AS)

```
USER SQLVMDB0 xxxxx 32M 64M G
   ACCOUNT SQLVMDB0
   OPTION MAXCONN 40
   IUCV *IDENT SQLVMDB0 GLOBAL
   IUCV ALLOW
```

Figure 56. USER DIRECT Entry for the AS SQLVMDB0 at ITSO

```
USER SQLMACH xxxxx 8M 12M G
   ACCOUNT SQLMACH
   OPTION MAXCONN 40
   IUCV *IDENT SQLDBA GLOBAL
   IUCV ALLOW
```

Figure 57. USER DIRECT Entry for the AS SQLMACH at ZPIZ

The SQLMACH entry used on the ZPIZ machine, for example, provides the AS function of DRDA to other systems.

The entry OPTION MAXCONN 40 specifies the total number of possible connections to this virtual machine.

The IUCV *IDENT SQLDBA GLOBAL statement authorizes the database machine, as a resource owner, to connect to the VM system service *IDENT. The name of the database specified in the DBNAME parameter during execution of the generation EXEC SQLDBINS is used as the resource identifier. The parameter SQLDBA allows the database machine to identify the AS. The GLOBAL parameter identifies this database machine as an AS resource which can be accessed by all ARs in a network.

3.3 ITSO and ZPIZ Database Setup

3.3.1 SQL/DS for VM Startup

The parameters for the SQLSTART exec which have an effect on the behavior of the database within DRDA are presented as follows:

PARAMETER	**ITSO**	**ZPIZ**
DBMODE=G | L | N	(see *IDENT IN VM Directory)	(see *IDENT in VM Directory)
PROTOCOL=SQLDS | AUTO	AUTO	AUTO
CHARNAME=name	INTERNATIONAL	SLOVENIA
CCSID=name	500	870

Table 7. ITSO and ZPIZ SQL/DS DRDA Initialization Parameters. The SQL/DS initialization parameters which have an impact on the behavior of the database within DRDA.

The various parameters are briefly described below:

DBMODE The DBMODE parameter identifies the database name as a LOCAL resource (DBMODE=L), a GLOBAL resource (DBMODE=G), or NON-APPC/VM (DBMODE=N). Consider the following when specifying the DBMODE parameter:

- When DBMODE=G is specified, and the IUCV *IDENT directory entry does not allow that resource name to be identified as a GLOBAL resource, the AS ends the startup.
- If DBMODE=L is specified, the AS is identified as a LOCAL resource even if the IUCV *IDENT entry specifies that resource as GLOBAL.
- If you specify L or G to run SQLSTART, and the database machine contains no GLOBAL or LOCAL IUCV *IDENT directory entries, the database manager ends processing.

PROTOCOL The PROTOCOL parameter specifies the types of protocols that the application server can handle. This parameter has two options on the SQLSTART EXEC:

PROTOCOL=AUTO

This option allows access to the SQL/DS AS from SQL/DS AR and non-SQL/DS AR.

PROTOCOL=SQLDS

This option allows access to the SQL/DS AS from SQL/DS AR only. This is the default.

In SQL/DS - SQL/DS communications, if PROTOCOL=AUTO is specified on both the AR and the AS, the initialization flows (handshaking) will be done in DRDA. However, after this handshaking is completed the communications will automatically revert to SQLDS-only (private) flows.

If the database acts both as DRDA AS and DRDA AR then PROTOCOL=AUTO must be specified on the SQLSTART. If the database is only going to communicate with other SQL/DS database systems, then PROTOCOL=SQLDS may be used. In this case private VM protocols will be used.

CHARNAME The CHARNAME parameter specifies the CCSIDs to be used as the AS defaults. The default CCSIDs determine the character sets and code pages to be used to interpret statements and return results. The valid CHARNAME values you can specify are for example ENGLISH (CCSID=37), INTERNATIONAL (CCSID=500), 870 (CCSID=870), and all the values that are in the CHARNAME column of the SYSTEM.SYSCCSIDS catalog table.

The SQLDBGEN parameter “DBNAME” will be the resource ID registered by *IDENT and therefore known to AVS. COMDIR entries pointing to that resource will use that DBNAME. If the DBNAME is longer than eight bytes, then the resource ID will be specified as something other than the DBNAME. In this case, mapping is performed by the file RESID NAMES when the database is started.

3.3.2 SQLINIT Usage

The SQLINIT EXEC initializes a user machine for SQL/DS access. With this EXEC, we specify the default database we wish to access and any special options to be used when accessing this database. If we want to access another database that is not the default established through SQLINIT, then we can issue the SQL command CONNECT to access the other database.

The SQLINIT parameters:

1. PROTOCOL

   Indicates the database access protocol to be used for communicating with the application server. Valid values here are SQLDS, AUTO, and DRDA.

   SQLDS specifies that only private flows will be used. If this option is specified, the AR cannot connect to a non-SQL/DS AS because SQL/DS-only protocol will be used. This protocol performs better and uses less storage than SAA Remote Unit of Work (DRDA) protocols. SQLDS is the default value.

   AUTOmatic specifies that the AR is to use private flows when communicating with an SQL/DS AS and SAA Remote Unit of Work (DRDA) flows when communicating with a non-SQL/DS AS.

   DRDA specifies that the AR has to use the DRDA flow with both SQL/DS and non-SQL/DS AS.

   Below is an example of PROTOCOL parameter usage.

```
sqlinit db(sqldba) protocol(sqlds) charname(slovenia)
ARI0717I Start SQLINIT EXEC: 06/25/96 14:54:35 UTC.
ARI0320I The default server name is SQLDBA.
ARI0796I End SQLINIT EXEC: 06/25/96 14:54:36 UTC.
```

2. QRYBLKSIZE

   Is the block size used to return rows of data when SAA Remote Unit of Work blocking is used to perform FETCHes. The number is specified in denominations of 1K and ranges anywhere between 1K and 32K. This parameter has an impact only when DRDA flows are being used, if the private flow is used it will be ignored. The default value is 8K.

3. CHARNAME

   This parameter specifies the name of the default character set for the user′s virtual machine (application requester). The valid values are contained in the new catalog table, SYSTEM.SYSCCSIDS. The specified CHARNAME value and its corresponding row in this table are used to set the default CCSID values for single-byte, mixed, and graphic data.

   The default for CHARNAME is INTERNATIONAL for ITSO. This corresponds to CCSID 500, which is a single-byte CCSID. ZPIZ′s CHARNAME is 870. This corresponds to CCSID 870.

3.3.3 Defining Slovenian Character Set

At the beginning of this section we would like to point out that a similar implementation can be done for other countries of Eastern Europe (Croatia, Poland, Czech...). When the user starts the database with CHARNAME = 870 and initializes the database with CHARNAME = SLOVENIA lower case letters are translated to upper case letters, if SET CASE UPPER is requested.

```
a (X¢81¢) becomes A (X¢C1¢)
b (X¢82¢) becomes B (X¢C2¢)
c (X¢83¢) becomes C (X¢C3¢)
d (X¢84¢) becomes D (X¢C4¢)
```

But some SLOVENIAN letters have not been converted:

```
å (X¢47¢) remains å (X¢47¢)
æ (X¢9C¢) remains æ (X¢9C¢)
¶ (X¢B6¢) remains ¶ (X¢B6¢)
```

On the ZPIZ side a new record was inserted into the SYSTEM.SYSCHARSET system table, which translates these three Slovenian letters into capitals. The new character set was set up by referring to the chapter ″Choosing a National Language and Defining Character″ in *SQL/DS System Administration*, GH09-8096. The attributes for these letters were changed from 0 to 3 and lower case letters were overtyped by capital letters. The result:

```
å (X¢47¢) becomes Å (X¢67¢)
æ (X¢9C¢) becomes ′ (X¢BC¢)
¶ (X¢B6¢) becomes ½ (X¢B8¢)
```

Now, when the user starts the database with CHARNAME = 870 and initializes the database with CHARNAME = SLOVENIA, the translation of all lower case letters to upper case is done correctly.

```
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
SET ERRORMODE CONTINUE;
COMMENT ¢*****--> BEGIN SLOVENIA INPUT COMMANDS <--******¢
INSERT INTO SYSTEM.SYSCHARSETS (NAME,CHARCLASS,CHARTRANS)
VALUES(
¢SLOVENIA¢,
¢100000030305666660000000000636666660000303066B66000000000000633267033333 3
333003000003333333330030000033333333003000600000303000300000AAAAAAAAA00000
00AAAAAAAAA00000000AAAAAAAA00000044444444444000000¢,
¢ âäàáãÅçÑ¢.<(+|&éêëèíîïìß!$*); -/ÂÄÀÁÃÅÇÑ],%_>?øÉÊËÈÍÎÏÌ:#@¢¢=†ØABCDE
FGHI«»œ-•• JKLMNOPQR–Œ’ŸÆ••STUVWXYZ¡¿œ•‡„[£¥fiﬂ§½¼½¾•‘’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
COMMIT WORK;
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
SET ERRORMODE CONTINUE;
COMMENT ¢*****--> BEGIN ENGLISH INPUT COMMANDS <--******¢
INSERT INTO SYSTEM.SYSCHARSETS (NAME,CHARCLASS,CHARTRANS)
VALUES(
¢ENGLISH¢,
¢1000000000005666660000000000636666660000000000066B66000000000000633267033333 3
3330000000033333333300000000033333333000000600000000000000000AAAAAAAAA00000
00AAAAAAAAA00000000AAAAAAAA00000044444444444000000¢,
¢ âäàáãåçñ¢.<(+|&éêëèíîïìß!$*); -/ÂÄÀÁÃÅÇÑ],%_>?øÉÊËÈÍÎÏÌ:#@¢¢=†ØABCDE
FGHI«»‰-•• JKLMNOPQR–ŒæŸÆ••STUVWXYZ¡¿œ•‡„[£¥fiﬂ§¶¼½¾•‘’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
```

Figure 58. SLOVENIA Character Set at ZPIZ

If the SQL/DS user starts the database with CHARNAME = 870, initializes the database with CHARNAME = SLOVENIA and tries to sort inserted records, the collating sequence is as follows:

```
Å (X¢67¢)
½ (X¢B8¢)
’ (X¢BC¢)
A (X¢C1¢)
B (X¢C2¢)
C (X¢C3¢)
D (X¢C4¢)
```

A new collating sequence for SLOVENIA characters was implemented as well. The sample coding for FIELD PROCEDURE SLOSORT is presented in A.3, “Field Procedure Sample” on page 199. The result of the new sort sequence is:

```
A (X¢C1¢)
B (X¢C2¢)
C (X¢C3¢)
Å (X¢67¢)
D (X¢C4¢)
 -
S (X¢E2¢)
’ (X¢BC¢)
T (X¢E3¢)
 -
Z (X¢E9¢)
½ (X¢B8¢)
```

```
i cms............. SQLVMDB0 ....................
VM/ESA REL. 2.1 04/15/95 13:52

The example of implementation Field Procedure follows:


ARI0717I Start SQLSTART EXEC: 07/13/96 12:39:40 CET.
ARI0663I FILEDEFS in effect are:
ARISQLLD DISK     ARISQLLD LOADLIB  Q1
BDISK    DISK     200
LOGDSK1  DISK     201
DDSK1    DISK     202
  .
  .
ARISSCR  DISK     ARISSCR  MACRO    *
ARISSTR  DISK     ARISSTR  MACRO    *
ARISCCS  DISK     ARISCCS  MACRO    *
NETID    DISK     SNA      NETID    *
ARISPOOL DISK     SQLVMDB0 ARISPOOL *
ARI0020I Virtual machine addressing mode = 31
  .
  .
ARI0015I PROTOCOL parameter value is AUTO.
ARI0015I ACCOUNT parameter value is N.
ARI0015I DUMPTYPE parameter value is F.
ARI0015I LOGMODE parameter value is Y.
ARI0015I STARTUP parameter value is W.
ARI0015I SYSMODE parameter value is M.
ARI0015I EXTEND parameter value is N.
ARI0015I CHARNAME parameter value is 870.
ARI0015I DBNAME parameter value is SQLVMDB0.
ARI0015I RESID parameter value is SQLVMDB0.
ARI0015I PARMID parameter value is SLO870.
  .
  .
ARI0282I LUW UNDO is completed.
ARI0281I LUW REDO is completed.
ARI0159D The specified CHARNAME is different than the
         current system CHARNAME. This will force the
         system CCSID values to be changed. Do you wish
         to change the system CHARNAME?

         Enter 0(No) to continue using the current CHARNAME
                     of INTERNATIONAL, or
               1(Yes) to change the system CHARNAME
                     to 870, or
               111(Quit) to cancel your request and end
                     SQLSTART processing.
1

ARI0143I The application server has been initialized
         with the following values:
         CHARNAME = 870, DBCS = NO, CHARSUB = SBCS,
           .
           .
         CCSIDSBCS = 870, CCSIDMIXED = 0, CCSIDGRAPHIC = 0.
```

Figure 59 (Part 1 of 2). SQL/DS Startup at ZPIZ

```
ARI0134I Application server SQLVMDB0 has been
         identified as a global resource.
ARI0060I SQL/DS initialization complete.
ARI0045I Ready for operator communications.
```

Figure 59 (Part 2 of 2). SQL/DS Startup at ZPIZ

```
create table slo870seq (name char(1) fieldproc slosort) in sample

ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 0
insert into slo870seq values(¢b¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(¢A¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(¢d¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(¢C¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(¢D¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(¢t¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(¢S¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(¢z¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(x¢67¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(x¢BC¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
insert into slo870seq values(x¢B8¢)
ARI0500I SQL processing was successful.
ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 1
```

Figure 60. Inserted Records at ZPIZ

```
select name from slo870seq

NAME
----
B
A
D
C
D
T
S
Z
Å
’
½
* End of Result ** 11 Rows Displayed ** Cost Estimate is 1 *
```

Figure 61. Selected Records at ZPIZ

```
select name from slo870seq order by name
--------------------
NAME
----
A
B
C
Å
D
D
S
’
T
Z
½
* End of Result ** 11 Rows Displayed ** Cost Estimate is 1 *
```

Figure 62. Selected Records Order by Field Procedure SLOSORT at ZPIZ

```
select hex(name) from slo870seq
--------------------
HEX(NAME)
-----------
C2
C1
C4
C3
C4
E3
E2
E9
67
BC
B8
* End of Result ** 11 Rows Displayed ** Cost Estimate is 1 *
```

Figure 63. Selected Records Hex Presentation at ZPIZ

```
--------------------
SET CASE
ARI7955I The system ended your query result to process your command.
ARI7725E The SET CASE command processing stopped.
         The command is not complete. You must specify
         UPPER or STRING
set case upper
ARI7727I The old CASE value was UPPER.
         The new CASE value is UPPER.
```

Figure 64. Checking SET CASE at ZPIZ

3.3.4 SQLINIT QUERY Output

SQLINIT QUERY is used to display the parameters set by the last invocation of SQLINIT. Here is an example:

```
SQLINIT QUERY

ARI0717I Start SQLINIT EXEC: 07/11/96 17:21:58 CET.
SELECTED TABLE IS: SQL/DS
 DBNAME=SQLVMDB0
 DBCS=NO
 SYNCHRONOUS=NO
 DATEFORMAT=ISO
 TIMEFORMAT=ISO
 TRACERA=00
 LDATELEN=0
 LTIMELEN=0
 RELEASE=3.5.0
 WORKUNIT=YES
 QRYBLKSIZE=8
 PROTOCOL=DRDA
 CHARNAME=INTERNATIONAL
 CCSIDSBCS=500
 CCSIDMIXED=0
 CCSIDGRAPHIC=0
 TRACEDRRM=0000
 TRACECONV=0
 SSSNAME=ARIQSTAT
ARI0796I End SQLINIT EXEC: 07/11/96 17:21:58 CET
```

Figure 65. SQLINIT QUERY at ITSO

```
SQLINIT QUERY

ARI0717I Start SQLINIT EXEC: 06/25/96 14:56:41 UTC.
SELECTED TABLE IS: SQL/DS
 DBNAME=SQLDBA
 DBCS=NO
 SYNCHRONOUS=NO
 DATEFORMAT=ISO
 TIMEFORMAT=ISO
 TRACERA=00
 LDATELEN=0
 LTIMELEN=0
 RELEASE=3.5.0
 WORKUNIT=YES
 QRYBLKSIZE=8
 PROTOCOL=DRDA
 CHARNAME=SLOVENIA
 CCSIDSBCS=870
 CCSIDMIXED=0
 CCSIDGRAPHIC=0
 TRACEDRRM=0000
 TRACECONV=0
 SSSNAME=
ARI0796I End SQLINIT EXEC: 06/25/96 14:56:41 UTC.
```

Figure 66. SQLINIT QUERY at ZPIZ

3.3.5 SQLGLOB Usage

The SQLGLOB EXEC allows you to set database-wide defaults for all SQLINIT parameters. It creates a file on the production disk containing these defaults. The syntax of this EXEC is similar to the SQLINIT EXEC.

The SQLGLOB EXEC was enhanced to accept new parameters which are the same as the SQLINIT new parameters described above, and they also have the same meaning.

3.4 AVS/VM Setup

3.4.1 Setting Up AGWPROF Profile

This profile, if it exists, is executed automatically by the AVS code when the command AGW START is executed. Examples of AGWPROFs that are used are given below:

```
¢AGW ACTIVATE GATEWAY  IPFA2GL4 GLOBAL¢
¢AGW CNOS      IPFA2GL4 IPFA21CD IBMRDBM 20 10 10¢
```

Figure 67. AGWPROF GCS at ITSO

```
¢AGW ACTIVATE GATEWAY Z00AVSL1 GLOBAL¢
¢AGW ACTIVATE GATEWAY Z00AVSL2 GLOBAL¢
```

Figure 68. AGWPROF GCS at ZPIZ

If the profile does not exist, the AVS commands contained in AGWPROF have to be manually entered. Those commands define the type of gateways to be activated.

In some situations, an AGW CNOS (Change Number Of Serious) command might be issued. It is recommended that you code the CNOS command into the GCS profile. It serves as useful documentation, and it provides the VTAM/NCP people useful information as well. For more details about AVS/VM please use *VM/ESA Connectivity Planning, Administration, and Operation*, SC24-5448.

3.4.2 Defining AVS to VTAM

Since AVS runs as a VTAM application, it must be defined to VTAM. It is then considered as a VTAM resource, and VTAM knows that it exists.

When AVS is defined to VTAM it provides gateways from a VM viewpoint, not to be confused with gateways in an SNA network.

Although there is only one AVS machine, several gateways can be defined to VTAM. The ITSO definitions include one gateway.

```
IPF2AVS  VBUILD TYPE=APPL
IPFA2GL4 APPL   APPC=YES,
                AUTHEXIT=YES,
                AUTOSES=20,
                DSESLIM=200,
                DMINWNL=100,
                DMINWNR=100,
                EAS=3999,
                MAXPVT=200K,
                SECACPT=ALREADYV,
                VERIFY=NONE,
                VPACING=2,
                MODETAB=DRDAMOD,
                DLOGMOD=IBMRDBM,
                SYNCLVL=CONFIRM,
                OPERCNOS=ALLOW,
                PARSESS=YES
```

Figure 69. VM/VTAM APPL for AVS/VM at ITSO

The definitions, shown for ZPIZ, include two gateways.

```
AVSVM    VBUILD TYPE=APPL
Z00AVSL1 APPL   APPC=YES,
                AUTHEXIT=YES,AUTOSES=20,
                DSESLIM=20,DMINWNL=10,DMINWNR=10,
                EAS=3999,MAXPVT=200K,
                SECACPT=ALREADYV,VERIFY=NONE,
                VPACING=2,MODETAB=AMODETAB,DLOGMODE=IBMRDB,
                SYNCLVL=CONFIRM,OPERCNOS=ALLOW,PARSESS=YES
Z00AVSL2 APPL   APPC=YES,
                AUTHEXIT=YES,AUTOSES=20,
                DSESLIM=20,DMINWNL=10,DMINWNR=10,
                EAS=3999,MAXPVT=200K,
                SECACPT=ALREADYV,VERIFY=NONE,
                VPACING=2,MODETAB=AMODETAB,DLOGMODE=IBMRDB,
                SYNCLVL=CONFIRM,OPERCNOS=ALLOW,PARSESS=YES
```

Figure 70. AVS/VM AGW Q ALL at ZPIZ

3.5 SQL/DS for VM DRDA Flow

Once the standard program to program communication is in place and the databases are running, the DRDA flows are ready to begin. DRDA flows are the “voices” of the server and requester programs talking to each other.

To summarize what has been covered so far, the following must be true before DRDA flows can begin:

- The systems to be connected are running.
- The systems are connected by standard SNA communication links that support LU 6.2.

- End users are properly defined to VM and have active communication directories that point to database resources in the network and indicate the gateways to be used in accessing them.
- The gateways are active (sessions may either already exist or be automatically created as needed).
- The database systems are running and (for SQL/DS) have identified themselves as resources using the *IDENT service.

Conversations can now begin. There are four types of DRDA flows. Each conversation will include the first flow and should include the fourth flow. Each conversation may also include one or more of the second and/or the third flows.

1 Initialization Flow

Here, the two sides exchange information about themselves. If they agree to speak to each other, then an SNA conversation is started on an available session.

2 Bind Flow

In this flow, the AR sends SQL statements to the AS to be stored as part of a program for later execution. An example would be a DB2/2 user creating an SQL/DS package.

3 SQL Statement Execution Flow

In this flow, the AR requests execution of SQL commands against the location where the data resides. These commands may have arrived at the target site earlier as part of a BIND flow, or they may arrive on the SQL Statement Execution Flow (as in an ISQL SELECT).

4 Termination Flow

This flow ends the conversation in an orderly fashion.

3.5.1 SQL Connect Flow from VM AR to Local VSE AS at ITSO

Figure 71 shows the components involved and the required customization. In short, the VM CMS user SQLUSER logged on, and issued an SQLINIT to connect to the VSE AS database SQLVSDB1. All required data is retrieved from its UCOMDIR and sent to the target network (VTAM) that passed the request to the target LU IPFA21CD (VSE CICS) which the VSE AS is running.

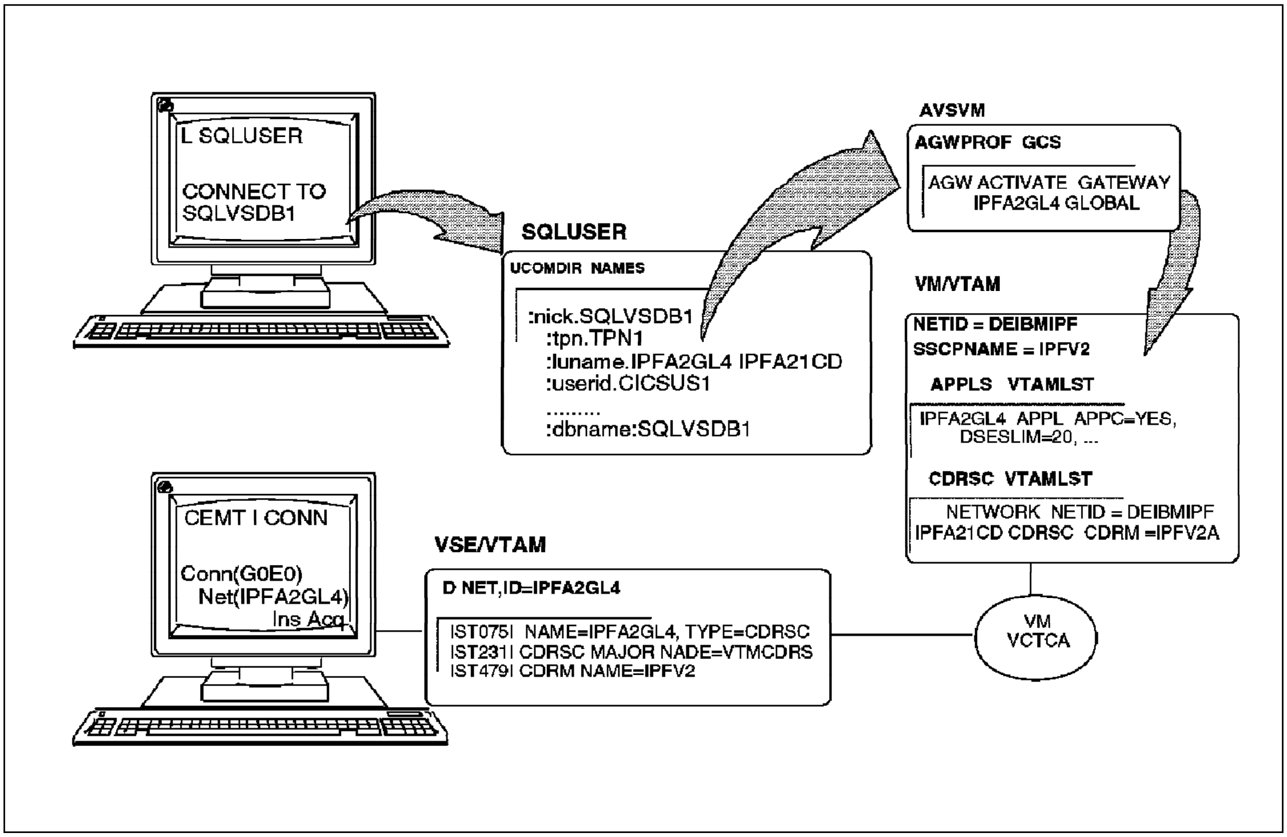


Figure 71. SQL/DS VM AR Connect Flow to Local SQL/DS VSE AS at ITSO

Please check the status of AVSVM, CICS, VSE/VTAM:

1. agw q all

```
IPFA2GL4                        ACTIV
```

2. cemt i conn(g0e0)

```
Conn(g0e0)     Net(IPFA2GL4)     Ins Acq
```

3. d net,id=IPFA2GL4,e

```
NAME = DEIBMIPF.IPFA2GL4 , TYPE = CDRSC
CDRSC MAJOR NODE              = VTMCDRS
CPNAME1                       = DEIBMIPF.IPFV2
SESSIONS:
IPFA21CD                        ACTIV
```

```
d net,id=IPFA21CD,e

    NAME = DEIBMIPF.IPFA21CD , TYPE = APPLC
    APPL MAJOR NODE              = APPLDRD
    SESSIONS:
    IPFA2GL4                       ACTIV
```

3.5.2 SQL Connect Flow from VM AR to Local VSE AS at ZPIZ

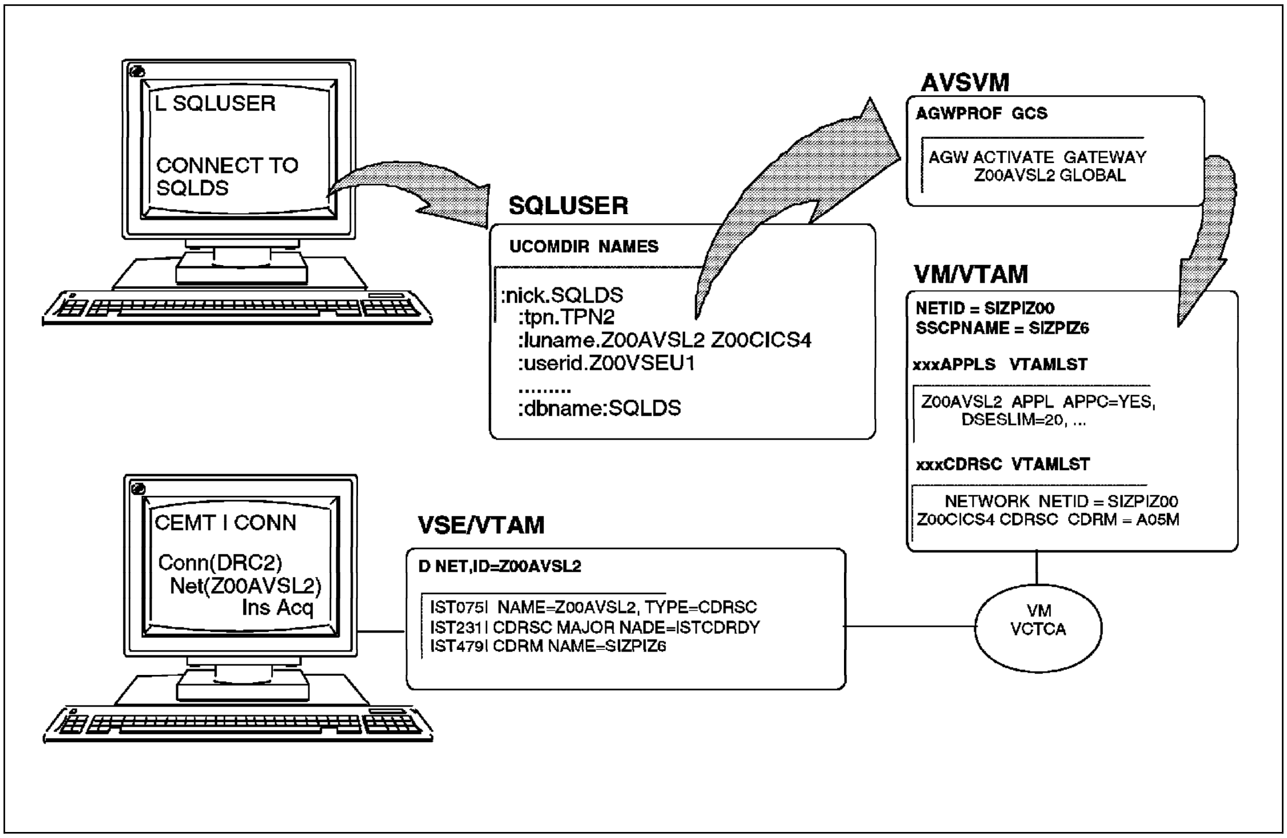


Figure 72. SQL/DS VM AR Connect Flow to Local SQL/DS VSE AS at ZPIZ

The sequence of events for ITSO and ZPIZ is as follows:

1 An SQL/DS CONNECT command (CONNECT to SQLVSDB1 at ITSO or CONNECT TO SQLDS at ZPIZ) is issued, specifying a remote target.

2 The SQL/DS CONNECT function locates the COMDIR entry (UCOMDIR NAMES) for that target and uses it to resolve the LU name (IPFA21CD at ITSO or Z00CICS4 at ZPIZ) and TPN name (TPN1 at ITSO or TPN2 at ZPIZ) that AVS will need to locate the resource.

   If a COMDIR entry exists but is invalid, then the connection request will fail. This example will continue presuming that a valid COMDIR entry does exist.

3 The AVS machine converts the VM/APPC commands to SNA APPC and uses the LU provided in the COMDIR to establish a conversation to the remote gateway.

4 VM/VTAM passes the request to a CDRM located at another node (IPFV2A at ITSO or A05M at ZPIZ) which then connects to a CICS applid (IPFA21CD at ITSO or Z00CICS4 at ZPIZ).

5 If the connection with CICS is not acquired, the connection will fail.

6 If the connection with CICS is acquired then the TPN transaction (TPN1 at ITSO or TPN2 at ZPIZ) specified on the request, will be started.

7 The transaction TPN (TPN1 at ITSO or TPN2 at ZPIZ) will then check the SNT table for the proper user ID (CICSUS1 at ITSO or Z00VSEU1 at ZPIZ) and password, that was sent on the request.

8 If the user ID/password match the TPN program, ARIAXED will check the DBNAME directory. If the matching entry for TPN is found in the DBNAME directory, the CONNECT request is passed to the proper database manager (SQLVSDB1 at ITSO or SQLDS at ZPIZ).

3.5.3 SQL Connect Flow from ZPIZ VM AR to Remote VSE AS at ITSO

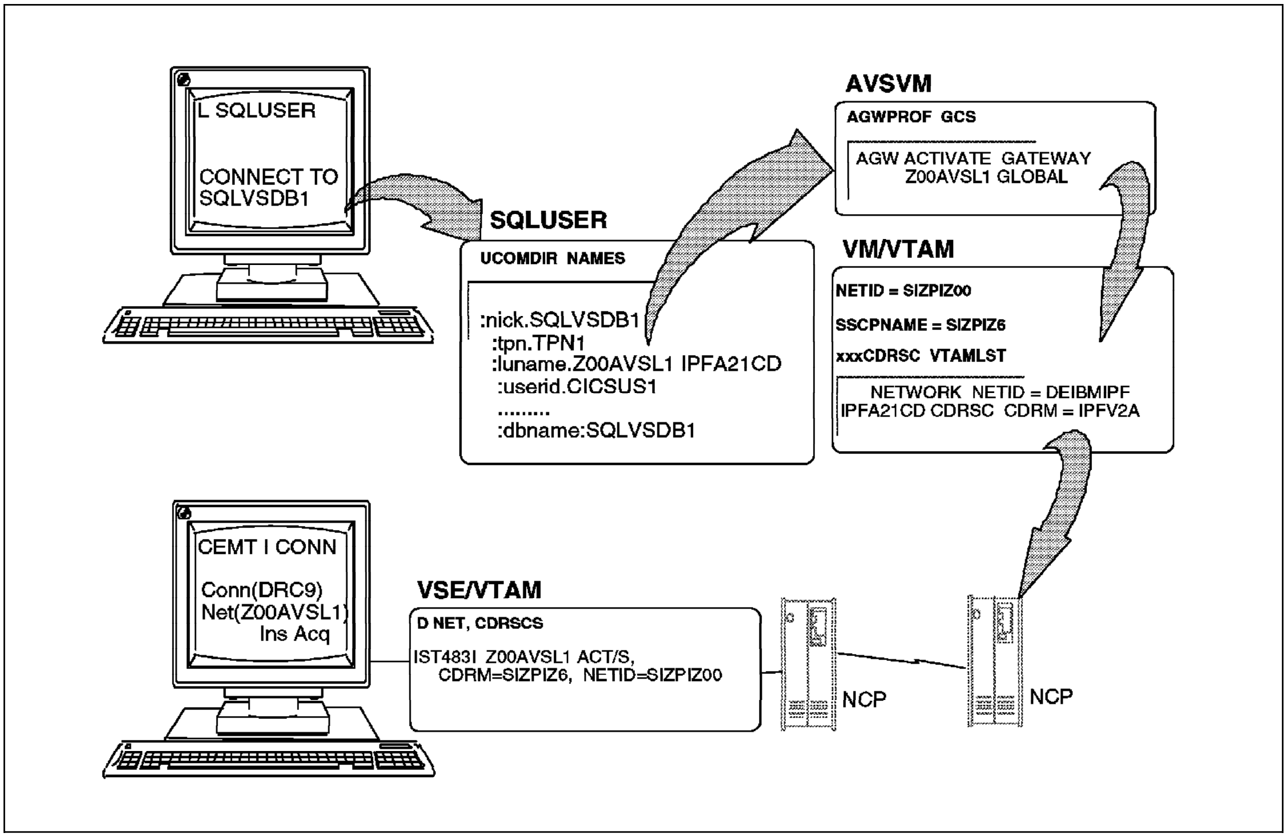


Figure 73. ZPIZ VM AR Connect Flow to Remote SQL/DS VSE AS at ITSO

The sequence of events is similar to the previous case, where a VM AR connects to a local VSE AS.

Please check the status of AVSVM, CICS, VSE/VTAM, VM/VTAM:

1. agw q all

```
Z00AVSL1                        ACTIV
```

2. cemt i conn(DRC9) ... ITSO side check for ZPIZ VM/ESA SQL/DS AR

```
Conn(DRC9)     Net(Z00AVSL1)     Ins Acq
```

3. d net,id=Z00AVSL1,e

```
NAME = SIZPIZ00.Z00AVSL1 , TYPE = CDRSC
CDRSC MAJOR NODE               = VTMCDRS
CDRM  NAME                     = SIZPIZ6
CPNAME                         = SIZPIZ00.SIZPIZ6
SESSIONS:
IPFA21CD                         ACTIV
```

d net,id=IPFA21CD,e

```
NAME = DEIBMIPF.IPFA21CD , TYPE = APPLC
APPL MAJOR NODE                = VTMCDRS
SESSIONS:
Z00AVSL1                         ACTIV
```

4. d net,id=Z00AVSL1,e

```
NAME = Z00AVSL1            TYPE = APPL
APPL MAJOR NODE                = A06APPLS
SESSIONS:
IPFA21CD                         ACTIV
```

3.5.4 SQL Connect Flow from ZPIZ VM AR to Remote VM AS at ITSO

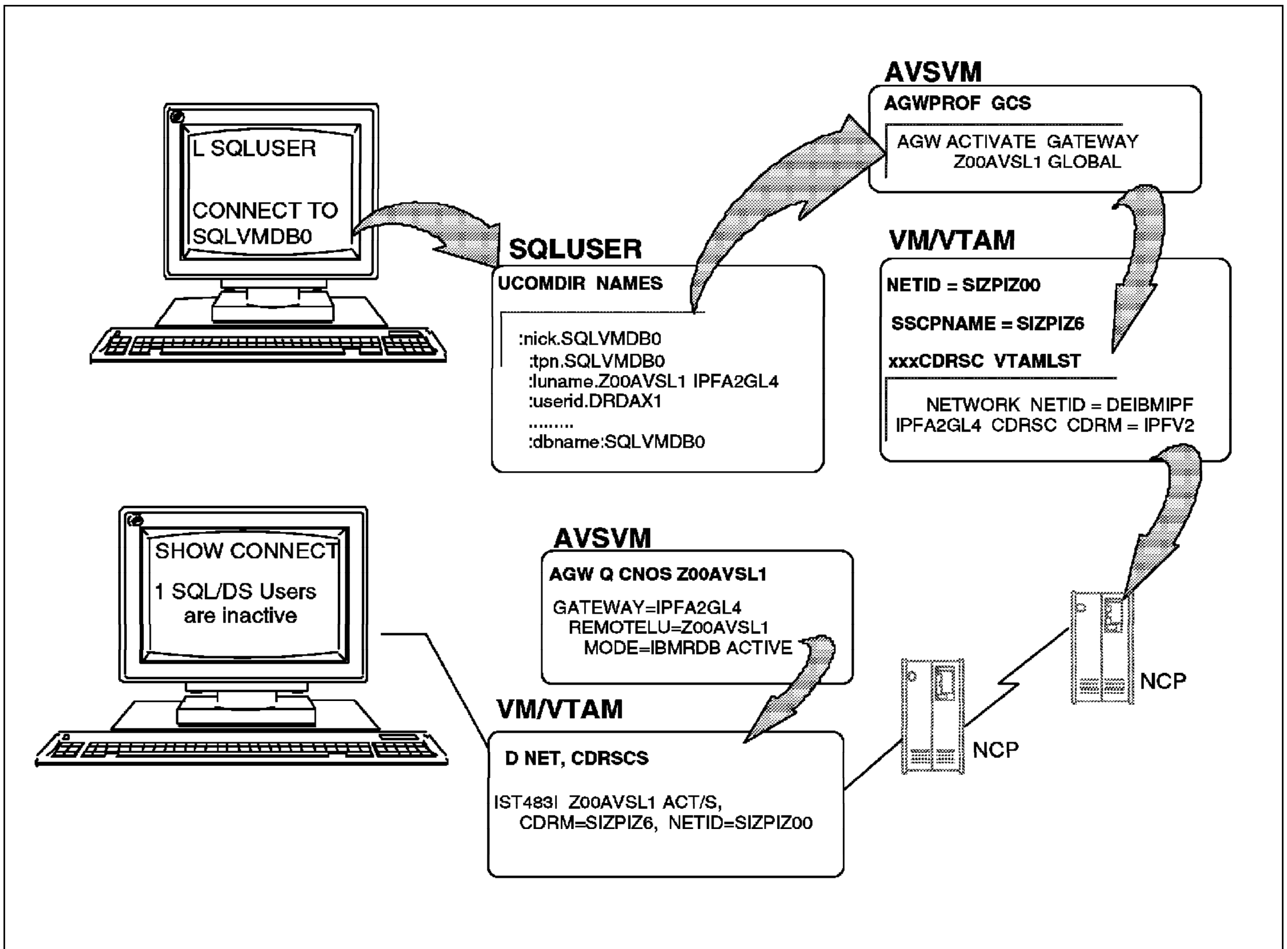


Figure 74. ZPIZ VM AR Connect Flow to Remote SQL/DS VM AS at ITSO

The sequence of events is as follows:

1 An SQL/DS CONNECT command is issued specifying a remote target.

2 The SQL/DS CONNECT function locates the COMDIR entry for that target and uses it to resolve the LU name and TPN name that AVS will need to locate the resource.

  If a COMDIR entry exists but is invalid, then the connection request will fail. This example will continue presuming that a valid COMDIR entry does exist.

3 The AVS machine converts the VM/APPC commands to SNA APPC and uses the LU provided in the COMDIR to establish a conversation to the remote gateway.

4 The remote ITSO AVS checks the VM resource table on its own system to see if the requested resource resides there. If it does, then the connection can be established.

5 Presuming the resource (an SQL/DS database) is found, the VM directory for its system checks for the existence of the user ID and (if SECURITY=PGM) the password that was sent on the request.

6 Presuming a match is found, that user ID is passed to the database manager. This will become the CONNECT user ID for the requesting user on the target SQL/DS system.

In the interests of keeping the examples simple, the security considerations involved have been ignored.

3.5.5 Various Console Captures

Various ITSO and ZPIZ console captures are shown on the next pages.

Display of VM/VTAM CDRSCS at ZPIZ.

- **Z00CICS4** is the CICS applid at ZPIZ.
- **FAEZSVM** is the cross domain resource at IGN.
- **IPFA1VSC** is the VM/VTAM gateway at ITSO.
- **IPFA2GL4** is the AVS/VM gateway at ITSO.
- **IPFA21CD** is the CICS resource at ITSO.

```
send vtam vtam d cdrscs

VTAM   : IST097I DISPLAY ACCEPTED
VTAM   : IST350I DISPLAY TYPE = CDRSCS
VTAM   : IST089I ISTPDILU TYPE = CDRSC SEGMENT    , ACTIV
VTAM   : IST172I NO CDRSCS EXIST
VTAM   : IST089I ISTCDRDY TYPE = CDRSC SEGMENT    , ACTIV
VTAM   : IST483I Z00T010A ACT/S----Y, CDRM = SIZPIZ  , NETID = SIZPIZ00
VTAM   : IST089I A06CDRSC TYPE = CDRSC SEGMENT    , ACTIV
VTAM   : IST483I Z00CICS4 ACT/S      , CDRM = A05M    , NETID = SIZPIZ00
VTAM   : IST483I FAEZSVM  ACTIV      , CDRM = FAE     , NETID = ATIBMFA
VTAM   : IST483I IPFA1VSC ACTIV      , CDRM = IPF     , NETID = DEIBMIPF
VTAM   : IST483I IPFA2GL4 ACT/S      , CDRM = IPFV2   , NETID = DEIBMIPF
VTAM   : IST483I IPFA21CD ACT/S      , CDRM = IPFV2A  , NETID = DEIBMIPF
VTAM   : IST314I END
```

Figure 75. VM/VTAM D CDRSCS at ZPIZ

Below is a more detailed display of the ITSO gateway IPFA2GL4 at ZPIZ indicating 32 active sessions between both AVS/VM gateways.

```
send vtam vtam d id=ipfa2gl4,e

VTAM   : IST097I DISPLAY ACCEPTED
VTAM   : IST075I NAME = IPFA2GL4, TYPE = CDRSC
VTAM   : IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
VTAM   : IST599I REAL NAME = DEIBMIPF.IPFA2GL4
VTAM   : IST977I MDLTAB=***NA*** ASLTAB=***NA***
VTAM   : IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
VTAM   : IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
VTAM   : IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
VTAM   : IST231I CDRSC MAJOR NODE = A06CDRSC
VTAM   : IST479I CDRM NAME = IPFV2, VERIFY OWNER = NO
VTAM   : IST082I DEVTYPE = CDRSC
VTAM   : IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
VTAM   : IST171I ACTIVE SESSIONS = 0000000032, SESSION REQUESTS = 0000000000
VTAM   : IST206I SESSIONS:
VTAM   : IST634I NAME     STATUS     SEND RECV VR TP NETID
VTAM   : IST635I Z00AVSL1 ACTIV-S    0001 0000  0  0 SIZPIZ00
VTAM   : IST635I Z00AVSL1 ACTIV-S    0001 0000  0  0 SIZPIZ00
VTAM   : IST635I Z00AVSL1 ACTIV-S    0001 0000  0  0 SIZPIZ00
VTAM   : IST635I Z00AVSL1 ACTIV-S    0001 0000  0  0 SIZPIZ00
 ...
VTAM   : IST635I Z00AVSL1 ACTIV-P    0000 0001  0  0 SIZPIZ00
VTAM   : IST635I Z00AVSL1 ACTIV-P    0000 0001  0  0 SIZPIZ00
 ...
VTAM   : IST314I END
```

Figure 76. VM/VTAM D ID=IPFA2GL4,E at ZPIZ

The local side of the AVS/VM gateway, where the sessions are established, is shown below.

From a ZPIZ AVS/VM viewpoint there are 46 sessions active, including active local and remote (VM and VSE) sessions.

```
send vtam vtam d id=z00avsl1,e

VTAM    : IST097I DISPLAY ACCEPTED
VTAM    : IST075I NAME = Z00AVSL1, TYPE = APPL
VTAM    : IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
VTAM    : IST977I MDLTAB=***NA*** ASLTAB=***NA***
VTAM    : IST861I MODETAB=AMODETAB USSTAB=***NA*** LOGTAB=***NA***
VTAM    : IST934I DLOGMOD=IBMRDB USS LANGTAB=***NA***
VTAM    : IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
VTAM    : IST231I APPL MAJOR NODE = A06APPLS
VTAM    : IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
VTAM    : IST869I USERID = AVSVM
VTAM    : IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
VTAM    : IST171I ACTIVE SESSIONS = 0000000046, SESSION REQUESTS = 0000000000
VTAM    : IST206I SESSIONS:
VTAM    : IST634I NAME     STATUS     SEND RECV VR TP NETID
VTAM    : IST635I Z00T010A ACTIV-S    0001 0000  0  0 SIZPIZ00
VTAM    : IST635I Z00T010A ACTIV-S    0001 0000  0  0 SIZPIZ00
VTAM    : IST635I Z00T010A ACTIV-S    0001 0000  0  0 SIZPIZ00
 ...
VTAM    : IST635I IPFA2GL4 ACTIV-S    0001 0000  0  0 DEIBMIPF
VTAM    : IST635I IPFA2GL4 ACTIV-S    0001 0000  0  0 DEIBMIPF
 ...
VTAM    : IST635I IPFA21CD ACTIV-S    0001 0000  0  0 DEIBMIPF
VTAM    : IST635I IPFA2GL4 ACTIV-S    0001 0000  0  0 DEIBMIPF
VTAM    : IST635I IPFA2GL4 ACTIV-S    0001 0000  0  0 DEIBMIPF
 ...
 ...
VTAM    : IST314I END
```

Figure 77. VM/VTAM D ID=Z00AVSL1,E at ZPIZ

AVS/VM has its own type of commands. This screen shows AVS/VM at ZPIZ, the defined gateways, active sessions, user IDs and conversations established.

```
send avsvm agw q all

AVSVM   : GATEWAYS:
AVSVM   : GATEWAY = Z00AVSL1 GLOBAL ACTIVE          CONV COUNT = 000000001
AVSVM   : GATEWAY = Z00AVSL2 GLOBAL ACTIVE          CONV COUNT = 000000000
AVSVM   : CNOS:
AVSVM   : GATEWAY = Z00AVSL1 REMOTELU = IPFA2GL4 MODE = IBMRDB
AVSVM   :                          ACTIVE          CONV COUNT = 000000000
AVSVM   : CURRENT VALUES:       LIMIT = 00020 WINNER = 00010 LOSER = 00010
AVSVM   : CNOS COMMAND VALUES:  LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : GATEWAY = Z00AVSL1 REMOTELU = IPFA2GL4 MODE = IBMRDBM
AVSVM   :                          ACTIVE          CONV COUNT = 000000000
AVSVM   : CURRENT VALUES:       LIMIT = 00010 WINNER = 00005 LOSER = 00005
AVSVM   : CNOS COMMAND VALUES:  LIMIT = 00010 WINNER = 00005 LOSER = 00005
AVSVM   : GATEWAY = Z00AVSL1 REMOTELU = IPFA21CD MODE = IBMRDB
AVSVM   :                          ACTIVE          CONV COUNT = 000000000
AVSVM   : CURRENT VALUES:       LIMIT = 00020 WINNER = 00010 LOSER = 00010
AVSVM   : CNOS COMMAND VALUES:  LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : GATEWAY = Z00AVSL1 REMOTELU = IPFBOEDJ MODE = IBMRDB
AVSVM   :                          ACTIVE          CONV COUNT = 000000000
AVSVM   : CURRENT VALUES:       LIMIT = 00020 WINNER = 00010 LOSER = 00010
AVSVM   : CNOS COMMAND VALUES:  LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : GATEWAY = Z00AVSL1 REMOTELU = IPFCL0E0 MODE = IBMRDB
AVSVM   :                          QUIESCED        CONV COUNT = 000000000
AVSVM   : CURRENT VALUES:       LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : CNOS COMMAND VALUES:  LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : GATEWAY = Z00AVSL1 REMOTELU = IPFSL0E0 MODE = IBMRDB
AVSVM   :                          QUIESCED        CONV COUNT = 000000000
AVSVM   : CURRENT VALUES:       LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : CNOS COMMAND VALUES:  LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : GATEWAY = Z00AVSL1 REMOTELU = Z00T010A MODE = IBMRDB
AVSVM   :                          ACTIVE          CONV COUNT = 000000001
AVSVM   : CURRENT VALUES:       LIMIT = 00020 WINNER = 00010 LOSER = 00010
AVSVM   : CNOS COMMAND VALUES:  LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : GATEWAY = Z00AVSL2 REMOTELU = Z00CICS4 MODE = IBMRDB
AVSVM   :                          ACTIVE          CONV COUNT = 000000000
AVSVM   : CURRENT VALUES:       LIMIT = 00020 WINNER = 00010 LOSER = 00010
AVSVM   : CNOS COMMAND VALUES:  LIMIT = 00000 WINNER = 00000 LOSER = 00000
AVSVM   : USERIDS:
AVSVM   : REMOTELU=Z00T010A REMOTE USERID=Z00VMU1 LOCAL USERID=SQLUSER
AVSVM   : CONVERSATIONS:
AVSVM   : GATEWAY = Z00AVSL1, REMOTELU = Z00T010A, RESOURCE = SQLDBA
AVSVM   : USERID   = ,ACCESS USERID = SQLUSER , CONVERSATION = 01000084
AVSVM   :                VTAM STATE = 05, VM STATE = 04
AVSVM   : AGWCMD149I End
```

Figure 78. VM/VTAM APPL for AVS/VM at ZPIZ

CNOS - number of sessions established from gateway Z00AVSL1 on the ZPIZ side:

- remote LU IPFA2GL4 (AVSVM at ITSO)
- remote LU IPFA21CD (CICS APPLID at ITSO)
- remote LU IPFBOEDJ (AIX at ITSO)
- remote LU IPFCL0E0 (PS/2 via 3745 CU at ITSO)
- remote LU IPFSL0E0 (PS/2 via 3174 CU at ITSO)
- remote LU Z00T010A (PS/2 at ZPIZ)

CNOS - number of sessions established from gateway Z00AVSL2 on the ZPIZ side:

- remote LU Z00CICS4 (CICS APPLID at ZPIZ)

Conversation between gateway Z00AVSL1 and PS/2 requester (Z00T010A) is active.

Two agents are connected to the VM SQL/DS named SQLVMDB0. The first one is SQLUSER at BOEVMIS2, a local user. The second one is user DRDAX1 from DDCS/2 which is in conversation status. Please note that this display shows SQL/DS and DRDA protocol used.

```
Status of Connected SQL/DS Users             1996-07-12  07:15:44
   Checkpoint agent is not active.
   User Agent:   1   User-ID: SQLUSER  SQL-ID: SQLUSER
      is R/O  APPL  7BD9
      Agent is processing and is in communication  wait.
        State started: 1996-07-12  07:15:34
        Conversation started: 1996-07-12  07:15:25
        LUWID: SNANETID.*IDENT.AD272F3D3FC4.0001
        EXTNAM: SQLUSER.1
        Requester: SQLDS/VM  V3.5.0   at BOEVMIS2
        Package: SQLDBA.ARIISQL      Section: 4
   User ID: DRDAX1    SQL ID: DRDAX1
      User is inactive.
        State started: 1996-07-12  07:11:28
        Conversation started: 1996-07-12  07:11:27
        LUWID: DEIBMIPF.IPFCL0E0.AD27324127D1.0001
        EXTNAM: DB2BP.EXE           007A0001
        Requester: DDCS/2    V2.1.1   at
  1  SQL/DS users are active.
  0  SQL/DS users are waiting.
  1  SQL/DS users are inactive.
  9  SQL/DS agents are available.
  486  SQL/DS user connections are available.
ARI0065I SQL/DS operator command processing is complete.
```

Figure 79. SQL Show Connect Example at ITSO

3.5.6 Multi-Language Character Set Conversions

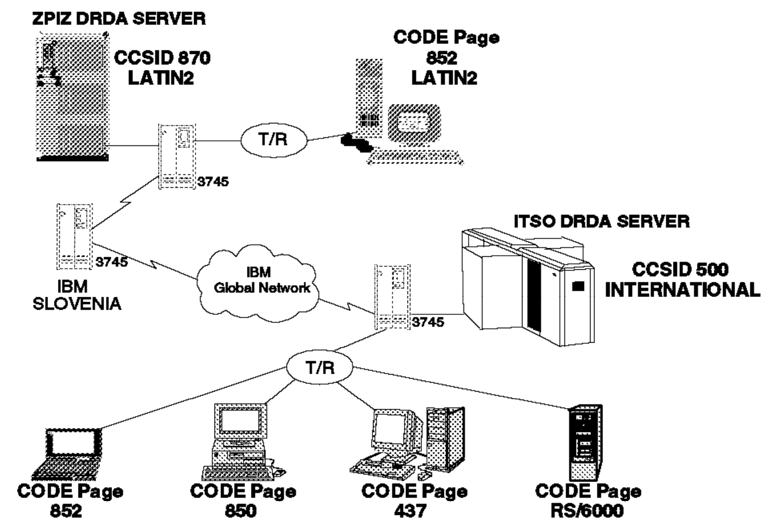


Figure 80. Multi-Language Character Set Conversions

After SQL/DS was successfully installed at both the ITSO and ZPIZ sites, the ZPIZ customer with CCSID (870) tries to access ITSO′s SQL/DS initiated with CCSID (500). The connection failed. The translation table between these two code pages does not exist. IBM provided the appropriate translation by inserting records into the SYSTEM.SYSSTRINGS table. SQL connections on VM/ESA and VSE/ESA SQL/DS of the following CCSID (CP/CS) are now enabled:

- ZPIZ CCSID 870 (L2 SLOVENIA) <-------> ITSO 037(L1 USA)
- ZPIZ CCSID 870 (L2 SLOVENIA) <-------> ITSO 500(L1 INTERNATIONAL)

Before you run CMS EXEC make sure that SQL/DS is initiated with the private protocol SQLDS, so that conversion is not enabled. The appropriate insert to SYSTEM.SYSSTRINGS was done at the ITSO and ZPIZ sites.

```
>>>>>   Insert 4846 fig82 e6 here   <<<<<

SET ERRORMODE CONTINUE;
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW TO SQLDBA;
DELETE FROM SYSTEM.SYSSTRINGS WHERE INCCSID=037 AND OUTCCSID=870;
INSERT INTO SYSTEM.SYSSTRINGS VALUES(037,870,¢SS¢, NULL,NULL,¢ ¢,
¢    ¢;          •      ¢,
¢ âäàáãåçñê.<(+]&éèëïíîìÀß|$*);Ã-/ÂÄÅÁÑøÇÊÈ,%_>?ÏÉÌËØÍÎ«»:#@¢¢=†‰abcde
fghi••æ-—Œ jklmnopqrÆ•¡Ÿ¿•‡stuvwxyz„[œ•£¥ fifl¶¼§½¾•‘¢!’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
DELETE FROM SYSTEM.SYSSTRINGS WHERE INCCSID=870 AND OUTCCSID=037;
INSERT INTO SYSTEM.SYSSTRINGS VALUES(870,037,¢SS¢, NULL,NULL,¢ ¢,
¢    ¢;          •      ¢,
¢ âäàáãåçñ•.<(+!&é¢ëêíîèïß‘$*);[-/ÂÄìÁ ÀÇÃ|,%_>?ÅÉÑË]ÍÎøÊ:#@¢¢=†Èabcde
fghiÏÌØ-«» jklmnopqr••‰Ÿ—•Œstuvwxyzæ Æœ••¡¿‡„£¥§fifl¶¼½¾’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
DELETE FROM SYSTEM.SYSSTRINGS WHERE INCCSID=500 AND OUTCCSID=870;
INSERT INTO SYSTEM.SYSSTRINGS VALUES(500,870,¢SS¢, NULL,NULL,¢ ¢,
¢    ¢;          •      ¢,
¢ âäàáãåçñ¢.<(+|&éêëèíîïìß!$*); -/ÂÄÀÁÃÅÇÑø,%_>?ÊÉÈËÏÍÎÌØ:#@¢¢=†«abcde
fghi»‰•-•— jklmnopqrŒæÆŸ••¡stuvwxyz¿‡œ•„[£¥fifl¶§¼½¾•‘]’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
DELETE FROM SYSTEM.SYSSTRINGS WHERE INCCSID=870 AND OUTCCSID=500;
INSERT INTO SYSTEM.SYSSTRINGS VALUES(870,500,¢SS¢, NULL,NULL,¢ ¢,
¢    ¢;          •      ¢,
¢ âäàáãåçñ¢.<(+|&éêëèíîïìß!$*); -/ÂÄÀÁÃÅÇÑ‘,%_>?]ÉøËÊÍÎÈÏ:#@¢¢=†Ìabcde
fghiØ«»-‰• jklmnopqr•—ŒŸæ•Æstuvwxyz•¡œ•¿‡„[£¥fi§fl¶¼½¾•’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
COMMIT WORK;
>>>>>   End of Insert   <<<<<
```

Figure 81. SYSTEM.SYSSTRINGS Table

When you start a connection from ZPIZ to ITSO:

CONNECT TO SQLVMDB0

All the Slovenian characters were correctly displayed at ZPIZ. Using DML insert...hex, the appropriate Slovenian hex letters were inserted into the ITSO database (refer to Figure 105 on page 97). Software and hardware were also set up correctly (microcode and 3270 sessions).

```
/*****************************************************************/
/*                1)  SQL/DS USERID SQLDBA WITH DBA AUTHORITY     */
/*****************************************************************/
 TRACE OFF
¢FILEDEF  SYSIN    DISK  SLO  MACRO    A1¢
¢FILEDEF  SYSPRINT DISK  SLO  DBSULST  A1¢
¢EXEC     SQLDBSU¢
 EXIT     rc
```

Figure 82. Using SQL/DS DBSU Utility

Chapter 4. VSE/ESA Implementation

4.1 ITSO and ZPIZ VSE/ESA System Setup

In this section we discuss the most important setup tasks required to enable SQL/DS for VSE/ESA to perform DRDA AS functions. Actually VSE/ESA cannot act as a DRDA requester, as SQL/DS in the VM/ESA environment does.

The most difficult and time-consuming area is that of establishing the LU 6.2 sessions over which the DRDA traffic will flow. It requires work on both AR and AS sides, and many parameters must be compatible with their counterparts on the partner location.

To summarize what has to be customized in VSE/ESA specifically for DRDA:

- CICS
  - DFHSIT
  - DFHSNT
  - DFHTST
- CICS RDO definitions for:
  - Programs
  - Transactions
  - Connections
  - Sessions
- SQL/DS
  - DBNAME Directory (TPNs, Servers)
- VTAM (under VM)
  - Add CICS Application
  - CDRMS
  - CDRSCs (remote LUs)
- If VTAM/VSE owns the network (not under VM)
  - Add Network/Netids from where the clients are coming
  - CDRMS
  - LU names
  - Resources to local major nodes/NCP
- Multi-Language Considerations
  - Add CCSIDs to support Latin-1 and Latin-2 code pages
  - Add Field Procedures to change the default collating sequence to accommodate the cultural sort sequence for Slovenia.

All these customization tasks are described in more detail in the next pages and the related system tables are in the appendixes.

4.1.1 CICS/VSE Setup

The SQL/DS application server (AS) communicates with its application requester (AR) via CICS LU 6.2 links. The CICS partition to be used for this DRDA connection must have LU 6.2 links to remote systems with the AR.

We have updated the following CICS tables to enable the LU 6.2 support required for DRDA.

DFHSIT

```
DFHSIT TYPE=CSECT,
       ...
       APPLID=IPFA21CD,     DRDA CICS APPLICATION NAME
       GRPLIST=VSELIST,     DRDA GROUP LIST
       ISC=YES,             DRDA REQUIRES INTERSYSTEM COMMUNICATION
       SUFFIX=2A,           DRDA DFHSIT SUFFIX
       TST=2A,              DRDA TEMP STORAGE TABLE
       ...
END    DFHSITBA
```

Figure 83. DFHSIT DRDA Parameters at ITSO

```
DFHSIT TYPE=CSECT,
       ...
       APPLID=Z00CICS4,     DRDA CICS APPLICATION NAME
       GRPLIST=VSELST4,     DRDA GROUP LIST
       ISC=YES,             DRDA REQUIRES INTERSYSTEM COMMUNICATION
       SUFFIX=4A,           DRDA DFHSIT SUFFIX
       TST=4A,              DRDA TEMP STORAGE TABLE
       ...
END    DFHSITBA
```

Figure 84. DFHSIT DRDA Parameters at ZPIZ

- APPLID

  The CICS APPLID (IPFA21CD at ITSO or Z00CICS4 at ZPIZ) will be used to define the CICS application to VSE/VTAM. It will also be needed when establishing LU-LU sessions from:

  - DDCS/2 (the “Partner_LU”)
  - AVS/VM (the “LUname”)
  - DDCS for AIX (the “Partner_LU”)

DFHSNT

One DFHSNT entry for each remote user, or a group of users, allowed to connect to the DRDA server.

```
DFHSNT TYPE=ENTRY,
       USERID=DRDAX1,
       PASSWRD=SQL4VSE,
       RSLKEY=(1)
```

Figure 85. DFHSNT for DRDA Access at ITSO

```
DFHSNT TYPE=ENTRY,
      EXTSEC=NO,
      OPCLASS=(2),
      OPIDENT=Z00,
      TIMEOUT=60,
      USERID=Z00VSEU1,
      PASSWRD=RGNCSQ,
      OPPRTY=128,
      SCTYKEY=(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,
      19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,
      37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,
      55,56,57,58,59,60,61,62,63,64),
      RSLKEY=(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,
      19,20,21,22,23,24)
```

Figure 86. DFHSNT for DRDA Access at ZPIZ

DFHTST

An entry in DFHTST must be provided to support the ARICAXED error logging.

```
DFHTST TYPE=RECOVERY,DATAID=ARIAXELG
```

Figure 87. DFHTST Entry for AXE Logging at ITSO and ZPIZ

CICS RDO definitions

We have defined all the programs, transactions, connections, and sessions using the CICS RDO facility.

- An entry must be defined for the AXE program (ARICAXED).
- All transaction codes (TPNs), which will be available for DRDA users to access VSE SQL/DS DRDA servers, must be defined.
- LU 6.2 links (CONNECTIONs and SESSIONs) must be defined for each remote LU.

In order to facilitate the management of all the DRDA entries we have defined them in a unique group called DRDAITSO at ITSO and DRDAZPIZ at ZPIZ.

- DRDA definitions

  The CEDA DEFINE command is used to define the following resources:

  – LU 6.2 CONNECTIONs
  – LU 6.2 SESSIONs
  – TPN transactions
  – AXE program

- DRDA LIST creation

  In order to install the DRDAZPIZ Group at CICS initialization, we have added it to the current LIST:

  CEDA ADD GR(DRDAITSO) LIST(VSELIST)

  CEDA ADD GR(DRDAZPIZ) LIST(VSELST4)

4.1.2 DBNAME Directory

The DBNAME directory is used to map an incoming request for conversation allocation to a predetermined application server identified by the incoming TPN.

In the DBNAME directory all application IDs and database names are provided together with the transaction program names which must be defined in a DFHPCT or CICS RDO transaction definitions. The DBNAME directory used, is shown below:

```
*****************************************************
*TPN     APPLID     *DBNAME              PNM   PRIV *
*        1     1    22              3    44    5    *
*2..5____0......7___12...............9____45____0____*
 TPN1    SYSARI01   *SQLDS               F8
 TPN2    SYSARI01    SQLDS               F8
 TPN3    SYSARI01    SQLDS               F8    Y
 XMP1    SYSARI02    EXAMPLE
```

Figure 88. DBNAME Directory - ZPIZ

```
******************************************************************
*TPN     APPLID     *DBNAME              PID   PRIV             *
*        1     1    22              3    44    5                *
*2..5____0......7___12...............9____45____0________________*
 TPN1    SYSARI01   *SQLVSDB1            Z1    Y
 TPN2    SYSARI02    SQLVSDB2            Z2    Y
 TPN3    SYSARI01    SQLVSDB1            Z1    Y
 AXE0    SYSARI01    SQLVSDB1            Z1    Y
         SYSARI05    SQLVS350            F8
         SYSARI03    SQLVSDB0            F8
         SYSARI03    SQLVSDB0            F4
         SYSARI04    SQLVSDBP            F7
         SYSARI04    SQLVSDBP            F5
         SYSARI01    SQLVSDB1            BG
         SYSARI01    SQLVSDB1            F2
         SQLVM350    SQLVM350
         SQLVMDB0    SQLVMDB0
         SQLVMDB1    SQLVMDB1
         SQLVMDB2    SQLVMDB2
         SQLVMDB3    SQLVMDB3
         SQLVMDB4    SQLVMDB4
         SQLVMDB5    SQLVMDB5
         SQLVMDB6    SQLVMDB6
         SQLVMDB7    SQLVMDB7
         SQLVMDB8    SQLVMDB8
         SQLVMDB9    SQLVMDB9
         SQLVMDBA    SQLVMDBA
         SQLVMDBB    SQLVMDBB
 D6DB    SYSARI00    SQLDS
```

Figure 89. DBNAME Directory at ITSO

The following rules apply:

- An entry in the DBNAME directory for each TPN

  An entry must be defined for each TPN used by remote users. Multiple users may have access to the same TPN.

- Each TPN must be defined in DFHPCT

  An entry for each TPN must be defined in DFHPCT. As the TPN is a CICS transaction name, it must be part of the PCT in order to be activated.

- Multiple TPNs to one APPLID

  Multiple TPNs can refer to the same APPLID. This could be very useful to give specific advantages to one user of a database server. CICS transaction definition allows security (RSL) and class (TClass) to be defined at transaction level. These features associated with a TPN can also be used to provide different uses of the AXE program to different users.

- Exclusive use of a Real Agent

  Exclusive use of a real agent can be associated to a TPN (TPN1 at ITSO or TPN3 at ZPIZ). This privilege marked with a "Y" in column 50 of the TPN entry is only valid for the DRDA environment. It will give the user of this TPN the privilege of having a real agent available for its exclusive use during its entire LU 6.2 conversation. TPNs with no exclusive privilege will have their real agent released at the end of each unit of work.

- DBNAME to APPLID relationship should be constant

  For example in the DBNAME directory that is shown, the database with the DBNAME of SQLVSDB1 at ITSO or SQLDS at ZPIZ always maps to the APPLID SYSARI01.

4.1.3 SQL Connection Flow between ITSO and ZPIZ and Local

VM AR Connect Flow to local SQL/DS VSE AS at ITSO
Please refer to Figure 71 on page 72 in Chapter 3

VM AR Connect Flow to local SQL/DS VSE AS at ZPIZ
Please refer to Figure 72 on page 73 in Chapter 3

VM AR Connect Flow to remote SQL/DS VSE AS at ITSO
Please refer to Figure 73 on page 74 in Chapter 3

VM AR Connect Flow to remote SQL/DS VM AS at ITSO
Please refer to Figure 74 on page 75 in Chapter 3

4.1.4 CICS Screen Captures

The screen captures from the RDO processes are shown on the next pages. Some of them are extracts only to show just the vital or corresponding information.

```
 CEDA E GR(DRDAITSO)
 ENTER COMMANDS
  NAME     TYPE         GROUP                              TIME
  S0E0     CONNECTION   DRDAITSO                           19.11.24
  L0E0     CONNECTION   DRDAITSO                           19.08.50
  MMDR     CONNECTION   DRDAITSO                           19.08.50
  VS01     CONNECTION   DRDAITSO                           19.13.23
  OEDJ     CONNECTION   DRDAITSO                           19.11.24
  Z00S     CONNECTION   DRDAITSO                           12.05.42
  ARICAXED PROGRAM      DRDAITSO                           19.14.44
  IPFCL0E0 SESSIONS     DRDAITSO                           19.20.28
  IPFSL0E0 SESSIONS     DRDAITSO                           19.15.15
  IPFA2GL4 SESSIONS     DRDAITSO                           19.15.15
  Z00AVS01 SESSIONS     DRDAITSO                           19.16.19
  IPFBOEDJ SESSIONS     DRDAITSO                           12.07.03
  Z00T010A SESSIONS     DRDAITSO                           12.07.03
  AXE0     TRANSACTION  DRDAITSO                           19.18.16
  TPN1     TRANSACTION  DRDAITSO                           19.18.38
```

Figure 90. CEDA E G(DRDAITSO) at ITSO

```
CEMT I CONN
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Conn(S0E0) Net(IPFSL0E0)     Ins Acq
  Conn(L0E0) Net(IPFCL0E0)     Ins Acq
  Conn(MMDR) Net(IPFA2GL4)     Ins Acq
  Conn(OEDJ) Net(IPFBOEDJ)     Ins Acq
  Conn(VS01) Net(Z00AVSL1)     Ins Acq
  CONN(Z00S) Net(Z00T010A)     Ins Acq
```

Figure 91. CEMT I CONN at ITSO

```
CEDA E G(DRDAZPIZ)
  ENTER COMMANDS
   NAME     TYPE         GROUP
   ----------------------------
   DRC1     CONNECTION   DRDAZPIZ
   DRC2     CONNECTION   DRDAZPIZ
   DRC3     CONNECTION   DRDAZPIZ
   DRC4     CONNECTION   DRDAZPIZ
   DRC5     CONNECTION   DRDAZPIZ
   DRC6     CONNECTION   DRDAZPIZ
   DRC7     CONNECTION   DRDAZPIZ
   DRC8     CONNECTION   DRDAZPIZ
   ARICAXED PROGRAM      DRDAZPIZ
   ARICDIRD PROGRAM      DRDAZPIZ
   ARICMOD  PROGRAM      DRDAZPIZ
   ARIISQL  PROGRAM      DRDAZPIZ
   ARIITRM  PROGRAM      DRDAZPIZ
   ARIITRX  PROGRAM      DRDAZPIZ
   ARIMS001 PROGRAM      DRDAZPIZ
   ARIRCONT PROGRAM      DRDAZPIZ
   ARI0OLRM PROGRAM      DRDAZPIZ
   DRS1     SESSIONS     DRDAZPIZ
   DRS2     SESSIONS     DRDAZPIZ
   DRS3     SESSIONS     DRDAZPIZ
   DRS4     SESSIONS     DRDAZPIZ
   DRS5     SESSIONS     DRDAZPIZ
   DRS6     SESSIONS     DRDAZPIZ
   DRS7     SESSIONS     DRDAZPIZ
   DRS8     SESSIONS     DRDAZPIZ
   CAXE     TRANSACTION  DRDAZPIZ
   CIRA     TRANSACTION  DRDAZPIZ
   CIRB     TRANSACTION  DRDAZPIZ
   CIRC     TRANSACTION  DRDAZPIZ
   CIRD     TRANSACTION  DRDAZPIZ
   CIRR     TRANSACTION  DRDAZPIZ
   CIRT     TRANSACTION  DRDAZPIZ
   CISQ     TRANSACTION  DRDAZPIZ
   ISQL     TRANSACTION  DRDAZPIZ
   TPN1     TRANSACTION  DRDAZPIZ
   TPN2     TRANSACTION  DRDAZPIZ
   TPN3     TRANSACTION  DRDAZPIZ
```

Figure 92. CEDA E G(DRDAZPIZ) at ZPIZ

```
CEMT I CONN
  STATUS:  RESULTS - OVERTYPE TO MODIFY
   Conn(DRC1) Net(Z00T010A)     Ins Acq
   Conn(DRC2) Net(Z00AVSL2)     Ins Acq
   Conn(DRC3) Net(IPFA2GL4)     Ins Acq
   Conn(DRC4) Net(IPFCL0E0)     Ins Rel
   Conn(DRC5) Net(IPFCL0C0)     Ins Rel
   Conn(DRC6) Net(IPFSL0E0)     Ins Rel
   Conn(DRC7) Net(IPFBOEDJ)     Ins Rel
   Conn(DRC8) Net(IPFP101B)     Ins Rel
```

Figure 93. CEMT I CONN at ZPIZ

CONNECTION	NETNAME	COMMENT
DRC1	Z00T010A	Connection to ZPIZ DDCS/2
DRC2	Z00AVSL2	Connection to ZPIZ AVS/VM
DRC3	IPFA2GL4	Connection to ITSO AVS/VM
DRC4	IPFCL0E0	Connection to ITSO DDCS/2
DRC5	IPFCL0C0	Connection to ITSO DDCS/2
DRC6	IPFSL0E0	Connection to ITSO DDCS/2
DRC7	IPFBOEDJ	Connection to ITSO DDCS for AIX
DRC8	IPFP101B	Connection to ITSO DDCS/2

Table 8. CICS Connectivity

```
CEDA D C(L0E0) G(DRDAITSO)
  OBJECT CHARACTERISTICS
   CEDA  View
    Connection     : L0E0
    Group          : DRDAITSO
   CONNECTION IDENTIFIERS
    Netname        : IPFCL0E0
    INDsys         :
   REMOTE ATTRIBUTES
    REMOTESystem   :
    REMOTEName     :
   CONNECTION PROPERTIES
    ACcessmethod   : Vtam         Vtam | IRc | INdirect
    Protocol       : Appc         Appc | Lu61
    SInglesess     : No           No | Yes
    Datastream     : User         User | 3270 | SCs | STrfield | Lms
    RECordformat   : U            U | Vb
   OPERATIONAL PROPERTIES
    AUtoconnect    : Yes          No | Yes | All
 +  INService      : Yes          Yes | No
```

Figure 94. CEDA D C(L0E0) G(DRDAITSO) at ITSO Local via 3174

```
CEDA D C(Z00S) G(DRDAITSO)
  OBJECT CHARACTERISTICS
   CEDA  View
    Connection     : Z00S
    Group          : DRDAITSO
   CONNECTION IDENTIFIERS
    Netname        : Z00T010A
    INDsys         :
   REMOTE ATTRIBUTES
    REMOTESystem   :
    REMOTEName     :
   CONNECTION PROPERTIES
    ACcessmethod   : Vtam           Vtam | IRc | INdirect
    Protocol       : Appc           Appc | Lu61
    SInglesess     : No             No | Yes
    Datastream     : User           User | 3270 | SCs | STrfield | Lms
    RECordformat   : U              U | Vb
   OPERATIONAL PROPERTIES
    AUtoconnect    : Yes            No | Yes | All
 +  INService      : Yes            Yes | No
```

Figure 95. CEDA D C(Z00S) G(DRDAITSO) at ITSO for ZPIZ via 3745

```
CEDA D S(IPFCL0E0) G(DRDAITSO)
  OBJECT CHARACTERISTICS
   CEDA  View
    Sessions       : IPFCL0E0
    Group          : DRDAITSO
   SESSION IDENTIFIERS
    Connection     : L0E0
    SESSName       :
    NETnameq       :
    MOdename       : IBMRDBM
   SESSION PROPERTIES
    Protocol       : Appc               Appc | Lu61
    MAximum        : 00020 , 00010      0-32767
    RECEIVEPfx     :
    RECEIVECount   : No                 No | 1-999
    SENDPfx        :
    SENDCount      : No                 No | 1-999
    SENDSize       : 04096              1-30720
    RECEIVESize    : 04096              1-30720
```

Figure 96. CEDA D S(IPFCL0E0) G(DRDAITSO) at ITSO Local via 3174

```
CEDA D S(Z00T010A) G(DRDAITSO)
  OBJECT CHARACTERISTICS
   CEDA  View
    Sessions       : Z00T010A
    Group          : DRDAITSO
   SESSION IDENTIFIERS
    Connection     : Z00S
    SESSName       :
    NETnameq       :
    MOdename       : IBMRDB
   SESSION PROPERTIES
    Protocol       : Appc               Appc | Lu61
    MAximum        : 00020 , 00010      0-32767
    RECEIVEPfx     :
    RECEIVECount   : No                 No | 1-999
    SENDPfx        :
    SENDCount      : No                 No | 1-999
    SENDSize       : 04096              1-30720
    RECEIVESize    : 04096              1-30720
```

Figure 97. CEDA D S(Z00T010A) G(DRDAITSO) at ITSO for ZPIZ via 3745

```
CEDA V TR(CAXE) G(DRDAZPIZ)
  OBJECT CHARACTERISTICS
   CEDA  View
    TRansaction    : CAXE
    Group          : DRDAZPIZ
    PROGram        : ARICAXED
    TWasize        : 00000              0-32767
    PROFile        : DFHCICST
    PArtitionset   :
    STatus         : Enabled            Enabled | Disabled
    PRIMedsize     : 00000              0-65520
   REMOTE ATTRIBUTES
    DYnamic        : No                 No | Yes
    REMOTESystem   :
    REMOTEName     :
    TRProf         :
    Localq         :                    No | Yes
   SCHEDULING
    PRIOrity       : 001                0-255
    TCLASS         : NO                 NO | 1-10
   ALIASES
    Alias          :
    TAskreq        :
    Xtranid        : 07F6C4C2
```

Figure 98 (Part 1 of 2). CEDA D TR(CAXE) G(DRDAZPIZ) az ZPIZ

```
  RECOVERY
   DTimout        : No               No | 1-6800
   Indoubt        : Backout          Backout | Commit | Wait
   REStart        : No               No | Yes
   SPurge         : Yes              No | Yes
   TPurge         : Yes              No | Yes
   DUmp           : No               Yes | No
   TRACe          : No               Yes | No
  SECURITY
   Extsec         : No               No | Yes
   TRANsec        : 01               1-64
   RSL            : 00               0-24 | Public
   RSLC           : No               No | Yes | External
```

Figure 98 (Part 2 of 2). CEDA D TR(CAXE) G(DRDAZPIZ) az ZPIZ

ITSO: use default AXE0 transaction

```
CEDA V TR(TPN1) G(DRDAZPIZ)
 OBJECT CHARACTERISTICS
  CEDA  View
   TRansaction    : TPN1
   Group          : DRDAZPIZ
   PROGram        : ARICAXED
   TWasize        : 00000            0-32767
   PROFile        : DFHCICST
   PArtitionset   :
   STatus         : Enabled          Enabled | Disabled
   PRIMedsize     : 00000            0-65520
  REMOTE ATTRIBUTES
   DYnamic        : No               No | Yes
   REMOTESystem   :
   REMOTEName     :
   TRProf         :
   Localq         :                  No | Yes
  SCHEDULING
   PRIOrity       : 001              0-255
   TCLASS         : NO               NO | 1-10
  ALIASES
   Alias          :
   TAskreq        :
   Xtranid        :
  RECOVERY
   DTimout        : No               No | 1-6800
   Indoubt        : Backout          Backout | Commit | Wait
   REStart        : No               No | Yes
   SPurge         : No               No | Yes
   TPurge         : No               No | Yes
   DUmp           : No               Yes | No
   TRACe          : No               Yes | No
  SECURITY
   Extsec         : No               No | Yes
   TRANsec        : 01               1-64
   RSL            : 00               0-24 | Public
   RSLC           : No               No | Yes | External
```

Figure 99. CEDA D TR(TPN1) G(DRDAZPIZ) at ZPIZ

4.1.5 VTAM/VSE Setup

There are no special considerations for VTAM/VSE under VM. Most of the VTAM/VSE system tables are shown in Appendix A, "System Tables and Sample Programs" on page 191.

Note that all customization tasks are documented in *Setup and Usage of SQL/DS in a DRDA Environment*, GG24-3733-01.

```
CATALOG APPLDRD.B                             REPLACE=YES
APPLDRD  VBUILD TYPE=APPL
IPFA21CD APPL  ACBNAME=IPFA21CD,                                 C
               AUTH=(ACQ,PASS,VPACE),                            C
               APPC=NO,                                          C
               SONSCIP=YES,                                      C
               EAS=30,                                           C
               PARSESS=YES,                                      C
               MODETAB=DRDAMOD,                                  C
               DLOGMOD=IBMRDBM,                                  C
               VPACING=0
POWER    APPL  AUTH=(ACQ)
PNET     APPL  AUTH=(PASS,ACQ),VPACING=3,MODETAB=VTMLOGTB,DLOGMOD=PNET
IESWAITT APPL  AUTH=(NOACQ)
```

Figure 100. VSE/VTAM Definition at ITSO

ZPIZ: Use Z00CICS4

4.1.6 Multi-Language Considerations

Apart from the conventional customization tasks that have to be done after SQL/DS is installed, we have done some additional customization steps.

1. Update SYSTEM.SYSSTRING to support CCSID=870 and CHARNAME=870 at ITSO and CHARNAME=SLOVENIA at ZPIZ
2. Assemble field procedure to support Latin-2
3. Catalog SQLPARM containing SQL startup parameters
4. Update and submit SQL startup job
5. Grant schedule (CICSDRDA at ITSO and Z00CICS4 at ZPIZ)
6. Create sample table to include Latin-2 field procedure

4.1.7 Assemble Field Procedure to Support Latin-2

SQL/DS V3.5 provides enhanced sample field procedures to change the default collating sequence to accommodate the Latin-2 cultural sort sequence for Slovenia, Poland, Romania, Serbia (L2), Montenegro (L2). Without a field procedure, string data is stored based on the system/390 collating sequence, which causes two major problems.

The first is that system/390 collating is not the appropriate collating sequence for some European alphabets.

The second is that SQL/DS cannot collate double-byte vowels correctly; it considers the two component characters individually rather than as a single character.

The technical solutions for both problems is to code the field procedure for the affected columns. On the ZPIZ side the SLOSORT field procedure was customized. The original FP870L2 was changed according to the customer requirements.

```
* $$ JOB JNM=SLOSORT,CLASS=0
// JOB SLOSORT
// LIBDEF *,SEARCH=PRD2.SQL350
// LIBDEF *,CATALOG=PRD2.SQL350
// OPTION CATAL
   PHASE SLOSORT,*
// EXEC ASSEMBLY
         COPY SLOSORT        SQL/DS 350 provided field procedure FP870L2
         END
/*
// EXEC LNKEDT,PARM=¢MSHP¢
/*
/&
* $$ EOJ
```

Figure 101. Job to Assemble Field Procedure for Latin-2 Support

This SLOSORT assembly job could be used for the field procedure sample listed in A.3, "Field Procedure Sample" on page 199.

4.1.8 SQLPARM Catalog Job

```
* $$ JOB JNM=SQLCATAL,CLASS=0
// JOB SQLCATAL
// EXEC LIBR,PARM=¢MSHP¢
ACCESS SUBL=PRD2.SQL350
CATALOG SQLPARM.A REPL=Y
RMTUSERS=10
NCUSERS=30
DBNAME=SQLDB
CHARNAME=SLOVENIA
/+
/*
/&
* $$ EOJ
```

Figure 102. SQLPARM Catalog Job at ZPIZ

Please refer to Figure 58 on page 63 in 3.3.3, "Defining Slovenian Character Set" on page 62.

4.1.9 SQL Startup Job

```
* $$ JOB JNM=SQLDRDA,CLASS=8,DISP=K
// JOB SQLDRDA    START SQL/DS IN MULTIPLE USER MODE
// LIBDEF PROC,SEARCH=PRD2.SQL350
// EXEC  PROC=ARIS35PL    *-- SQL/DS PRODUCTION LIBRARY ID PROC
// EXEC  PROC=ARIS35DB    *-- SQL/DS DATABASE ID PROC
// EXEC  ARISQLDS,SIZE=AUTO,PARM=¢PARMID=SQLPARM¢
/*
/&
* $$ EOJ
```

Figure 103. SQL Startup Job at ZPIZ

4.1.10 Grant Schedule

```
* $$ JOB JNM=ARIS35FD,CLASS=8
// JOB ARIS35FD     GRANT SCHEDULE FOR DFHSIT APPLID
// LIBDEF PROC,SEARCH=PRD2.SQL350
// EXEC PROC=ARIS35PL        *-- SQL/DS PRODUCTION LIBRARY ID PROC
// EXEC PROC=ARIS35DB        *-- SQL/DS DATABASE ID PROC
// EXEC PROC=ARISDBSD        *-- RUN DBS UTILITY IN SINGLE USER MODE
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
GRANT SCHEDULE TO Z00CICS4 IDENTIFIED BY SQLDBAPW;
/*
/&
* $$ EOJ
```

Figure 104. Grant Schedule Z00CICS4 at ZPIZ

ITSO GRANT SCHEDULE TO CICSDRDA IDENTIFIED BY CICSDRDA

CIRB CICSDRDA,10,,,,SQLVSDB1

4.1.11 Create Sample Table with Field Procedure for Latin-2 Support

```
* $$ JOB JNM=ARIDBS,CLASS=4
// JOB ARIDBS
// LIBDEF PROC,SEARCH=PRD2.SQL350
// LIBDEF *,SEARCH=PRD2.SQL350
// EXEC  PGM=ARIDBS,SIZE=AUTO
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
SET ERRORMODE CONTINUE;

DROP TABLE LATIN2;
CREATE TABLE LATIN2 (A CHAR(10) FIELDPROC SLOSORT) IN SAMPLE;

INSERT INTO LATIN2 (A) VALUES(¢¶¶½½abc¢);
INSERT INTO LATIN2 (A) VALUES(¢zzzzabc¢);
INSERT INTO LATIN2 (A) VALUES(¢ååååabc¢);
INSERT INTO LATIN2 (A) VALUES(¢ÅåÅÅabc¢);
INSERT INTO LATIN2 (A) VALUES(¢æ’’efgh¢);
INSERT INTO LATIN2 (A) VALUES(¢0e‰½uwxy¢);

SELECT * FROM LATIN2 ORDER BY 1;

COMMIT WORK;
/*
/&
* $$ EOJ
```

Figure 105. Create Sample Table with Field Procedure at ZPIZ

4.1.12 Additional CCSID Support and Choosing a National Language and Character Set

As the number of CCSIDs supported on databases across DB2 platforms increases, the demand for support for conversion from these new CCSIDs will increase as well. These enhancements address this demand, by increasing the support available for Traditional Chinese, Simplified Chinese, Korean, Cyrillic, Greek and Windows code pages. In order to support the requested conversions, it is necessary to add support for two new CCSIDs on the host as well. The last release of SQL/DS V3.5 supports some new character sets at run time. These characters sets are added to the list of IBM-supplied character sets (Cyrillic, Greek-423). The Greek-423 Charname uses code page 423, character set 218 and CCSID 423. Unfortunately, at the present time, the Slovenian character set required by our customer ZPIZ is not available. A new character set was supplied for them. Please refer to 3.3.3, “Defining Slovenian Character Set” on page 62 for more details.

SQL/DS connections on VM/ESA at ZPIZ with CCSID 870 and Charname Slovenia were established to SQL/DS VSE/ESA at ITSO, where the database system was started with CCSID 500 and Charname International.

VSE/ESA SQL/DS with the following CCSID (CP/CS) is now enabled:

```
- ZPIZ CCSID 870 (Charname - SLOVENIA) <-----> ITSO 037(L1 USA)
- ZPIZ CCSID 870 (Charname - SLOVENIA) <-----> ITSO 500(L1 INTERNATIONAL)
```

Before you submit the JCL, make sure that SQL/DS is initiated with the private protocol SQLDS, so that conversion is not enabled.

```
>>>>>   Insert 4846 fig106 e6 here   <<<<<

* $$ JOB JNM=SLO,CLASS=0,DISP=L
* $$ LST CLASS=B,DEST=(*,VMSQLADM)
* $$ PUN CLASS=B,DEST=(*,VMSQLADM)
// JOB SLO
// LIBDEF *,SEARCH=PRD2.SQL350
// SETPFIX  LIMIT=100K
// EXEC ARIDBS,SIZE=AUTO
SET ERRORMODE CONTINUE;
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW TO SQLDS;
COMMENT ¢* INSERT NEW ROWS FOR SBCS and MIXED TRANSLATIONS *¢
DELETE FROM SYSTEM.SYSSTRINGS WHERE INCCSID=037 AND OUTCCSID=870;
INSERT INTO SYSTEM.SYSSTRINGS VALUES(037,870,¢SS¢, NULL,NULL,¢ ¢,
¢    ¢;          •      ¢,
¢ âäàáãåçñê.<(+]&éèëïíîìÀß|$*);Ã-/ÂÄÅÁÑøÇÊÈ,%_>?ÏÉÌËØÍÎ«»:#@¢¢=†‰abcde
fghi••æ-—Œ jklmnopqrÆ•¡Ÿ¿•‡stuvwxyz„[œ•£¥ fifl¶¼§½¾•‘¢!’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
DELETE FROM SYSTEM.SYSSTRINGS WHERE INCCSID=870 AND OUTCCSID=037;
INSERT INTO SYSTEM.SYSSTRINGS VALUES(870,037,¢SS¢, NULL,NULL,¢ ¢,
¢    ¢;          •      ¢,
¢ âäàáãåçñ•.<(+!&é¢ëêíîèïß‘$*);[-/ÂÄìÁ ÀÇÃ|,%_>?ÅÉÑË]ÍÎøÊ:#@¢¢=†Èabcde
fghiÏÌØ-«» jklmnopqr••‰Ÿ—•Œstuvwxyzæ/Eœ••¡¿‡„£¥§fifl¶¼½¾’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
DELETE FROM SYSTEM.SYSSTRINGS WHERE INCCSID=500 AND OUTCCSID=870;
INSERT INTO SYSTEM.SYSSTRINGS VALUES(500,870,¢SS¢, NULL,NULL,¢ ¢,
¢    ¢;          •      ¢,
¢ âäàáãåçñ¢.<(+|&éêëèíîïìß!$*); -/ÂÄÀÁÃÅÇÑø,%_>?ÊÉÈËÏÍÎÌØ:#@¢¢=†«abcde
fghi»‰•-•— jklmnopqrŒæÆŸ••¡stuvwxyz¿‡œ•„[£¥fifl¶§¼½¾•‘]’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
DELETE FROM SYSTEM.SYSSTRINGS WHERE INCCSID=870 AND OUTCCSID=500;
INSERT INTO SYSTEM.SYSSTRINGS VALUES(870,500,¢SS¢, NULL,NULL,¢ ¢,
¢    ¢;          •      ¢,
¢ âäàáãåçñ¢.<(+|&éêëèíîïìß!$*); -/ÂÄÀÁÃÅÇÑ‘,%_>?]ÉøËÊÍÎÈÏ:#@¢¢=†Ìabcde
fghiØ«»-‰• jklmnopqr•—ŒŸæ•Æstuvwxyz•¡œ•¿‡„[£¥fi§fl¶¼½¾•’“”†{ABCDEFGHIôöò
óõ}JKLMNOPQR•ûüùúÿ\STUVWXYZ•ÔÖÒÓÕ0123456789•ÛÜÙÚ¢);
COMMIT WORK;
/*
/&
* $$ EOJ

>>>>>   End of Insert   <<<<<
```

Figure 106. SYSTEM.SYSSTRINGS Table

Chapter 5. OS/2 Implementation

This chapter describes experiences with workstation and host setup in DRDA multi-language environments between ITSO Böblingen and ZPIZ Ljubljana. In order to establish connections, communication controllers, 3745 and 3174, were implemented at the ITSO and a 3745/NCP at ZPIZ.

The chapter is divided into:

1 Connectivity via 3745 ITSO to 3745 ZPIZ

2 Connectivity via 3174 ITSO to 3745 ZPIZ

5.1 Multi-Language Considerations

In order to connect to a data base with an incompatible code page, the following steps need to be performed. Although this information is correct at this time (Nov ′96), you should always check your DB2 manuals first, to see if there is more up-to-date information applicable to the same topic.

1. Make sure you are using DB2 server V2.1.2 or above.
2. Ask your IBM representative to obtain two conversion files for the pair of code pages that you want to support. For example, if you want to support connection between code page 819 and code page 915, you need 08190915.cnv and 09150819.cnv tables.
3. Place these binary files in the *<DB2Path>/sqllib/conv* directory on UNIX platforms such as AIX, and in the *<DB2Path>/sqllib/conv* directory on various Intel platforms such as OS/2 or Windows.
4. If you are only using Common Server DB2 C/S products, you need to place these files only on the server.
5. For connections between Common Server DB2 C/S and mainframe DB2 implementations, such as SQL/DS or DB2/MVS, you need to first ensure that the corresponding DB2 host platform supports the same conversion. You only need the Host-to-Workstation conversion table on the workstation side.
6. Once the connection between incompatible code page environments is established, you will get a warning message:

```
SQL0863W  A successful connection was made, but only single byte
characters should be used.  SQLSTATE=0153.
```

Figure 107. Connection Information

This is normal, and intended to warn you that you should only use the characters that exist in both code pages.

5.2 Connectivity via 3745 ITSO to 3745 ZPIZ

In our environment, independent LUs via 3745 are defined as static with CDRSC. The ITSO′s Token-Ring LAN is connected to the host via the 3745. In this environment the 3745 is shared by multiple systems. The owner of NCP is the ISC system. 3745 has two TICs (Token-Ring Interface Connector/Coupler): TIC1 and TIC2. The owner of TIC is the ISC system, that of TIC2 is the IS2 system. All LUs via 3745 are defined using TIC2 of the IS2 system. Therefore, the VTAM switched major node definition is defined by IS2 VTAM. All panels and files are given in A.1, "VM/VTAM Definitions - BOEVMIS2" on page 191 for BOEVMISC and A.2, "VSE/VTAM Definitions - BOEVMIS2" on page 196 for BOEVMIS2. The IPFNV3/NCP must be activated on both machines.

In the switched major node, independent LUs are defined as LOCADDR=00. In this sample IPFSL0E0 is defined as a predefined independent LU. CPNAME, or IDBLK and IDNUM should be defined for XID exchange (refer to Figure 207 on page 195). IPFSL0E0, which is an independent LU via 3745, uses the static LU definition. The CDRSC major node is ISTPDILU, which is created dynamically. The adjacent link station is IPFCP20E, which is defined in IPF2SWI0 using the static LU definition, ISTPDILU which is created automatically. When the LU has a session such as IPFSL0E0, you can display the NETID of this LU (d net,id=istpdilu,e).

```
IPFSL0E0  -  static LU independent definition.
IPF2SWI0  -  CDRSC major node.
ISTPDILU  -  The CDRSC  major node.
IPFCPS0E  -  adjacent link station.
```

5.2.1 AVSVM

AVSVM is part of the VM operating system and handles communications between VM and non-VM systems in the SNA network. AVS translates protocol between APPC/VM and APPC/VTAM and enables the SNA network to allocate conversation with global resources. AVS is a VTAM application that runs with VTAM either in a separate virtual machine in the same GCS group or in the VTAM virtual machine in a GCS group. AVS should be defined as a VTAM application major node. A gateway must be activated before use. This can be done interactively or in a profile in the AVS virtual machine. This profile is named AGWPROF GCS. (Refer also to Figure 208 on page 195).

5.2.2 SQL/DS

SQL/DS for VM/ESA and VSE/ESA must be installed with DRDA code. DRDA can be installed using the appropriate files which are already predefined (for VM/ESA ARISDBMA) and are to be found in the publications for SQL/DS Installation for VM and VSE.

5.2.3 OS/2

- Multi-Protocol Transport Services (MPTS) Customization
- OS/2 Communication Manager Customization via 3745
- DB2/2 and DDCS/2 Customization

5.2.4 MPTS Customization

Multi-Protocol Transport Services (MPTS) provides support that enables OS/2 to operate on a network by using Network Driver Interface Specification (NDIS). NDIS is a medium access control (MAC) interface for local area network (LAN) adapter drivers and protocol driver. NDIS handles the basic transmission and reception of packets on the network.

After you invoke MPTS, the logo window is displayed. When you edit the configuration in the Current Configuration list box, you will get the Parameters window. You must enter the Medium Access Control address (MAC) of the Token-Ring Adapter in your workstation into the Network Adapter Address parameter. You may get this address from the Token-Ring administration or host connection administrator.

The following panel show parameters that are selected in the Configure Workstation window.

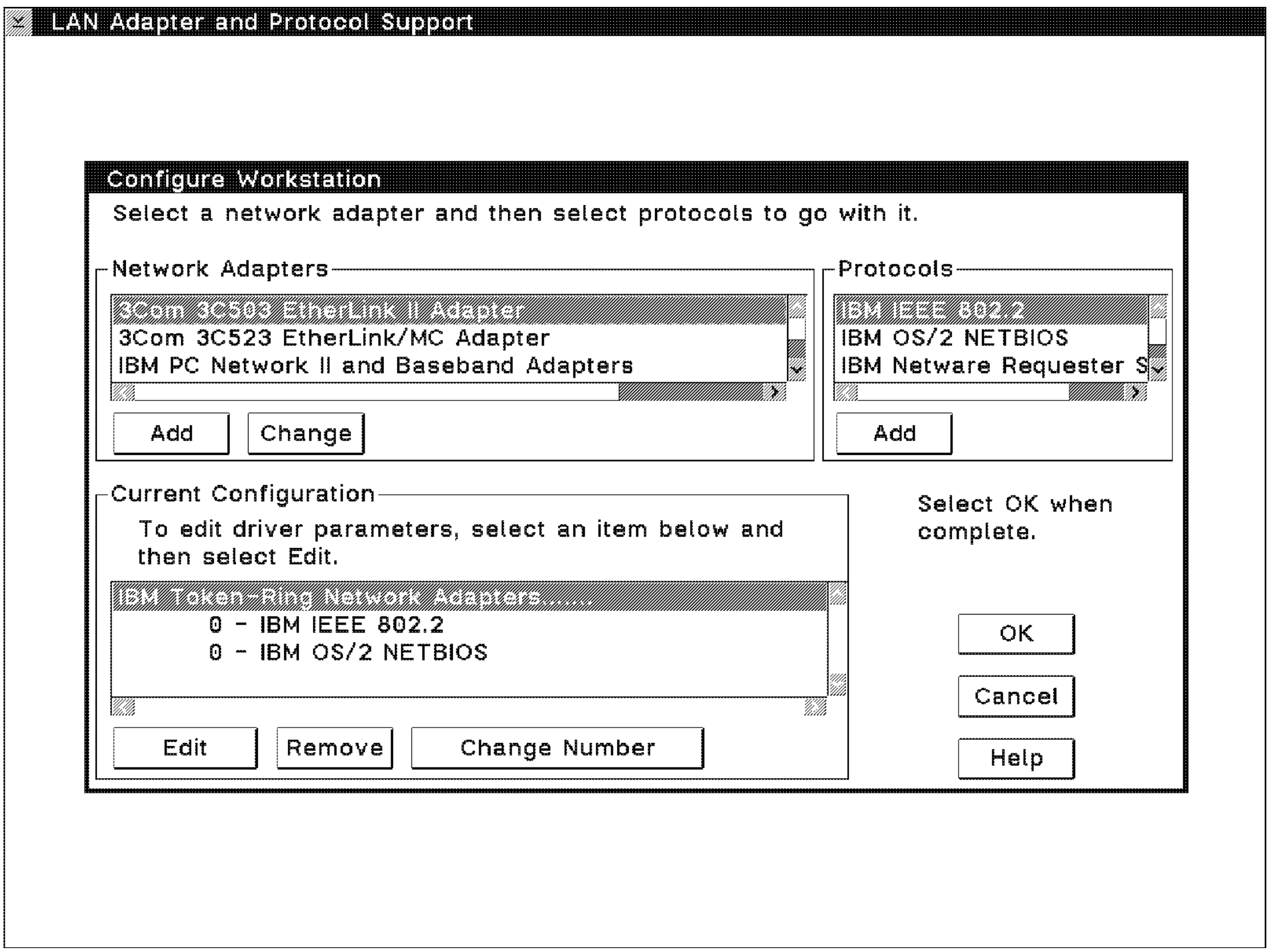


Figure 108. MPTS Configuration Menu

For our installation we selected the IBM Token-Ring Network Adapter with three protocols:

- *IBM IEEE 802.2*
- *IBM OS/2 NETBIOS*
- *IBM TCP/IP*

The IBM IEEE 802.2 protocol allows us to communicate with our host CICS/VSE using APPC over the Token-Ring.

The NETBIOS protocol allows us to communicate with our DB2/2 Client Application Enabler (CAE) over the Token-Ring.

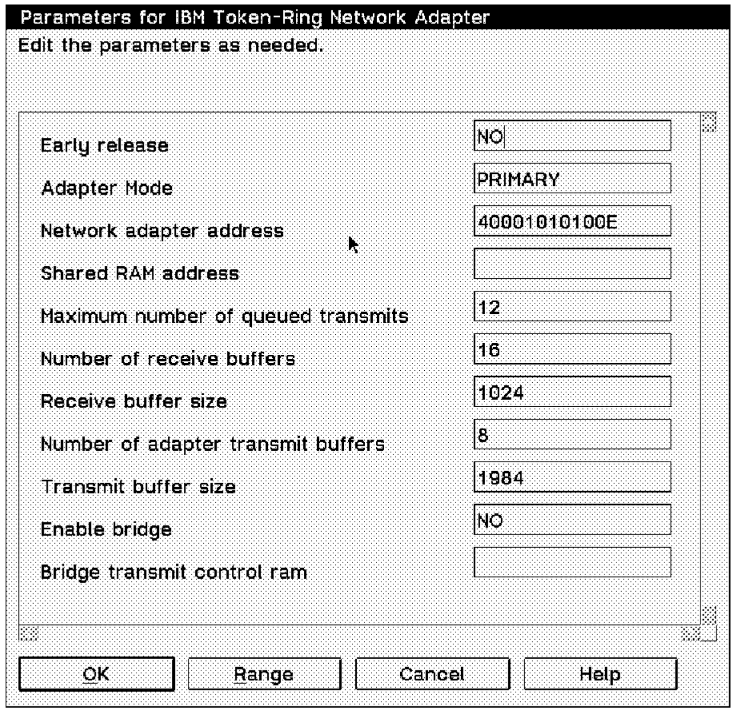


Figure 109. IBM Token-Ring Network Adapter Parameters

The address specified in the Network adapter address field must match the last six bytes of the PATH macro's DIALNO parameter in the switched major node definition described below.

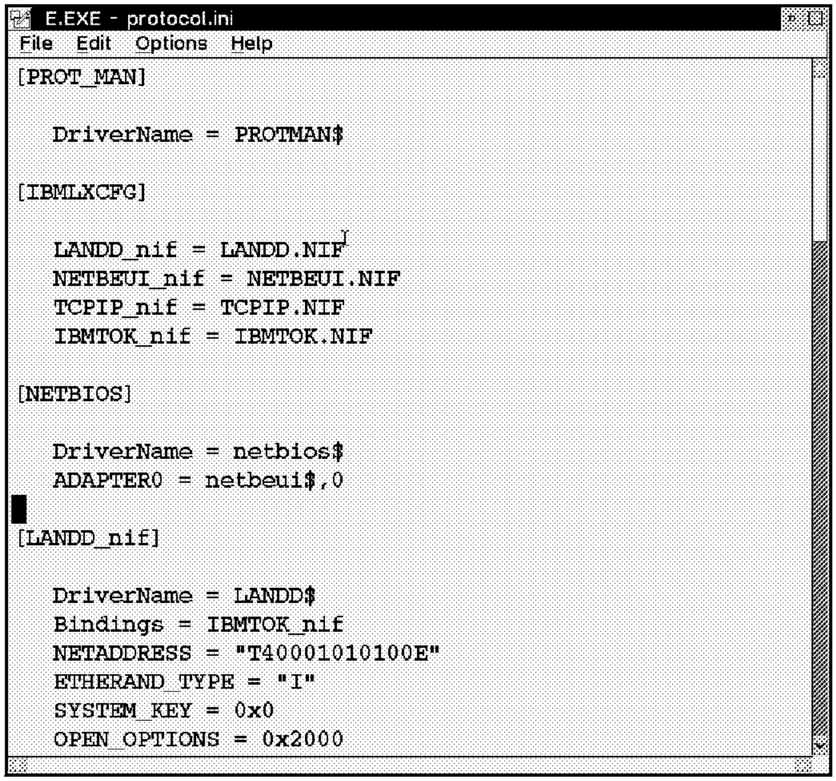


Figure 110. C:\IBMCOM\PROTOCOL.INI

```
*****************************************************************
*
* PS/2 FOR DRDA
*****************************************************************
IPFCPS0E PU    ADDR=0E,            COULD BE ANYTHING (NOT USED)       C
               IDBLK=05D,          IDBLK OF OS/2 COMM. MANAGER        C
               IDNUM=E000E,        IDENTIFICATION NUMBER              C
               DISCNT=NO,          VTAM DOES NOT HANG UP              C
               MAXOUT=1,           MAXIMUM NUBER OF PIUS              C
               MAXDATA=1920,       MAXDATA FOR OS/2 COMM. MANAGER     C
               USSTAB=ISTSNA,                                         C
               MODETAB=AMODETAB,                                      C
               DLOGMOD=MSDLCQ,                                        C
               MAXPATH=1,          NUMBER OF DIAL OUT PATHS TO PU     C
               VPACING=2,                                             C
               PUTYPE=2,                                              C
               CPNAME=CPS0E,                                          C
               ISTATUS=ACTIVE
         PATH  DIALNO=020440001010100E,                               C
               GRPNM=IPFG3L89,     LOGICAL GROUP NAME OF TIC 2        C
               GID=1,                                                 C
               PID=2
IPFSL0E0 LU    LOCADDR=00,MODETAB=DRDAMOD,DLOGMOD=IBMRDBM
IPFT2TEA LU    LOCADDR=02
IPFT2TEB LU    LOCADDR=03
IPFT2TEC LU    LOCADDR=04
IPFT2TED LU    LOCADDR=05
```

Figure 111. IPF2SWI VTAMLIST - Switched Major Node

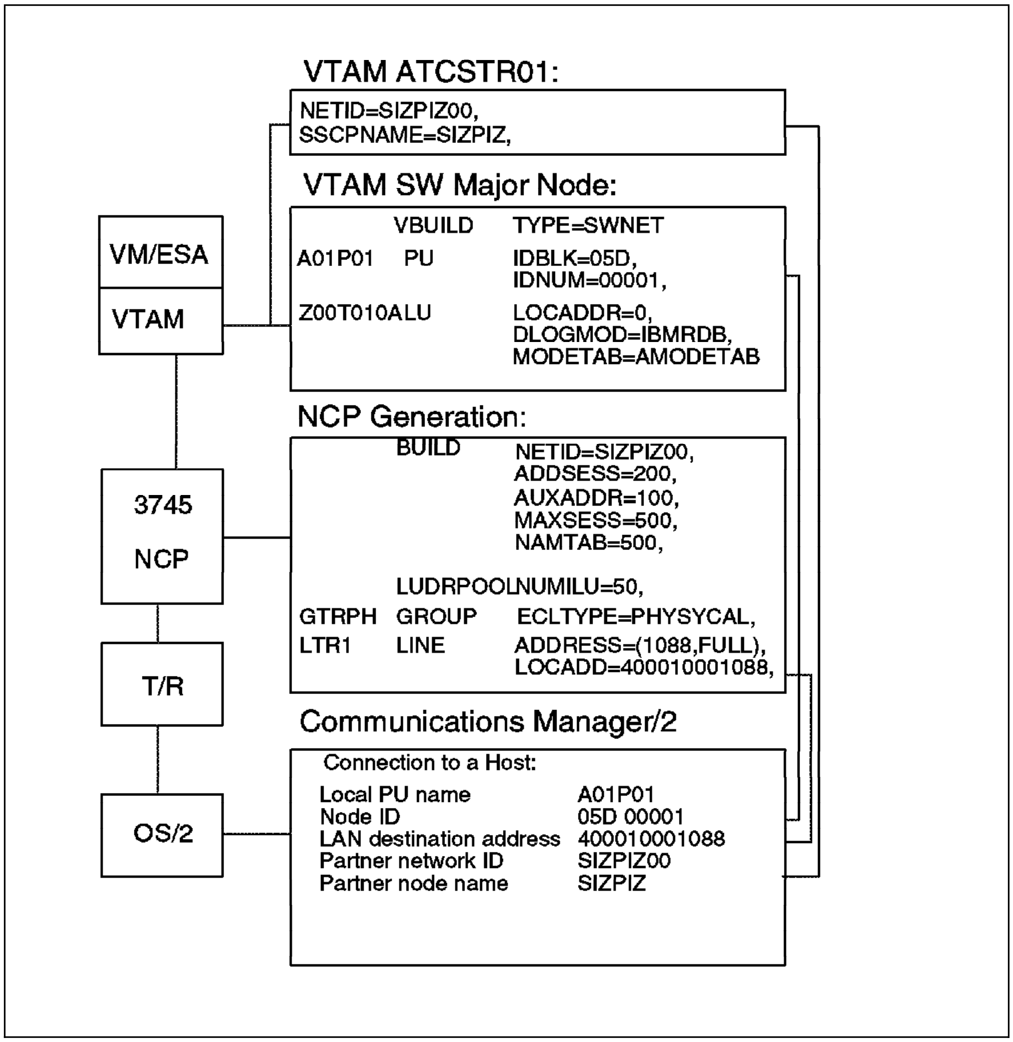


Figure 112. Definitions and Correlations VTAM, NCP, CM/2 at ZPIZ

As from VTAM 4.2, the CPNAME parameter within the Switched Major Node definition must be different to the PU name. After you have finished configuring MPTS, you have to re-boot the PS/2.

5.2.5 CM/2 Customization

CM/2 is a communication service platform providing a wide range of services for IBM OS/2 that communicate with host computers and other PWS. Before using the CM/2 in conjunction with DB2/2 and DDCS/2 to establish an LU6.2 session to the host, you must customize a configuration file to support Advance Program to Program Communication.

The following definitions are required for APPC (LU6.2):

- DLC Adapter Parameters definition, which specifies Data Link Control characteristics of the Token-Ring adapter.
- SNA Local node Characteristics definition, which specifies characteristics that are common to all APPC users on the workstation.
- At least one SNA Connection definition.
- The necessary optional SNA Features for SNA LU6.2 communication to the VSE/ESA and VM/ESA host which consists of:

– A **Local Logical Unit** definition for the local Queue Manager.
– A **Partner Logical Unit** definition for each remote Queue Manager system with which this workstation wants to communicate.
– A **MODE** definition to specify sets of session properties that are used in binding APPC sessions.
– A **Transaction Program** definition for every local transaction invoked when messages are sent to or received from a remote system.

Before setting up the communication between the workstation running OS/2 we have summarized the SNA and DRDA parameters. Naming conventions were introduced to enable easier customization of all software involved.

Table 9. SNA Customization for ITSO and ZPIZ for DRDA Environment via 3745

	ITSO Böblingen		**ZPIZ Ljubljana**	
	SQL/DS VSE	SQL/DS VM	SQL/DS VSE	SQL/DS VM
Network id	DEIBMIPF	DEIBMIPF	SIZPIZ00	SIZPIZ00
Controller Address	400010301002		400010001088	
IDBLK and IDNUM	05D E000E		05D 00001	
Local Node Name	IPFCPS0E	IPFCPS0E	A01P01	A01P01
Partner Node Name (SSCP)	IPFV2A	IPFV2	A05M	SIZPIZ ″SIZPIZ6″
Fully Qualified Partner LU Name	DEIBMIPF. IPFA21CD	DEIBMIPF. IPFA2GL4	SIZPIZ00. Z00CICS4	SIZPIZ00. Z00AVSL1
Partner LU Alias	VSEANL13	BOEVMIS2	Z00VS374	Z00VM374
Local LU Name	IPFSL0E0	IPFSL0E0	Z00T010A	Z00T010A
Local LU Alias	IPF3745	IPF3745	Z00T01	Z00T01
Mode Name	IBMRDBM	IBMRDBM	IBMRDB	IBMRDB
Transaction Program Name	TPN1	SQLVMDB0	TPN1	SQLDBA
Security	PGM	PGM	PGM	PGM
Communication Type	APPC	APPC	APPC	APPC
CPI Communication	IPFVS1S5	IPFVM1S5	Z00VS1S5	Z00VM1S5

Note:
1. IPF stands for Böblingen,Germany. Z00 for Ljubljana, Slovenia
2. VM stands for VM. VS stands for VSE.
3. 5 as the last character means that the 3745 communication controller was used.
4. S stands for symbolic destination name.
5. TPN1 (VSE - ceda i tran(tpn1). SQLVMDB0 (VM - IUCV res. id.)
6. PU ipfcps0e (d net,id=ipfsl0e0..... alslist = ipfcps0e).

The following are guidelines for the relevant parameters in the Communications Manager Profile List Sheet in the CM/2 configuration definition. They describe parameters appropriate for defining APPC APIs on a Token-Ring.

Please note that we have shown only the screens with ITSO parameters. Differing values for ZPIZ are listed below each screen where appropriate.

To start with the CM/2 customization:

1 Double click on the CM/2 folder

2 Double click on the CM/2 Setup icon and in the CM/2 *Installation and Setup* window, press the **OK** button

3 In the *CM Setup* window, select the **Setup...** button

4 In the *Open Configuration* window, enter configuration name and press the **OK** button

5.2.5.1 Configuring APPC APIs through the Token-Ring

We have used the Communications Manager setup to configure APPC APIs through Token-Ring. On the Communications Manager Configuration Definition window we have checked Additional definitions and selected Token-ring or other LAN types and APPC APIs.

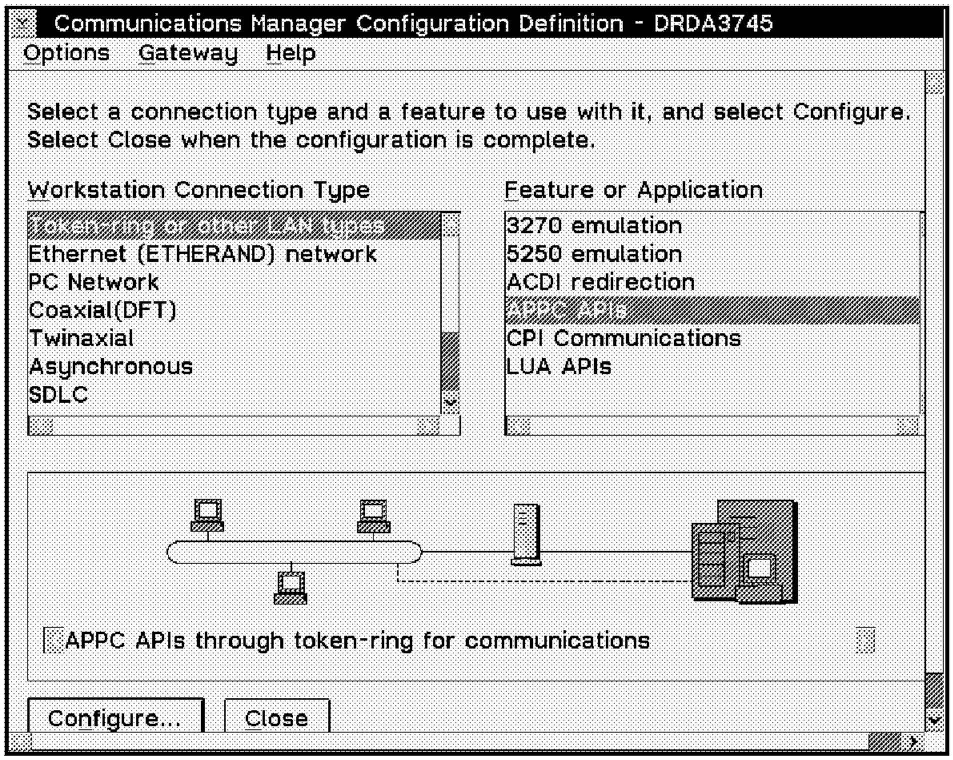


Figure 113. CM/2 Configuration Definition

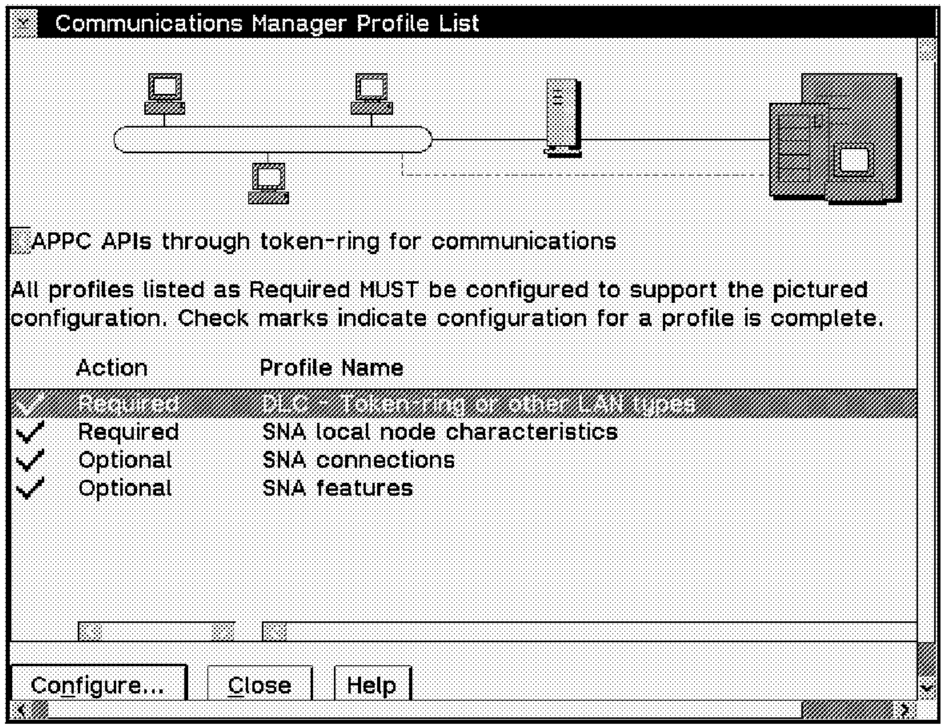


Figure 114. CM/2 Profile List

By selecting the DLC - Token-ring or other LAN types CM/2 profile name we have a window that enables us to configure the Data Link Control profile.

5.2.6 Configuring DLC Token-Ring

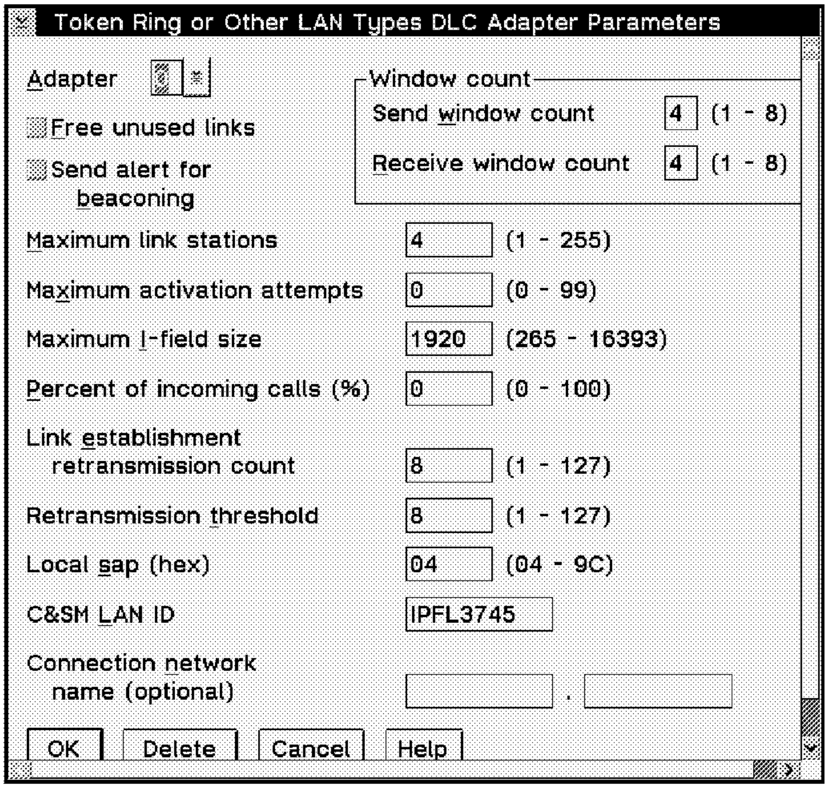


Figure 115. Token Ring DLC Adapter Parameters

ZPIZ C&SM LAN ID SIZPIZ01

You can take most defaults here. You should enter a name for the following parameter:

C&SM LAN ID. To identify the ring - used by system management products.

Note: If you increase the maximum number of link stations parameter, be sure to set the link station parameter in Network Transport Services/2 (NTS/2) accordingly.

5.2.7 Defining Local Node Characteristics

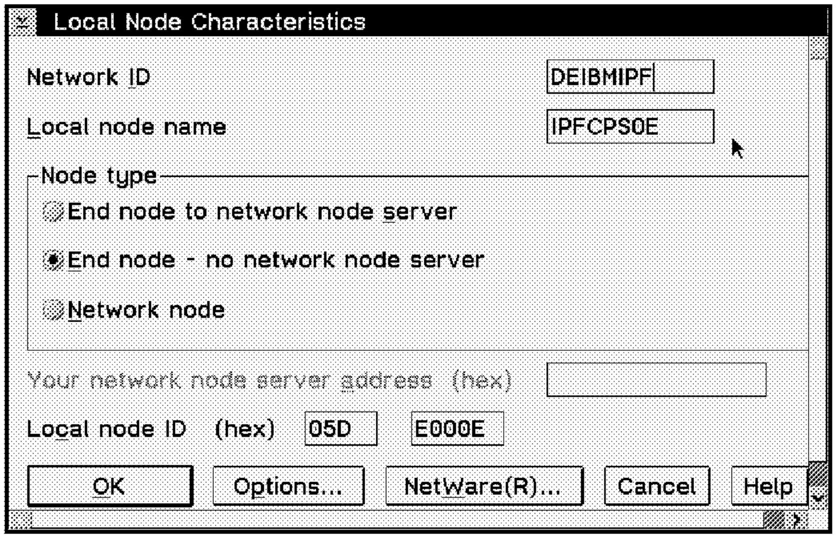


Figure 116. Local Node Characteristics

ZPIZ Network ID.............SIZPIZ00
Local node name.........A01P01
Local node ID (hex).... 05D 00001

The key fields are:

Network ID *DEIBMIPF* is the name of the network to which this PU belongs. This must match the NETID parameter of the VTAM start list.

Node type PS/2 is an end node accessed by the host through a 3745 gateway.

Local Node ID The values *05D E000E* are used for the XID exchange. They match IDBLK and IDNUM values specified on our PU definition statement in the VTAM Switched Network major node as in IPF2SWI0 VTAMLST (refer to Figure 207 on page 195).

Local Node Options

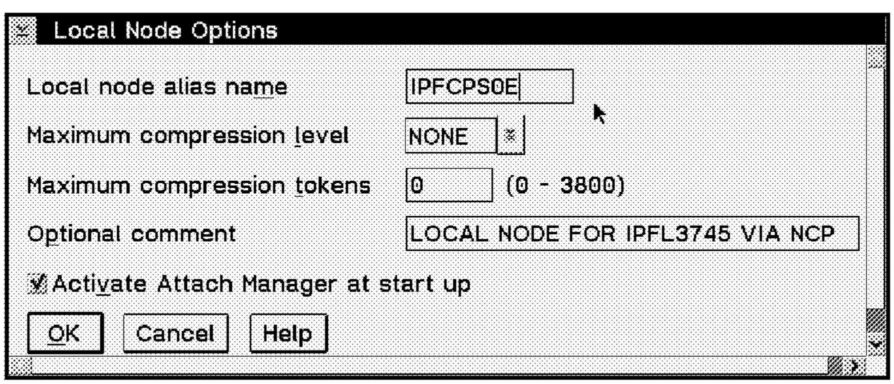


Figure 117. Local Node Options

ZPIZ Local Node Alias Name...A01P01

The key field is:

Activate Attach Manager
This is a feature of APPC which manages incoming requests to begin a communication session. In our case, the Attach Manager is started automatically when the Communications Manager is started.

5.2.8 Defining SNA Host Connections

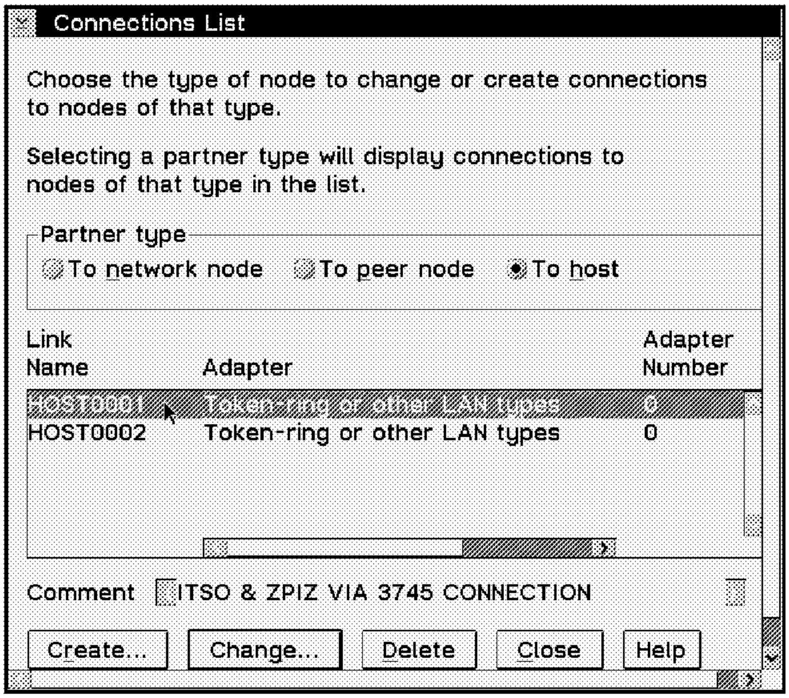


Figure 118. Connection Definition Panel 1 of 3

Select **To host**, as ***Partner type*** and click on the ***Create...*** button.

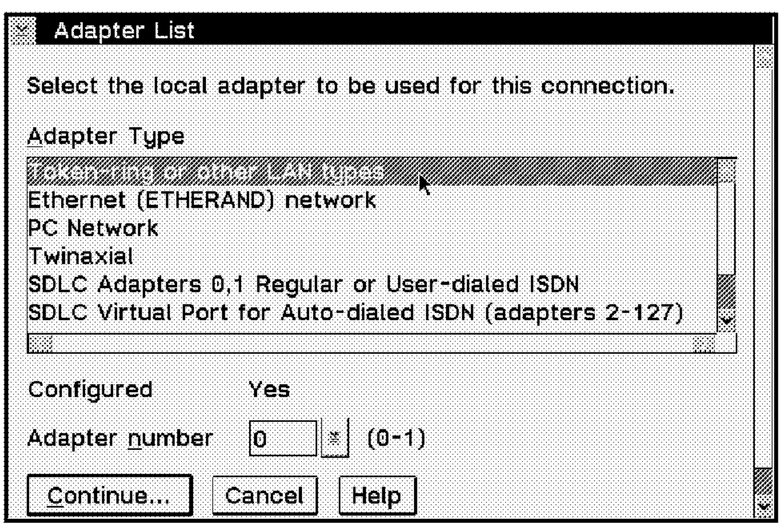


Figure 119. Connection Definition Panel 2 of 3

Select appropriate **Adapter type** and click on to the ***Continue...*** button.

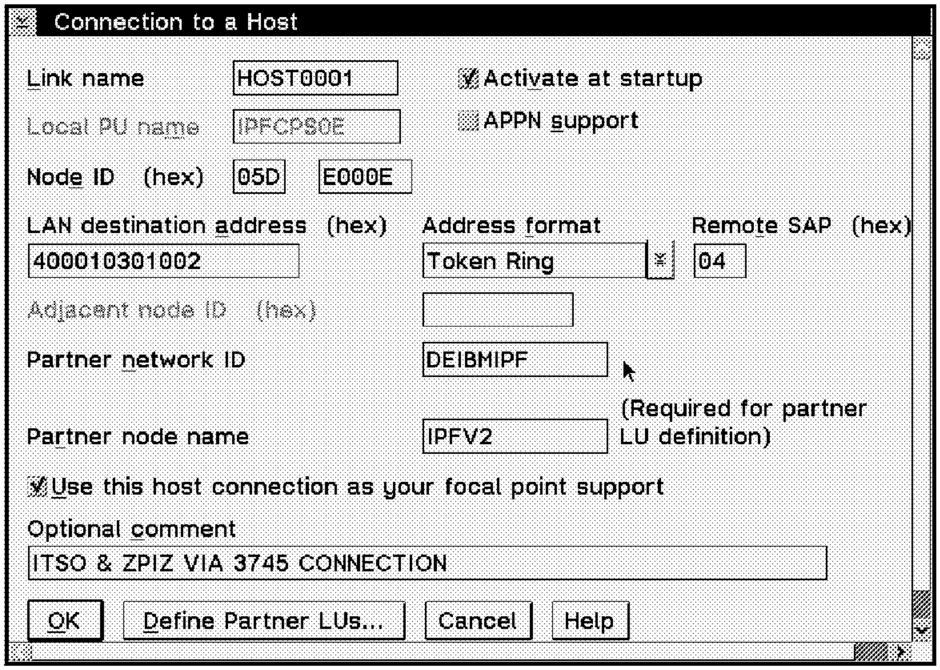


Figure 120. Connection Definition Panel 3 of 3

ZPIZ Local PU Name..................A01P01
APPN support...................checked
Node ID (hex)..................05D 00001
LAN Destination address (hex).. 400010001088
Partner network ID.............SIZPIZ00
Partner node name..............SIZPIZ

Please check ***APPN support***, but not ***Activate at startup***. In this case the link will be activated by the partner′s incoming connection. The Partner network ID and the Partner node name correspond to the host CP definitions on the ZPIZ side.

The key fields are:

Link Name *HOST0001* is the link name of the connection to the host.

LAN Destination address
400010301002 is the LAN MACADDRess of the 3745 gateway to our host.

Partner network ID
DEIBMIPF is the NETID of the host VTAM as specified in the VTAM start list.

Partner node name
IPFV2 VM/VTAM name (owner of NCP).

Node ID *05D E000E* are the IDBLK and IDNUM parameters as specified in the VTAM switched major node.

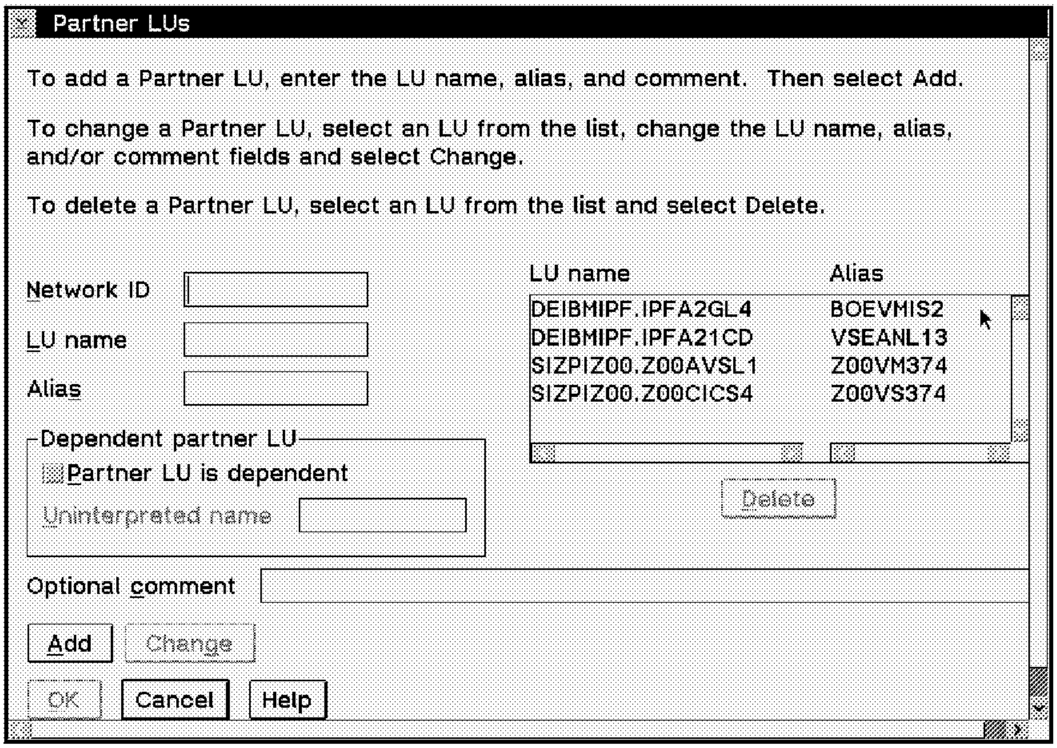


Figure 121. Partner LUs

ZPIZ	Network ID / LU Name / Alias...	SIZPIZ00.Z00AVSL01 VM122AVS
	Network ID / LU Name / Alias...	SIZPIZ00.Z00CICS4 VSE135
	Network ID / LU Name / Alias...	DEIBMIPF.IPFA2GL4 BOBAVS
	Network ID / LU Name / Alias...	DEIBMIPF.IPFA21CD BOBCICS

In the Partner LUs window, ZPIZ has defined four partner LUs. The VM122AVS and VSE135 alias names define the LUs with the network ID of the local node, and BOBAVS and BOBCICS alias names represent LUs defined in another network. We must use this method whenever the node does not have a connection to an APPN Network Node Server. We can also use this method when we are not sure about the capabilities of our partner node.

```
DDCS/2 workstation gateway (IPF3745:       NETBIOS over Token-Ring

DB2/2 Manager configuration

 Server  name = IPF3745                               DB2/2
                                                      Server

System Database Directory DB Connection Services Directory

  DB Alias = IPFVM1A5      Local DBN = IPFVM1D5
  DB Name  = IPFVM1D5      Target DBN= SQLVMDB0
  WS Name  = IPFVM1N5
                           PARM =          D:\SQLLIB\DCS1ARI.MAP

 Node Directory

  WS Name    = IPFVM1N5                                DDCS/2
  Network ID = DEIBMIPF                               Application
  Partner LU Alias = BOEVMIS2                          Requester
  Local LU Alias   = IPF3745
  Mode Name        = IBMRDBM

 Communications Manager

  Network ID       = DEIBMIPF
  Mode Name        = IBMRDBM
  Local LU Alias   = IPF3745
  Local LU Name    = IPFSL0E0
  Partner LU Alias = BOEVMIS2
  Partner LU Name  = IPFA2GL4


Host System:

VM/VTAM                                AVSVM

ATCSTR: NETID=DEIBMIPF   AGW ACTIVATE GATEWAY IPFA2GL4 GLOBAL
SNA Maj Node:            AGW CNOS IPFA2GL4 IPFSL0E0 IBMRDBM 20 10 10
  LU = IPFA2GL4          System User Directory:
  LU = IPFSL0E0
  DLOGMOD = IBMRDBM          IUCV *IDENT SQLVMDB0 GLOBAL


                         SQL/DS (Application Server)

                          User ID = DRDAX1
                          DBNAME = SQLVMDB0
```

Figure 122. DB2/2 DB Server to DDCS/2 Gateway to SQL/DS VM/ESA ITSO via 3745

```
DDCS/2 workstation gateway (IPF3745):      NETBIOS over Token-Ring

DB2/2 Manager configuration

 Server  name = IPF3745                                  DB2/2
                                                         Server

System Database Directory DB Connection Services Directory

  DB Alias = IPFVS1A5      Local DBN = IPFVS1D5
  DB Name  = IPFVS1D5      Target DBN= SQLVSDB1
  WS Name  = IPFVS1N5
                           PARM =       D:\SQLLIB\DCS1ARI.MAP

  Node Directory

  WS Name    = IPFVS1N5                                   DDCS/2
  Network ID = DEIBMIPF                                 Application
  Partner LU Alias = VSEANL13                            Requester
  Local LU Alias   = IPF3745
  Mode Name        = IBMRDBM

 Communications Manager

  Network ID       = DEIBMIPF
  Mode Name        = IBMRDBM
  Local LU Alias   = IPF3745
  Local LU Name    = IPFSL0E0
  Partner LU Alias = VSEANL13
  Partner LU Name  = IPFA21CD


Host System:

 VSE/VTAM                                  CICS/VSE

 ATCSTR: NETID=DEIBMIPF   CEDA DEF CONN(L0E0) Netname(IPFSL0E0)
 SNA Maj Node:
   APPL LU = IPFA21CD
   LU = IPFSL0E0
   DLOGMOD = IBMRDBM      DBNAME Dir (ARISDIRD): TPN1 SQLVSDB1


                          SQL/DS (Application Server)


                           DBNAME = SQLVSDB1
```

Figure 123. DB2/2 DB Server to DDCS/2 Gateway to SQL/DS VSE/ESA ITSO via 3745

```
DDCS/2 workstation gateway (IPF3745:        NETBIOS over Token-Ring

DB2/2 Manager configuration

 Server  name = IPF3745                               DB2/2
                                                      Server

System Database Directory DB Connection Services Directory

  DB Alias = Z00VM1A5      Local DBN = Z00VM1D5
  DB Name  = Z00VM1D5      Target DBN= SQLDBA
  WS Name  = Z00VM1N5
                           PARM =          D:\SQLLIB\DCS1ARI.MAP

 Node Directory

  WS Name    = Z00VM1N5                                 DDCS/2
  Network ID = DEIBMIPF                               Application
  Partner LU Alias = Z00VM374                          Requester
  Local LU Alias   = IPF3745
  Mode Name        = IBMRDB

 Communications Manager

  Network ID       = SIZPIZ00
  Mode Name        = IBMRDB
  Local LU Alias   = IPF3745
  Local LU Name    = IPFSL0E0
  Partner LU Alias = Z00VM374
  Partner LU Name  = Z00AVSL1


Host System:

VM/VTAM                                    AVSVM

ATCSTR: NETID=SIZPIZ00   AGW ACTIVATE GATEWAY Z00AVSL1 GLOBAL
SNA Maj Node:            AGW CNOS Z00AVSL1 IPFSL0E0 IBMRDBM 20 10 10
  LU = Z00AVSL1          System User Directory:
  LU = IPFSL0E0
  DLOGMOD = IBMRDB           IUCV *IDENT SQLDBA GLOBAL


                         SQL/DS (Application Server)

                          User ID = Z00VMU1
                          DBNAME = SQLDBA
```

Figure 124. DB2/2 DB Server to DDCS/2 Gateway to SQL/DS VM/ESA ZPIZ via 3745

```
DDCS/2 workstation gateway (IPF3745):      NETBIOS over  Token-Ring

DB2/2 Manager configuration

 Server  name = IPF3745                                  DB2/2
                                                         Server

System Database Directory DB Connection Services Directory

  DB Alias = Z00VS1A5      Local DBN = Z00VS1D5
  DB Name  = Z00VS1D5      Target DBN= SQLVSDB1
  WS Name  = Z00VS1N5
                           PARM =      D:\SQLLIB\DCS1ARI.MAP

  Node Directory

  WS Name    = Z00VS1N5                                   DDCS/2
  Network ID = DEIBMIPF                                 Application
  Partner LU Alias = Z00VS374                            Requester
  Local LU Alias   = IPF3745
  Mode Name        = IBMRDB

 Communications Manager

  Network ID       = SIZPIZ00
  Mode Name        = IBMRDB
  Local LU Alias   = IPF3745
  Local LU Name    = IPFSL0E0
  Partner LU Alias = Z00VS374
  Partner LU Name  = Z00CICS4


Host System:

 VSE/VTAM                                  CICS/VSE

 ATCSTR: NETID=SIZPIZ00   CEDA DEF CONN(DRC6) Netname(IPFSL0E0)
 SNA Maj Node:
   APPL LU = Z00CICS4
   LU = IPFSL0E0
   DLOGMOD = IBMRDB       DBNAME Dir (ARISDIRD): TPN1 SQLDS


                          SQL/DS (Application Server)


                           DBNAME = SQLDS
```

Figure 125. DB2/2 DB Server to DDCS/2 Gateway to SQL/DS VSE/ESA ZPIZ via 3745

5.2.9 Defining Optional SNA Features

We defined options for the following features:

- Local LUs
- Partner LUs
- Modes
- Conversation security
- CPI Communications

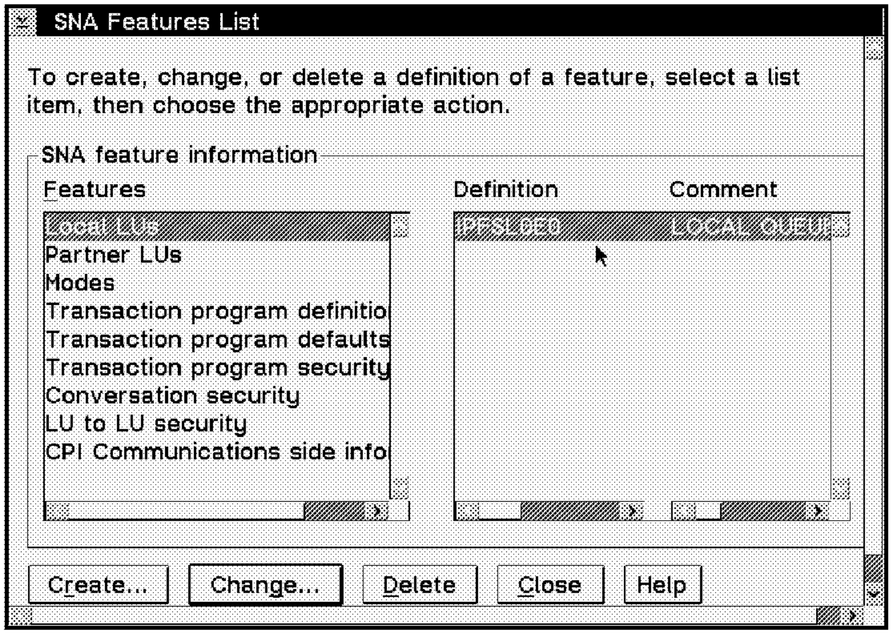


Figure 126. List of Optional SNA Features

5.2.9.1 Defining Local LU

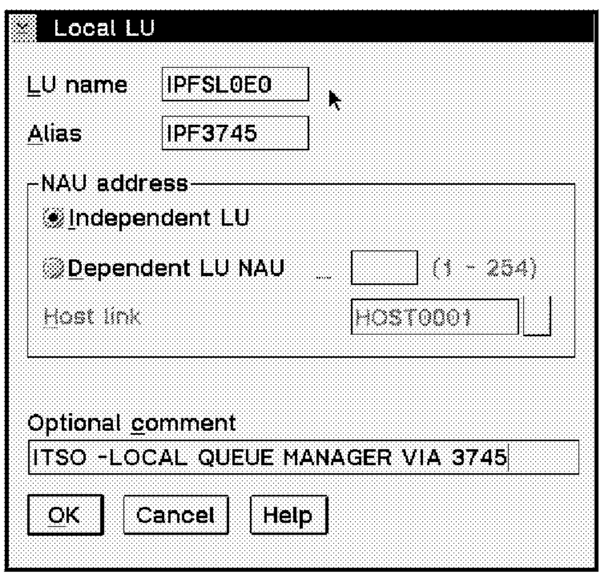


Figure 127. Local LU Definition Panel

ZPIZ LU name.........................Z00T010A
Alias...........................Z00T01

Normally we do not have to define a local LU because, as stated earlier, our local node name is also the default LU.

Note: When you define your local LU for your DDCS workstation, be sure to select ***Independent LU*** for the network accessible unit (NAU) address.

To ensure that our workstation uses this local LU when DDCS starts, we must set the OS/2 APPCLLU environment variable. We have added the following statement to our CONFIG.SYS file:

SET APPCLLU=Z00T01

The key fields are:

- **LU Name.** *IPFSL0E0* is the name by which the Logical Unit is known throughout the SNA network. It is defined in the VTAM switched network major node.
- **Alias.** *IPF3745* is the alias name by which transaction programs may refer to the LU.
- IPFSL0E0 is defined as an **Independent LU**. For more information about independent LUs please refer to Figure 207 on page 195.

5.2.9.2 Defining the Partner LU

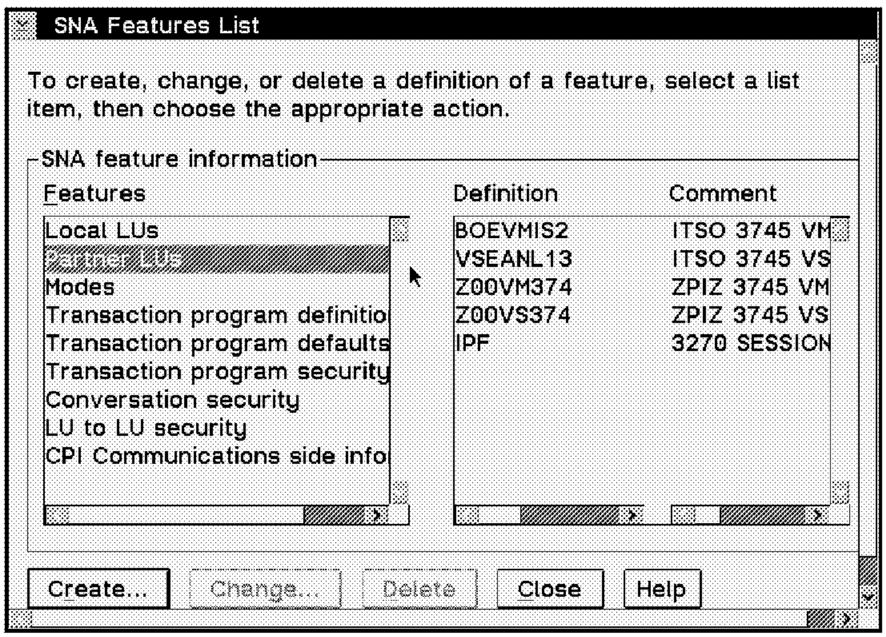


Figure 128. SNA Partner LUs for ITSO and ZPIZ

ZPIZ	Network ID	LU Name	Alias...	SIZPIZ00.Z00AVSL01	VM122AVS
	Network ID	LU Name	Alias...	SIZPIZ00.Z00CICS4	VSE135
	Network ID	LU Name	Alias...	DEIBMIPF.IPFA2GL4	BOBAVS
	Network ID	LU Name	Alias...	DEIBMIPF.IPFA21CD	BOBCICS

Partner LU information includes the fully qualified name and the alias name of the remote LU. We must configure a partner LU for every partner to which we want to refer by an alias name.

The fully qualified name of the partner LU consists of the Network ID of the partner′s LU and the partner′s LU name. We also have to enter the alias, which is a local name for the partner LU.

Note: If you use an APPN connection, you do not have to define a partner LU. When an EN (APPN) is connected to a serving NN and the resource is not found in the local directory, CM/2 directs the request to the serving NN, and a search request takes place in the network node. When an LEN (APPC) is connected to a serving NN and the resource is not found in the local directory, no further search is done by CM/2.

The key fields are:

Fully qualified LU name
Comprises the network ID and the partner LU name.
DEIBMIPF is the ID of the network on which the partner LU resides.
BOEVMIS2, VSEANL13, Z00VM374, Z00VS374 are the names of the partner LUs residing on the network.

Alias *BOEVMIS2, VSEANL13, Z00VM374, Z00VS374* are the alternative names or alias names by which transaction programs may refer to the LUs.

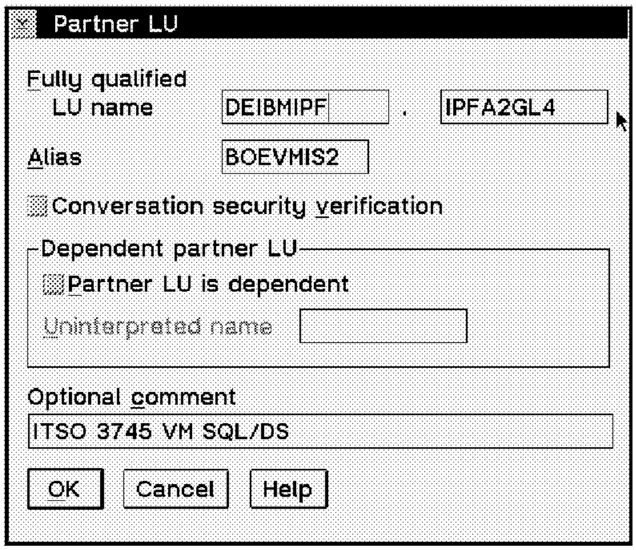


Figure 129. Partner LU - ITSO (AVSVM)

ZPIZ Network ID / LU Name.....SIZPIZ00.Z00AVSL1
Alias.....VM122AVS

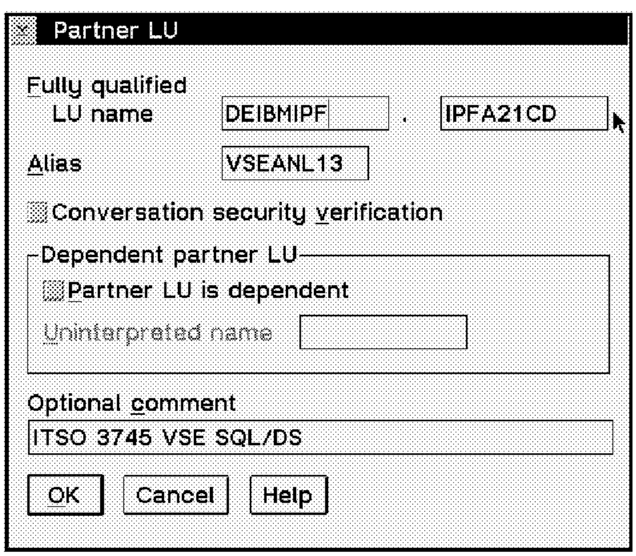


Figure 130. Partner LU - ITSO (CICS SIT APPLID)

ZPIZ Network ID / LU Name.....SIZPIZ00.Z00CICS4
Alias... VSE135

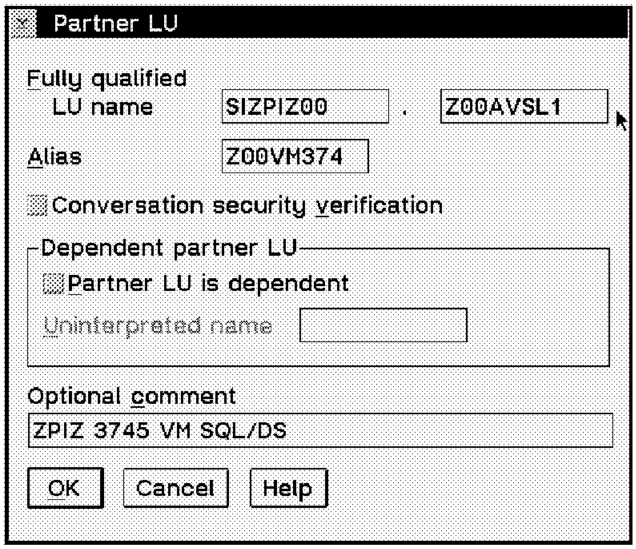


Figure 131. Partner LU - ZPIZ (AVSVM)

ZPIZ Network ID / LU Name.....DEIBMIPF.IPFA2GL4
Alias.....BOBAVS

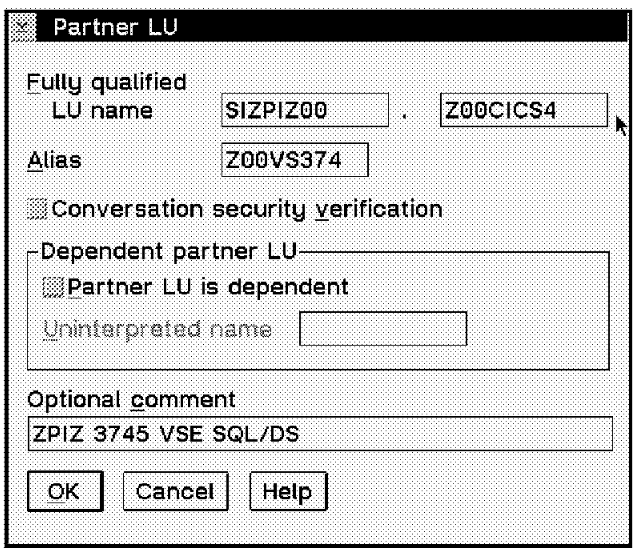


Figure 132. Partner LU - ZPIZ (CICS SIT APPLID)

ZPIZ Network ID / LU Name.....DEIBMIPF.IPFA21CD
Alias.....BOBCICS

5.2.9.3 Defining the Mode Definition

When establishing a session between two LUs, we must specify a mode. The mode contains information about the characteristics of the session.

Note: To establish a connection, the same mode name must exist on both sides of the link.

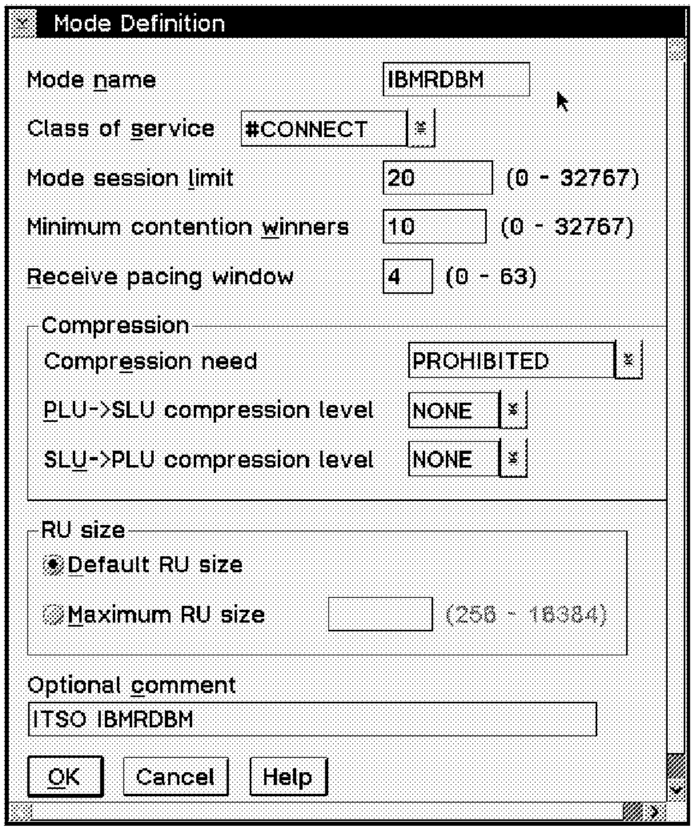


Figure 133. Mode Definition Panel for ITSO

ITSO Mode name......................IBMRDBM
Maximum RU size..............4096

The key fields are:

Mode Name *IBMRDBM* is the name of the IBM supplied mode that contains the session properties for the LU6.2 session. It must match the DLOGMOD value of the IPFSL0E0 definition in the VTAM switched major node definition, refer to Figure 207 on page 195.

Class of service
#CONNECT is for the interactive communications.

Mode Session Limit
The maximum number of sessions that can be active at the same time for the LUs using this mode.

Minimum contention winners
The minimum number of sessions in which you want a logical unit (LU) in this mode to win in a contention with a partner LU.

Receive Pacing Window
The receiving node rate sets the pace for message transmission.

Maximum RU size
We have used default RU size for our messages transferred on sessions using this mode. This number matches the maximum RUSIZES value specified in the VTAM MODENT statement for mode entry #CONNECT.

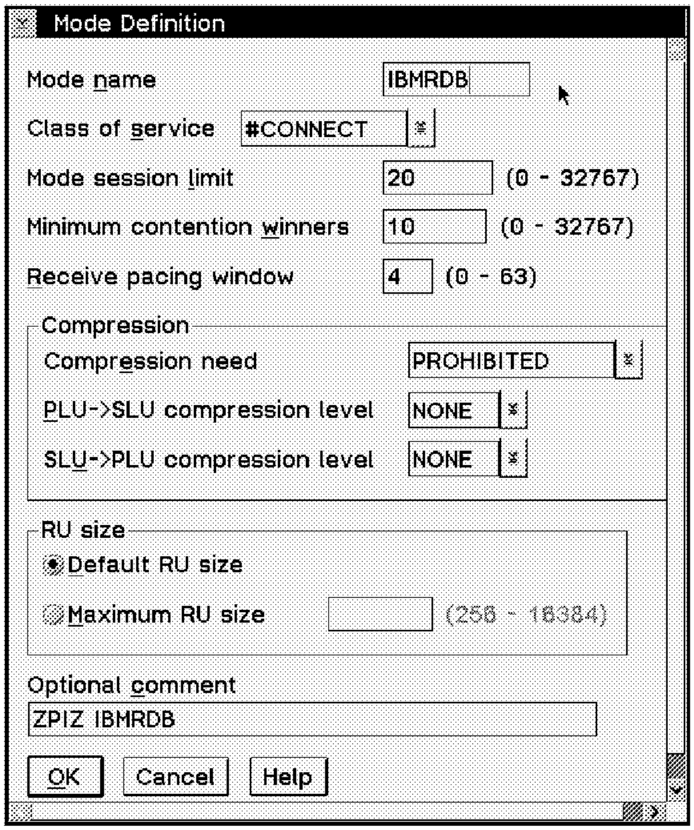


Figure 134. Mode Definition Panel for ZPIZ

ZPIZ Mode name......................IBMRDB
Maximum RU size..............4096

Mode definition - IBMRDB location ZPIZ

Mode Definition IBMRDB was implemented at ZPIZ, so the appropriate mode should be used at CM/2 setup and for CICS RDO.

5.2.9.4 Defining Conversation Security

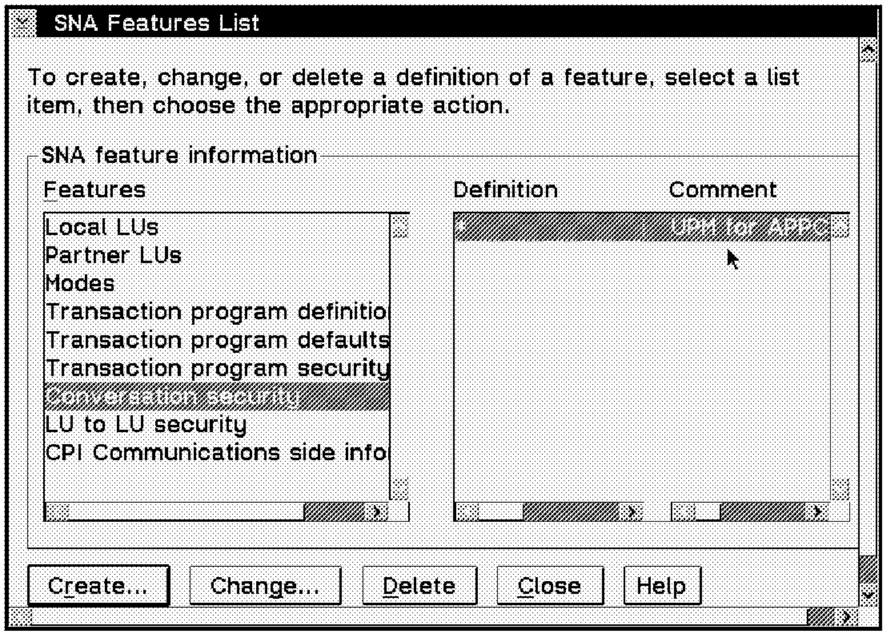


Figure 135. Conversation Security

5.2.9.5 Defining CPI Communications

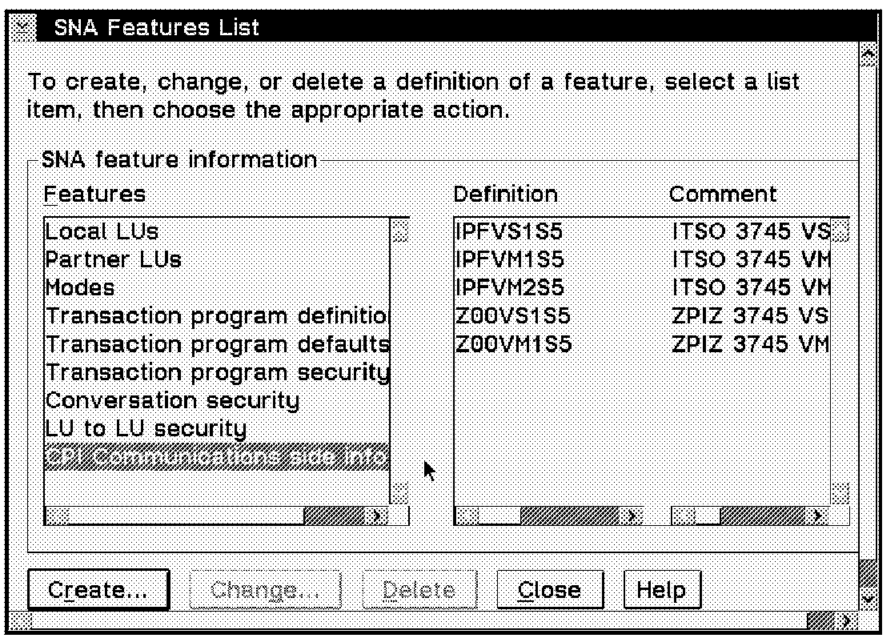


Figure 136. CPI Communication

ZPIZ Symbolic destination name.......VM122AVS
Symbolic destination name.......VSE135B
Symbolic destination name.......BOBAVS
Symbolic destination name.......BOBCICS

On the CPIC Communication Side Information window we provide information about the server we want to define.

Symbolic destination The symbolic destination name defines the name of the CPIC side information. We can choose any name we like. This name will be used when we catalog the node directory for this AS.

Partner LU For Partner LU we can use the fully qualified name or an alias if we want to communicate with a partner LU that has been previously defined.

Partner TP For Partner TP, we have specified the remote TPN. This information is case sensitive. For SQL/DS on VM this value is the name of the SQL server database. For SQL/DS on VSE this value is the name of the CICS APPC transaction.

Security type The security type is optional because the security type specified when we catalog an entry in the node directory overrides this specification.

Mode name For the mode name, we must select the name we have defined for our DRDA connection.

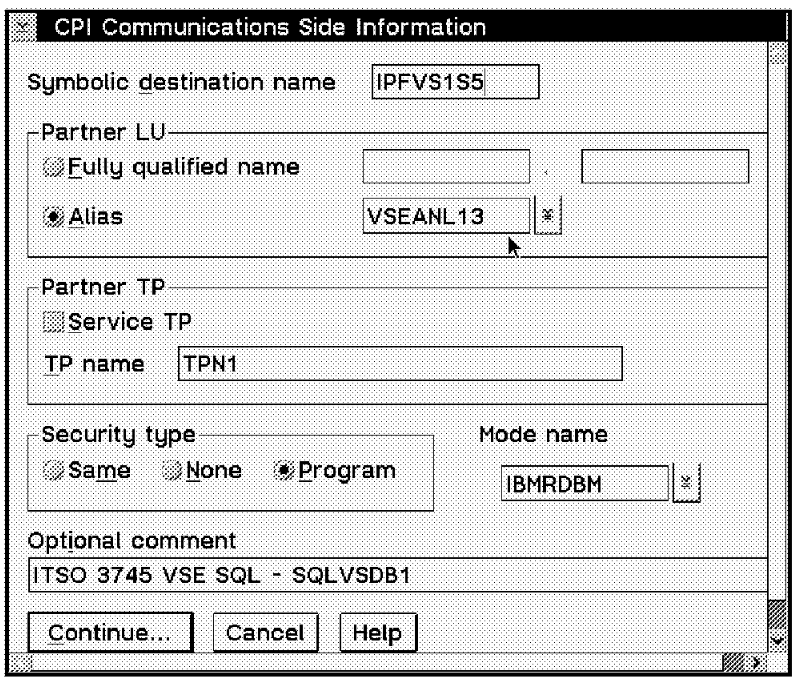


Figure 137. CPI Communication - ITSO VSE SQL/DS(DB - SQLVSDB1)

ZPIZ Symbolic destination name.......VSE135B
Alias...........................VSE135
TPN.............................TPN1
Mode name.......................IBMRDB

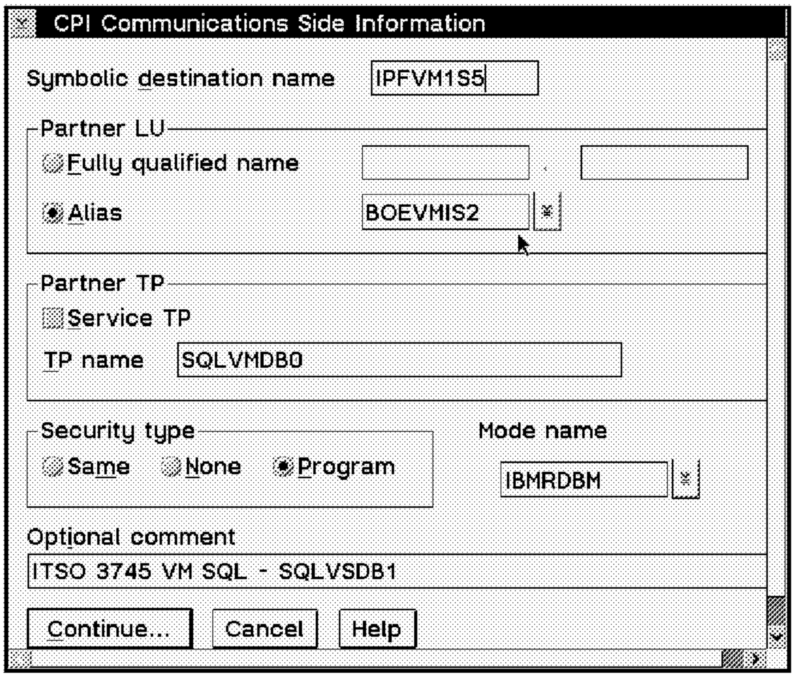


Figure 138. CPI Communication - ITSO VM SQL/DS(DB - SQLVMDB0)

ZPIZ Symbolic destination name.......VM122AVS
Alias..........................VM122AVS
TPN............................SQLDBA
Mode name......................IBMRDB

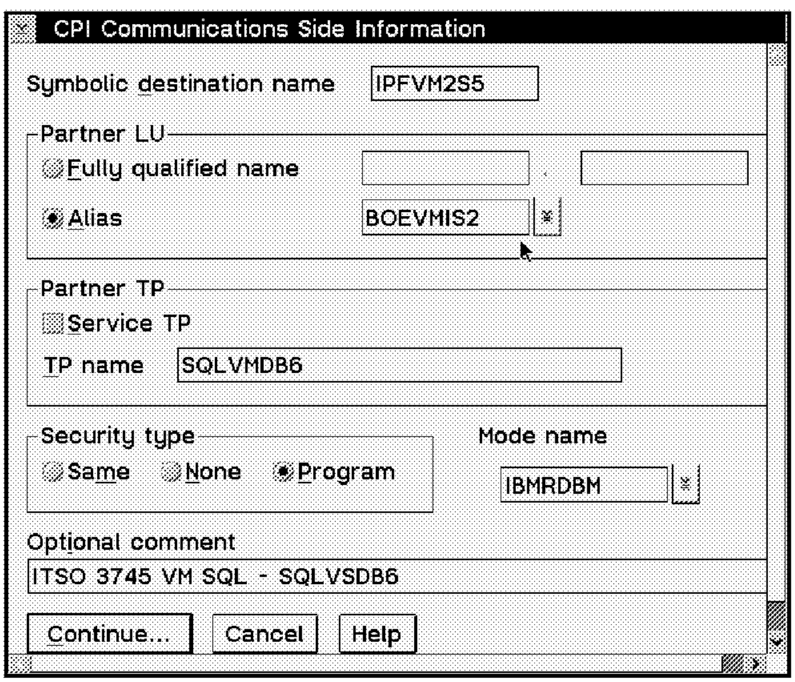


Figure 139. CPI Communication - ITSO VM SQL/DS(DB - SQLVMDB6)

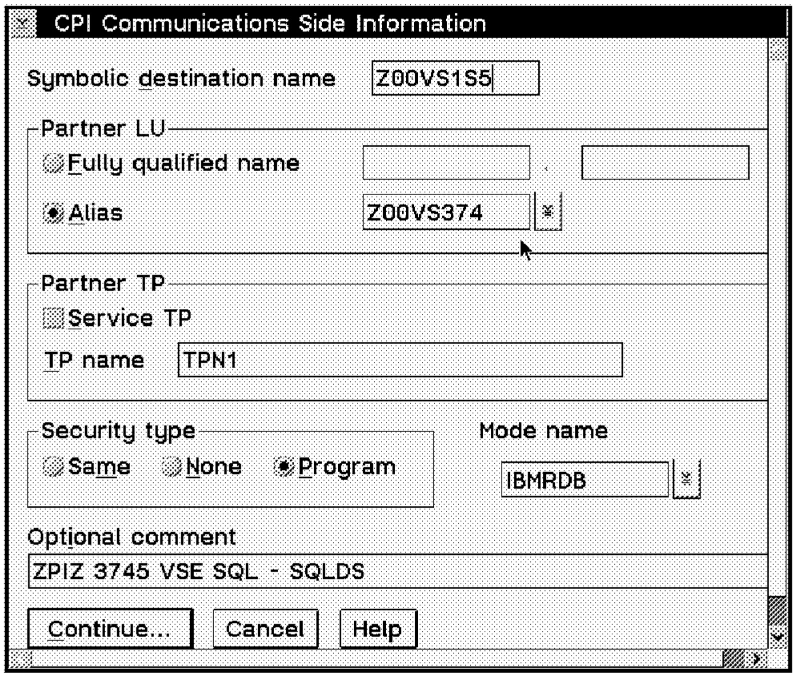


Figure 140. CPI Communication - ZPIZ VSE SQL/DS(DB - SQLDS)

ZPIZ Symbolic destination name.......BOBCICS
Alias...........................BOBCICS
TPN............................TPN1
Mode name.......................IBMRDB

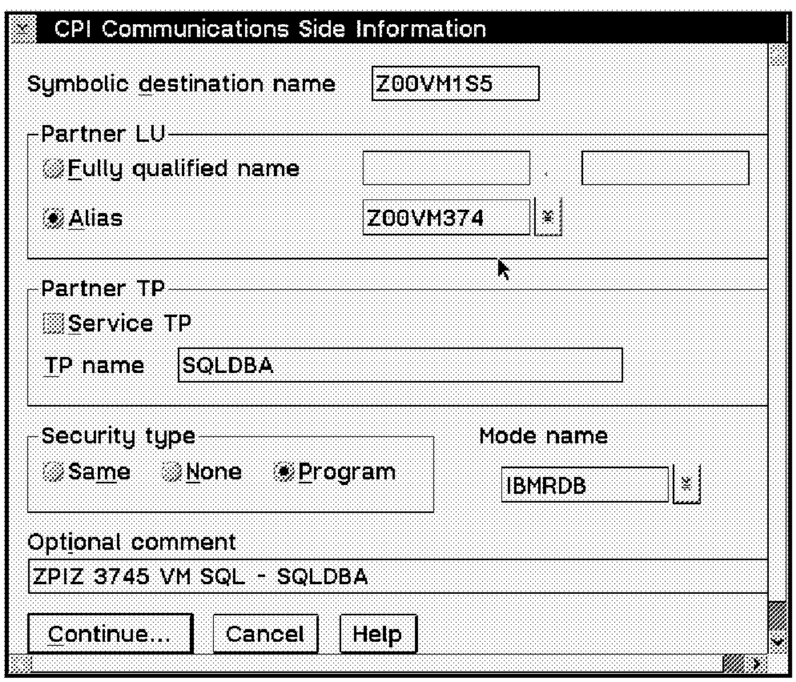


Figure 141. CPI Communication - ZPIZ VM SQL/DS(DB - SQLDBA)

ZPIZ Symbolic destination name.......BOBAVS
Alias..........................BOBAVS
TPN............................SQLVMDB0
Mode name.......................IBMRDBM

5.2.10 DB2/2 and DDCS/2 Customization

To connect to a DRDA database server, DDCS/2 requires information in the database directories. We must specify entries in the following directories to enable the DDCS workstation to connect to a DRDA AS:

- Database connection services (DCS) directory
- Node directory
- System database directory

Table 10. Database Directory Entries for ITSO and ZPIZ Environment via 3745

	ITSO Böblingen		**ZPIZ Ljubljana**	
	SQL/DS VSE	SQL/DS VM	SQL/DS VSE	SQL/DS VM
Database Directory				
Database alias	IPFVS1A5	IPFVM1A5	Z00VS1A5	Z00VM1A5
Database name	IPFVS1D5	IPFVM1D5	Z00VS1D5	Z00VM1D5
Node name	IPFVS1N5	IPFVM1N5	Z00VS1N5	Z00VM1N5
Authentication	DCS	DCS	DCS	DCS
DCS Database Directory				
local Database	IPFVS1D5	IPFVM1D5	Z00VS1D5	Z00VM1D5
Target Database	SQLVSDB1	SQLVMDB0	SQLDS	SQLDBA
Node Directory				
Node name	IPFVS1N5	IPFVM1N5	Z00VS1N5	Z00VM1N5
Protocol	APPC	APPC	APPC	APPC
Symb.dest.name	IPFVS1S5	IPFVM1S5	Z00VS1S5	Z00VM1S5
Security	Program	Program	Program	Program

Note:
1. IPF - Böblingen,Germany. Z00 for Ljubljana, Slovenia.
2. VM - for VM, VS for VSE.
3. 5 - 3745 communication controller was used.
4. S - Symbolic Destination Name (DCS).
5. N - Node Name.
6. D - Database Name.

5.2.10.1 Cataloging in the DCS Directory

The DCS directory contains an entry for each DRDA AS that can be accessed from the workstation. When the application issues a CONNECT statement specifying a remote database using APPC protocol, the database manager searches the DCB directory for a matching database name entry. If it finds a matching entry, it uses the DRDA protocol for the connection. If it does not find a matching entry, it uses the private protocol. To catalog a DCS database do the following:

Click on the Database connection services directory icon on the Database Director - Tree view window. The Database Connection Services Directory - Details View is displayed, listing the DCS databases for this instance.

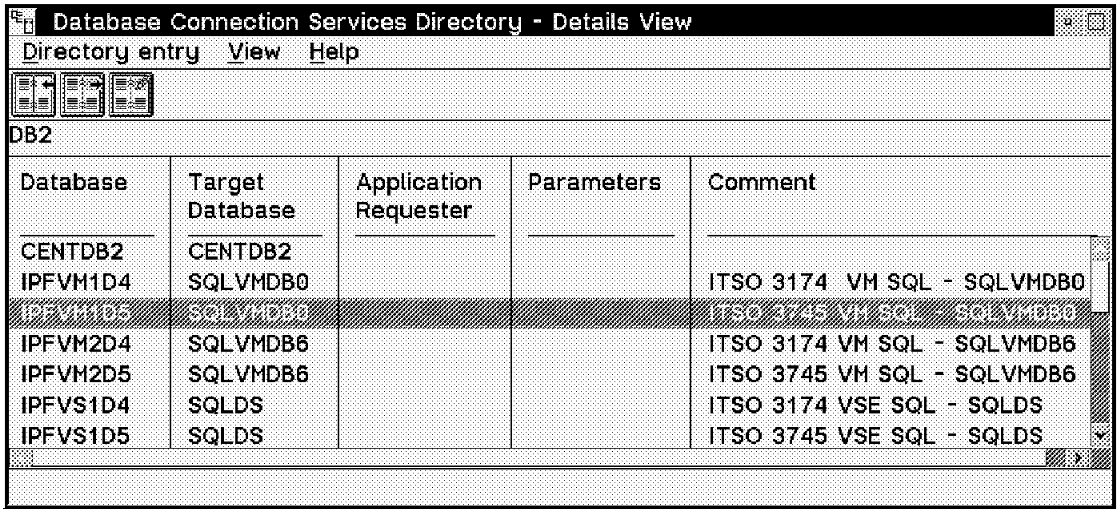


Figure 142. Database Connection Services 1 of 2

ITSO target database name SQLVMDB0
database alias IPFVM1D5

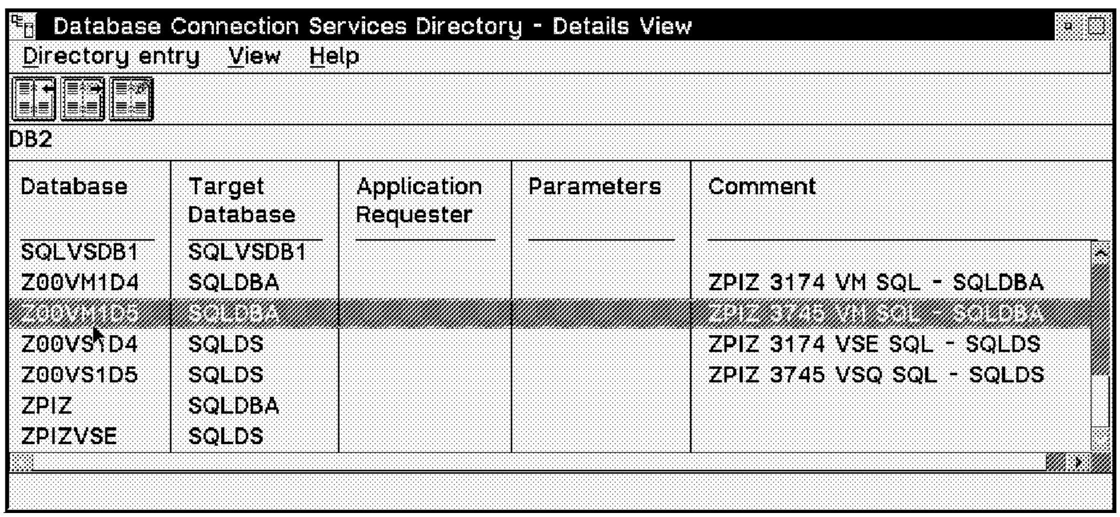


Figure 143. Database Connection Services 2 of 2

ZPIZ target database name ....SQLDBA
database alias Z00VM1D5

5.2.10.2 Cataloging in the Node Directory

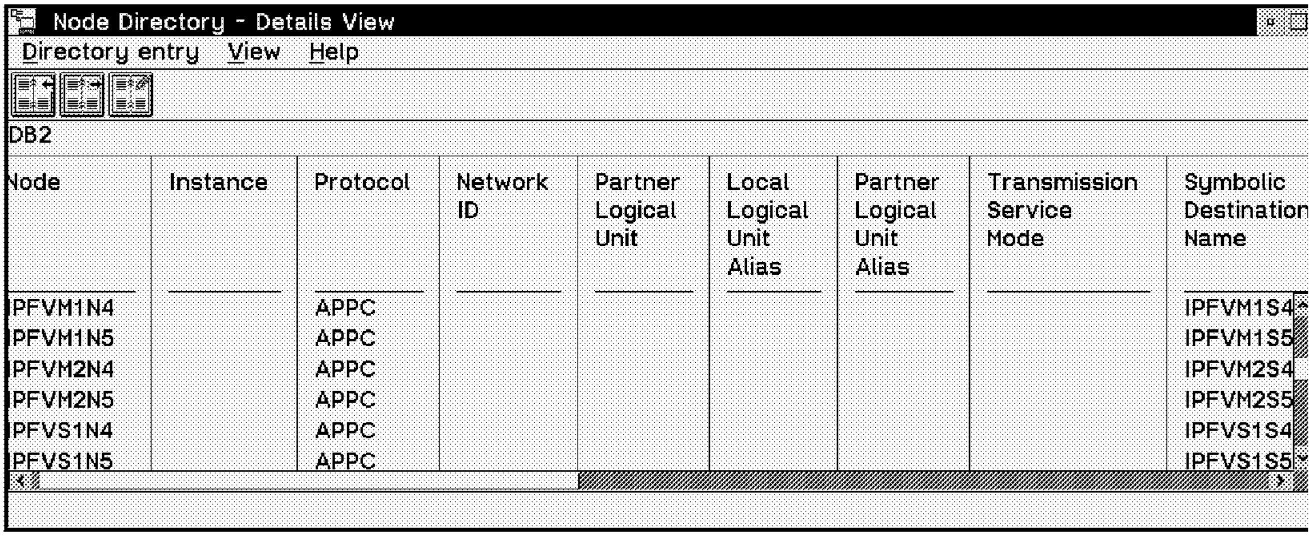


Figure 144. Node Directory

ITSO Node (WS name)...................IPFVM1N5
Protocol.........................APPC
Symbolic Destination Name.........IPFVM1S5
(will be used in CM/2 CPI communication definition)

5.2.10.3 Cataloging in the System Database Directory

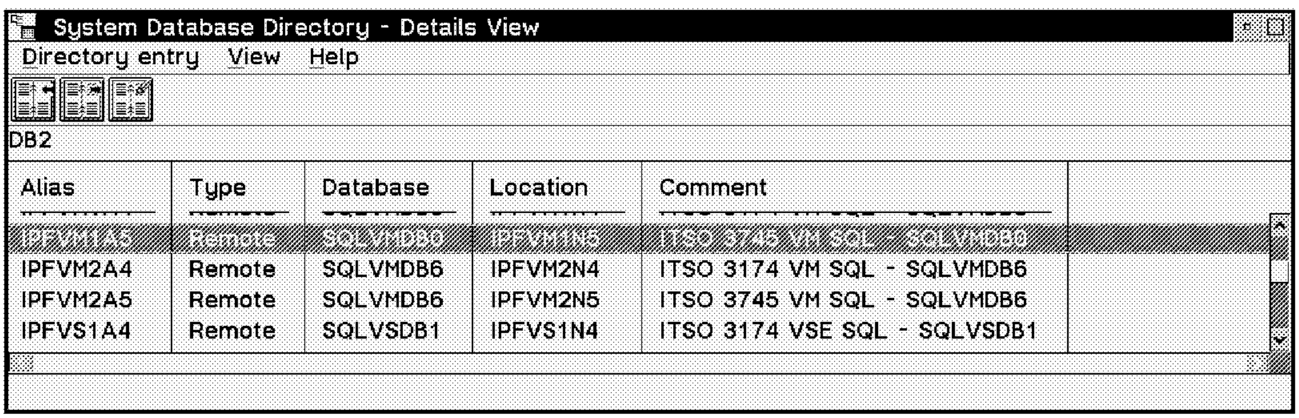


Figure 145. System Database Directory 1 of 2

ITSO Location.........IPFVM1N5
This value should match the node name in the node directory.

Database.........SQLVMDB0
This value should match the target database name or database name in the database connection services directory.

Alias...........IPFVM1A5
This value will be used to establish the connection with the database in the **connect to** command.

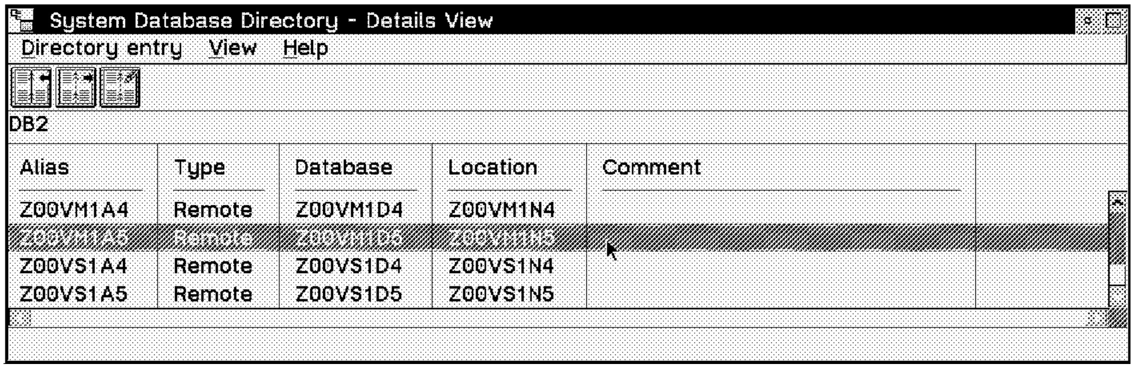


Figure 146. System Database Directory 2 of 2

ZPIZ

Location.........Z00VM1N5
This value should match the node name in the node directory.

Database.........Z00VM1D5
This value should match the target database name or database name in the database connection services directory.

Alias............Z00VM1A5
This value will be used to establish the connection with the database in the **connect to** command.

5.2.11 SNA Subsystem

The following pages show the information about logical link and LU6.2 sessions. You can take the following steps to find such information:

1. Open the CM/2 folder
2. Click on the Subsystem Management
3. Select SNA Subsystem

You will then find the menu containing all the resources.

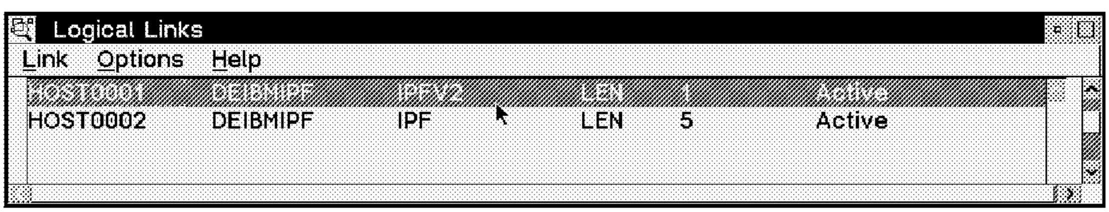


Figure 147. Logical Link

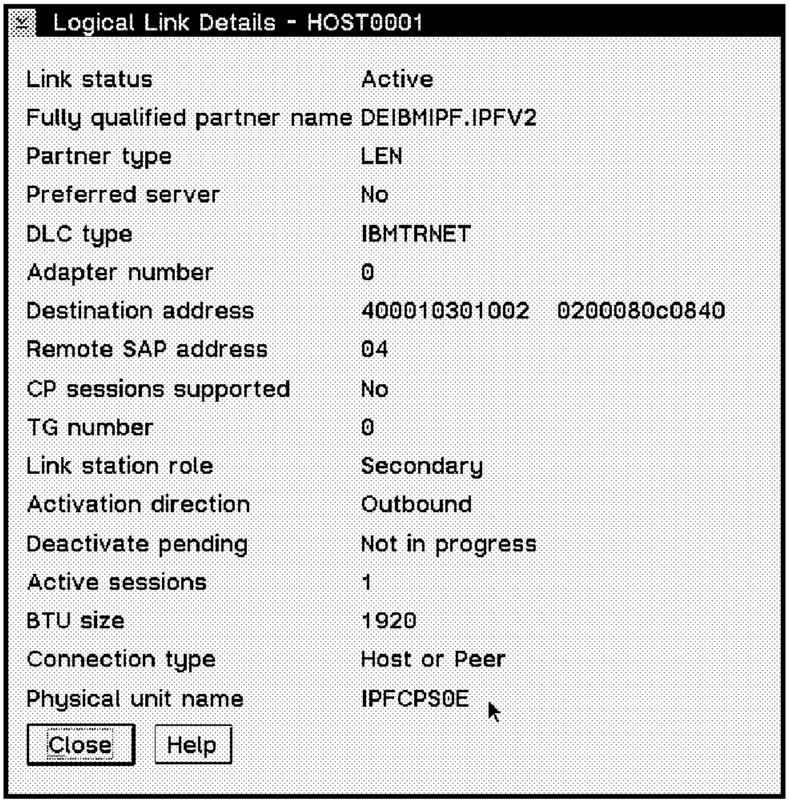


Figure 148. Logical Link Details

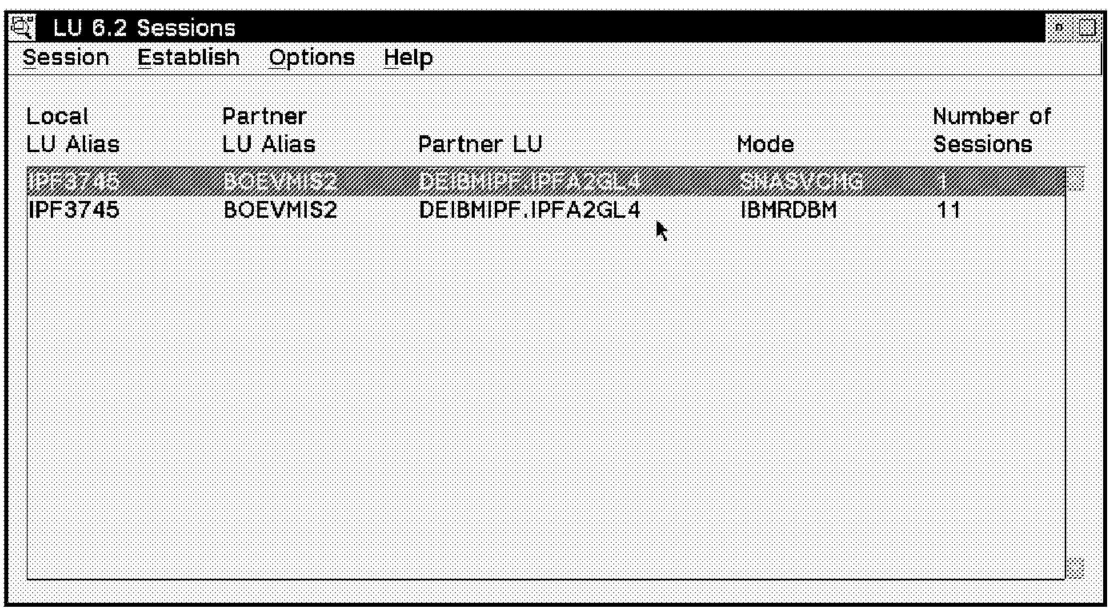


Figure 149. LU 6.2 Session (BOEVMIS2)

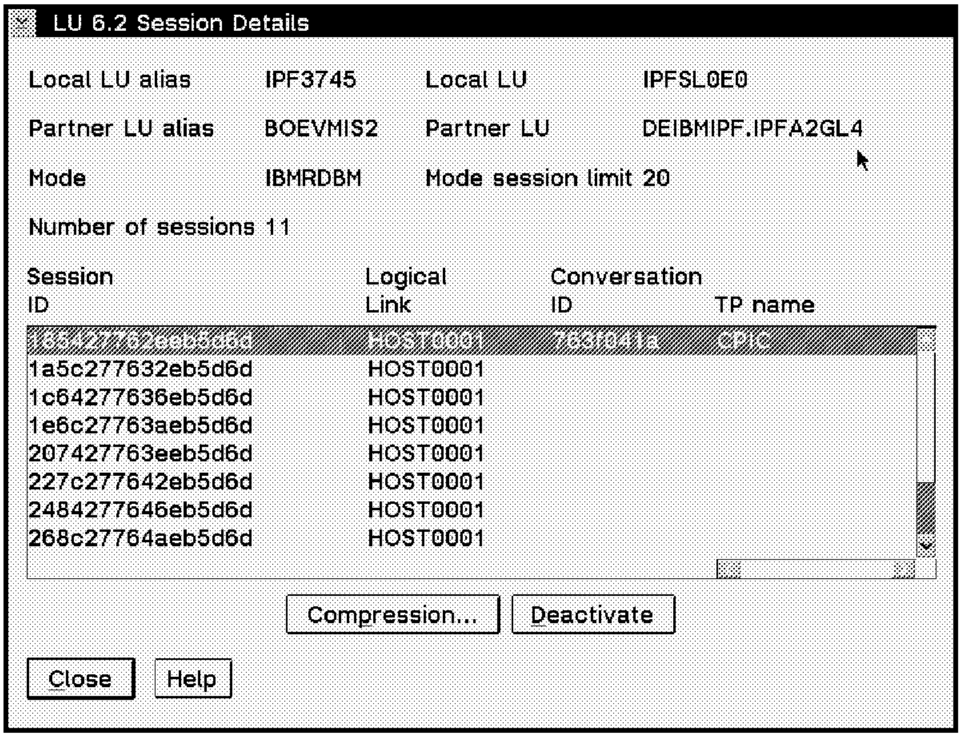


Figure 150. LU 6.2 Session Details (BOEVMIS2)

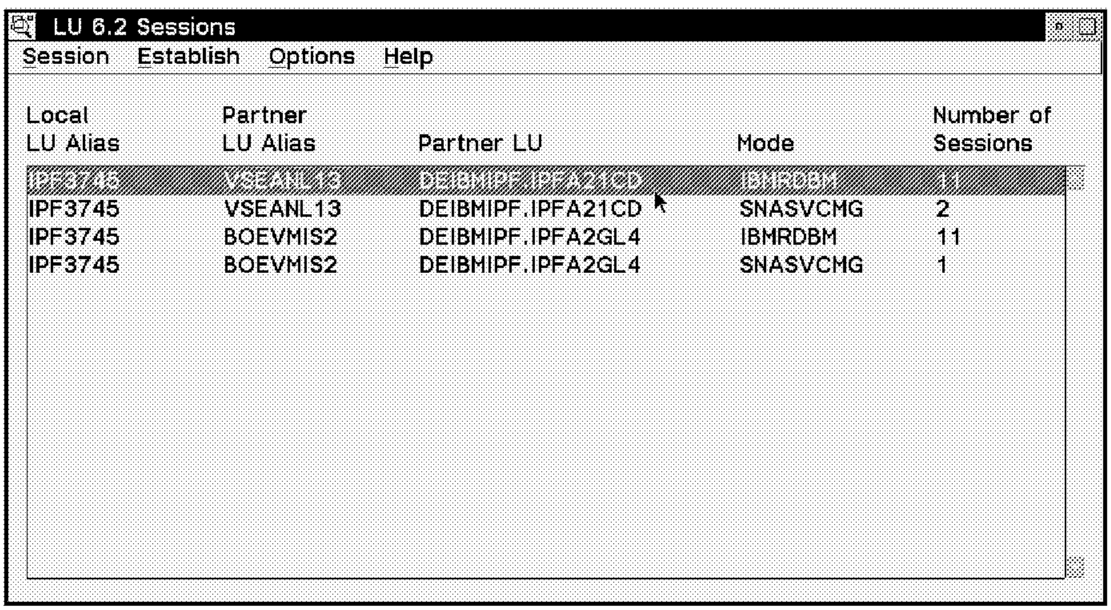


Figure 151. LU 6.2 Sessions (VSEANL13)

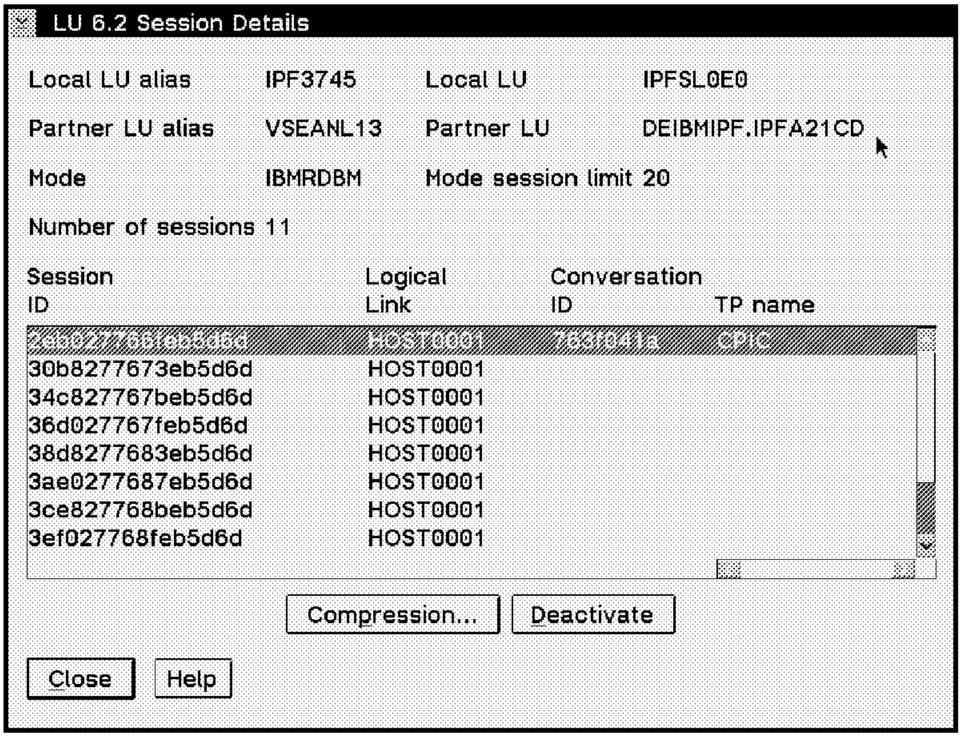


Figure 152. LU 6.2 Session Details (VSEANL13)

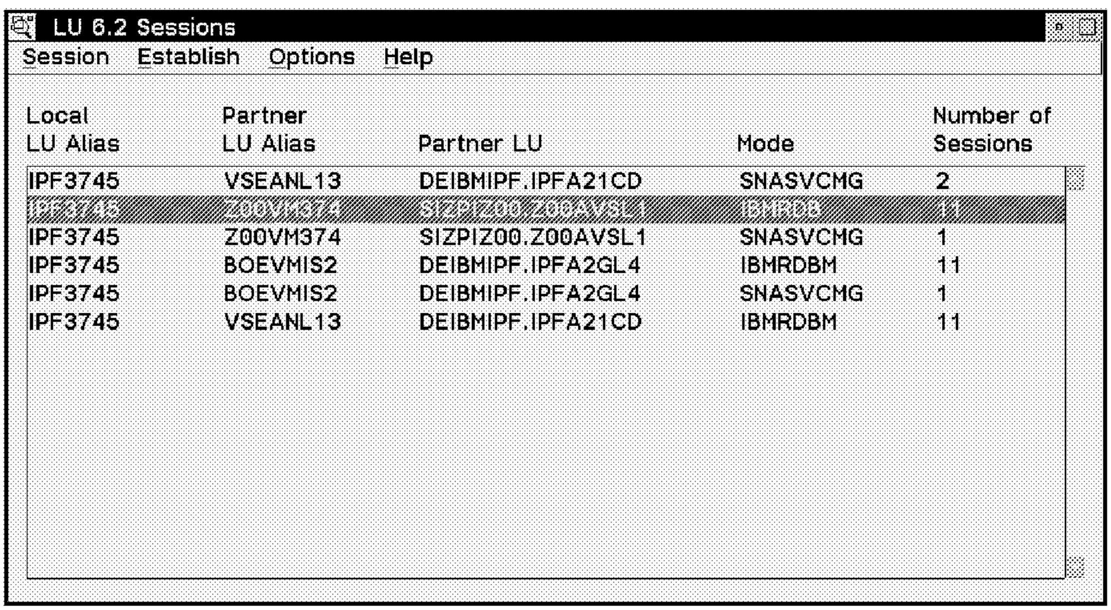


Figure 153. LU 6.2 Sessions (Z00VM374)

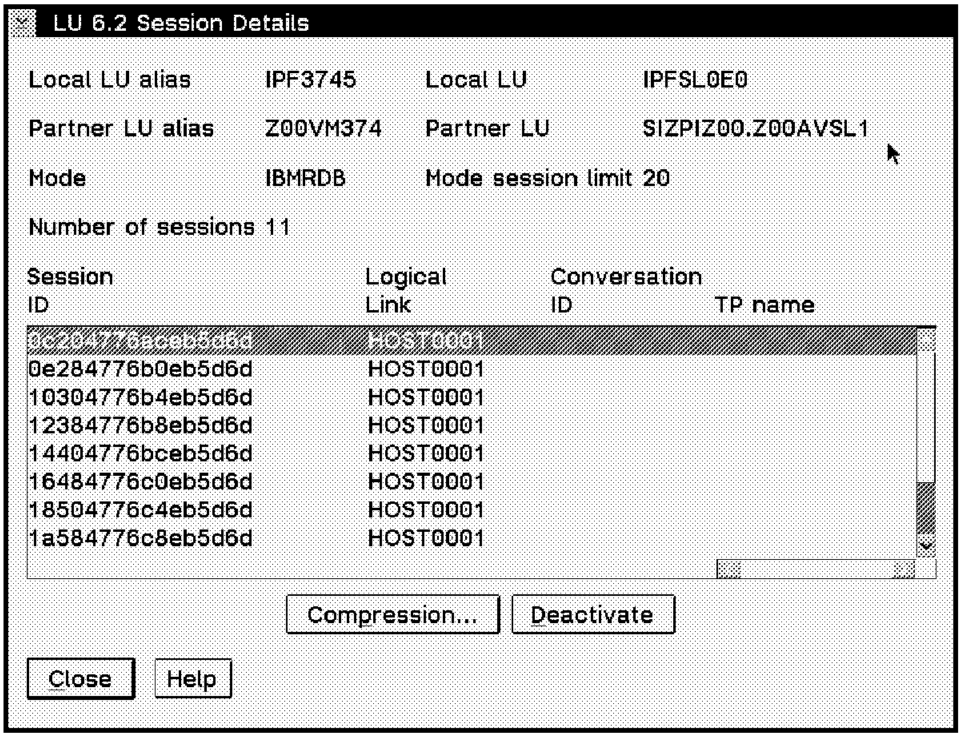


Figure 154. LU 6.2 Session Details (Z00VM374)

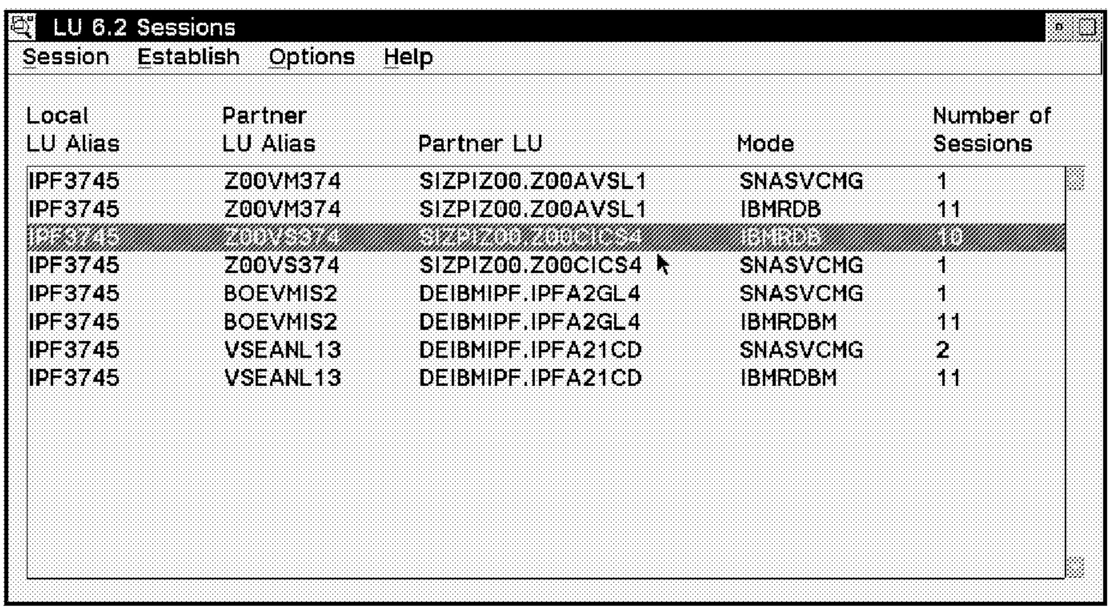


Figure 155. LU 6.2 Sessions (Z00VS374)

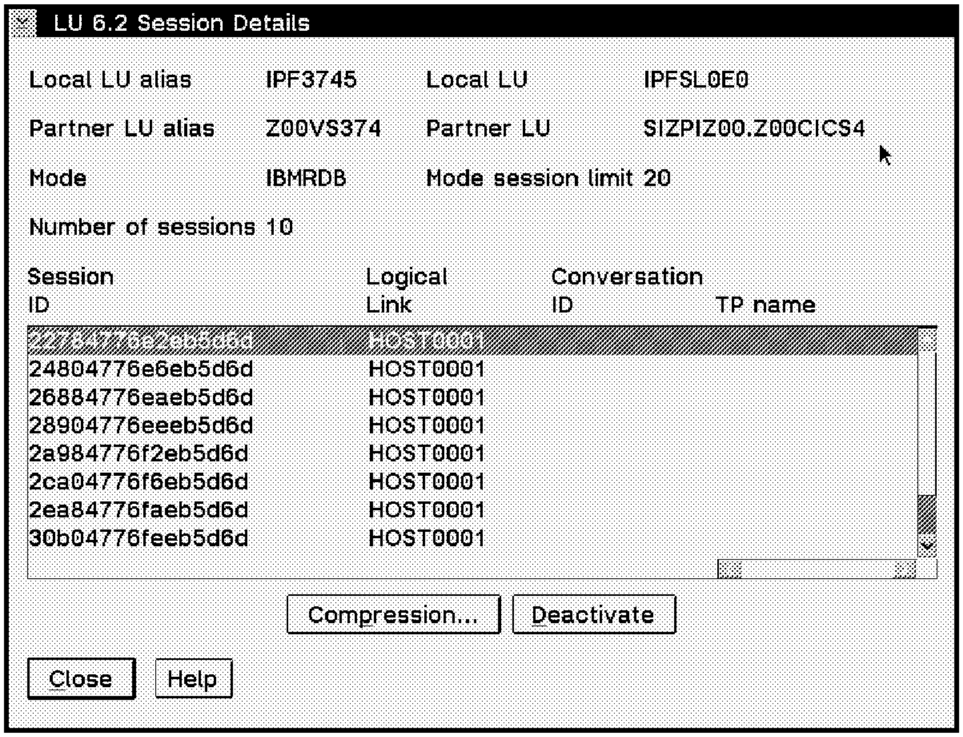


Figure 156. LU 6.2 Session Details (Z00VS374)

5.3 Connectivity via 3174 ITSO to 3745 ZPIZ

5.3.1 OS/2 Setting for 3174 Cluster Controller

The LU IPFCL0E0 is recognized by VM/VTAM as an independent LU via 3174, using a dynamic LU definition with cross-domain resource (CDRSC). The CDRSC major node is IPF2CDRS which is predefined. An adjacent link station is IPFCP200. Dynamic LU definition of independent LUs should be done by specifying XID=YES,DYNLU=YES in the PU statement of the gateway controller only. These operands should not be specified for the other PU statements. Then an LU definition is not required in the local SNA major nodes. Dynamic LU definition of independent LUs can be done in the cross domain resource major node also. In cross domain resources, the adjacent link station should be defined as ALSLIST=(IPFCP200). For connection to CICS/ESA of VSE/ESA IPFA21CD should be defined in this major node. Refer to Figure 204 on page 193 and Figure 203 on page 193.

```
IPFCL0E0  -  Dynamic LU independent definition
IPF2CDRS  -  CDRSC major node
IPFCP200  -  Adjacent link station
IPFCP20E  -  PU
```

This part is very similar to the previous one, covering connectivity via 3745. Nevertheless, some panels which might be of interest when implementing 3174 connectivity are given.

- OS/2 Communication Manager Customization
- DB2/2 and DDCS/2 Customization

5.3.2 OS/2 Communication Manager Customization

Table 11. SNA Customization for ITSO and ZPIZ for DRDA Environment via 3174

	ITSO Böblingen		**ZPIZ Ljubljana**	
	SQL/DS VSE	SQL/DS VM	SQL/DS VSE	SQL/DS VM
Network id	DEIBMIPF	DEIBMIPF	SIZPIZ00	SIZPIZ00
Controller Address	400020201001		400010001088	
IDBLK and IDNUM	05D E000E		05D 00001	
Local Node Name	IPFCP20E	IPFCP20E	A01P01	A01P01
Partner Node Name (SSCP)	IPFV2A	IPFV2	A05M	SIZPIZ6
Fully Qualified Partner LU Name	DEIBMIPF. IPFA21CD	DEIBMIPF. IPFA2GL4	SIZPIZ00. Z00CICS4	SIZPIZ00. Z00AVSL1
Partner LU Alias	VSEANL13	BOEVMIS2	Z00VS317	Z00VM317
Local LU Name	IPFCL0E0	IPFCL0E0	Z00T010A	Z00T010A
Local LU Alias	IPF3174	IPF3174	Z00T01	Z00T01
Mode Name	IBMRDBM	IBMRDBM	IBMRDB	IBMRDB
Transaction Program Name	TPN1	SQLVMDB0	TPN1	SQLDBA
Security	PGM	PGM	PGM	PGM
Communication Type	APPC	APPC	APPC	APPC
CPI Communication	IPFVS1S4	IPFVM1S4	Z00VS1S4	Z00VM1S4

Note:
1. IPF stands for Böblingen,Germany. Z00 for Ljubljana, Slovenia
2. VM stands for VM. VS stands for VSE.
3. 4 as the last character means that the 3174 communication controller was used.
4. S stands for symbolic destination name
5. TPN1 (VSE - ceda i tran(tpn1). SQLVMDB0 (VM - IUCV res. id.)
6. PU ipfcp20e (d net,id=ipfcl0e0..... alslist = ipfcp20e)

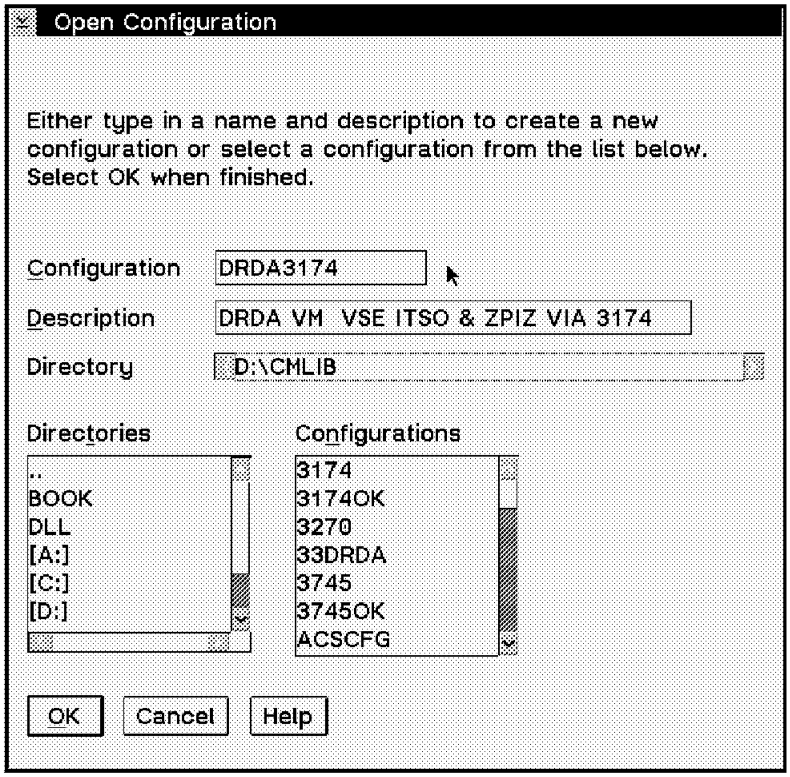


Figure 157. DRDA 3174 Configuration

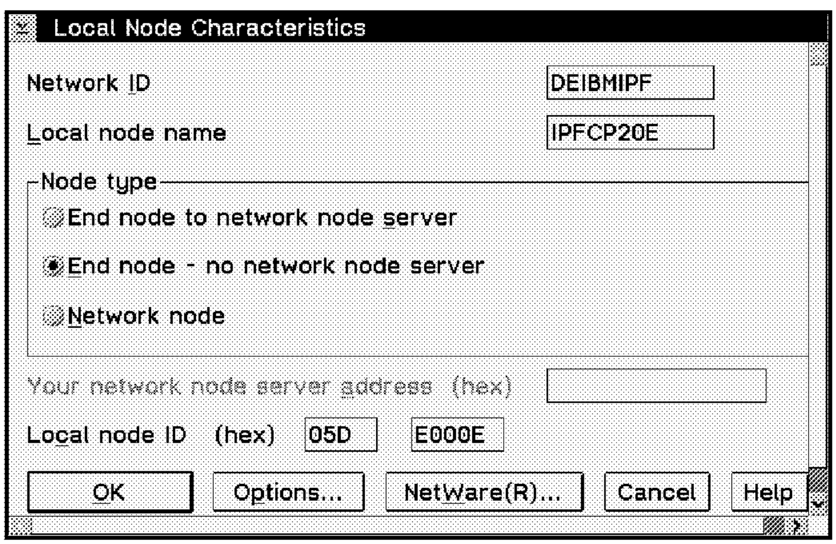


Figure 158. Local Node Characteristics

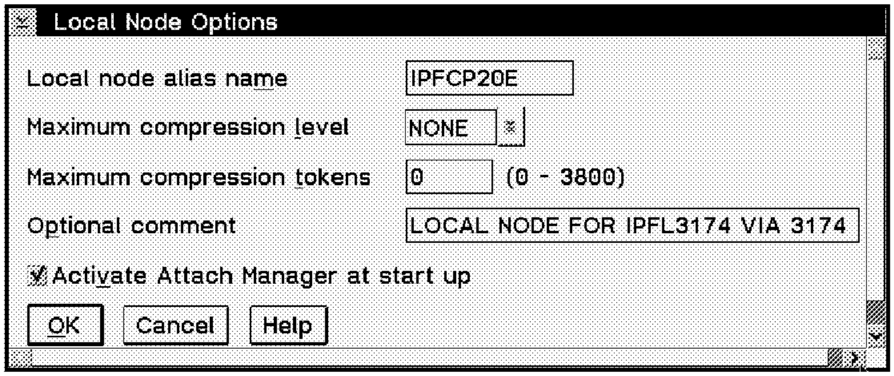


Figure 159. Local Node Options

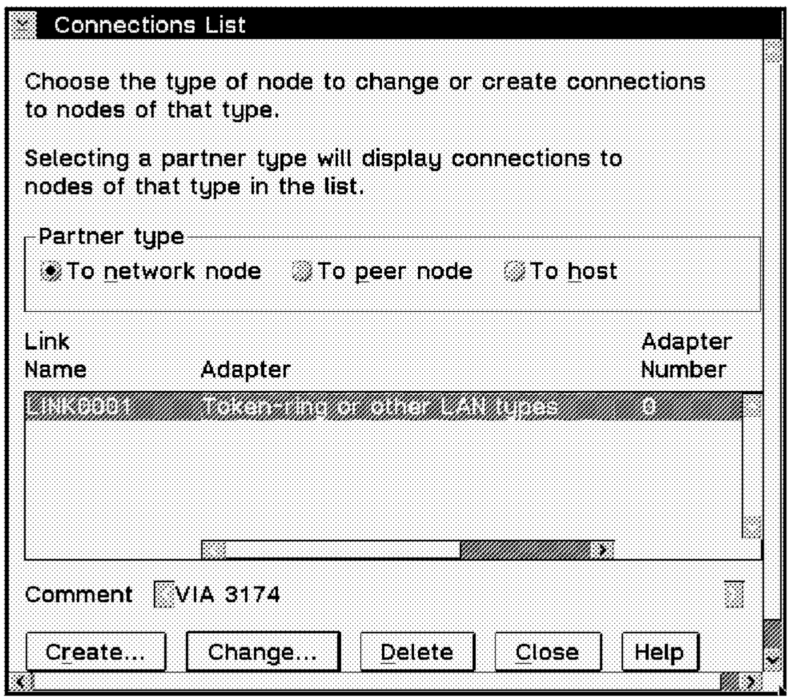


Figure 160. Connection Definition Panel 1 of 2

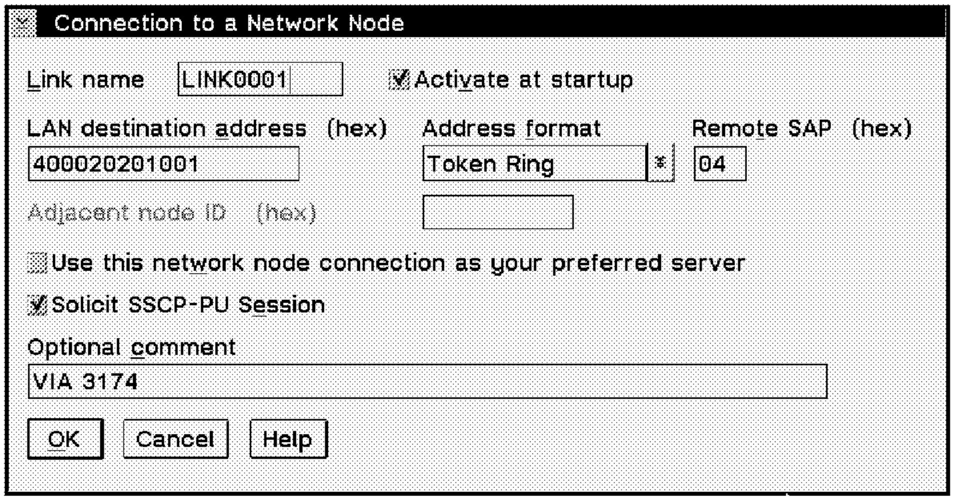


Figure 161. Connection Definition Panel 2 of 2

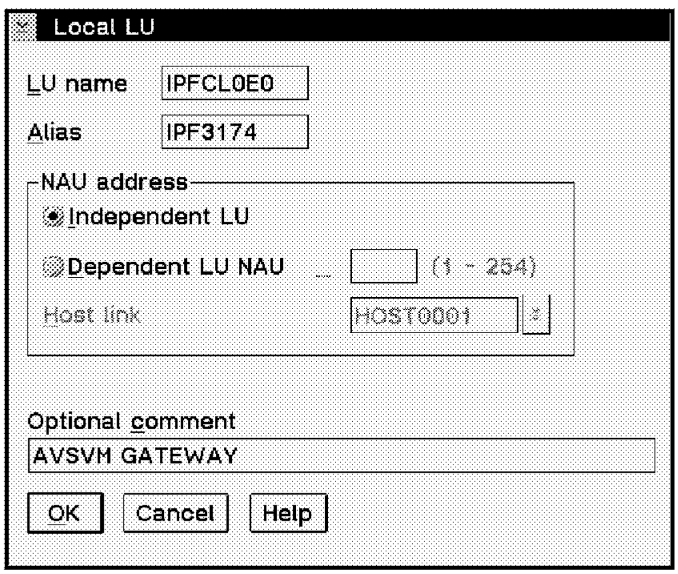


Figure 162. Local LU

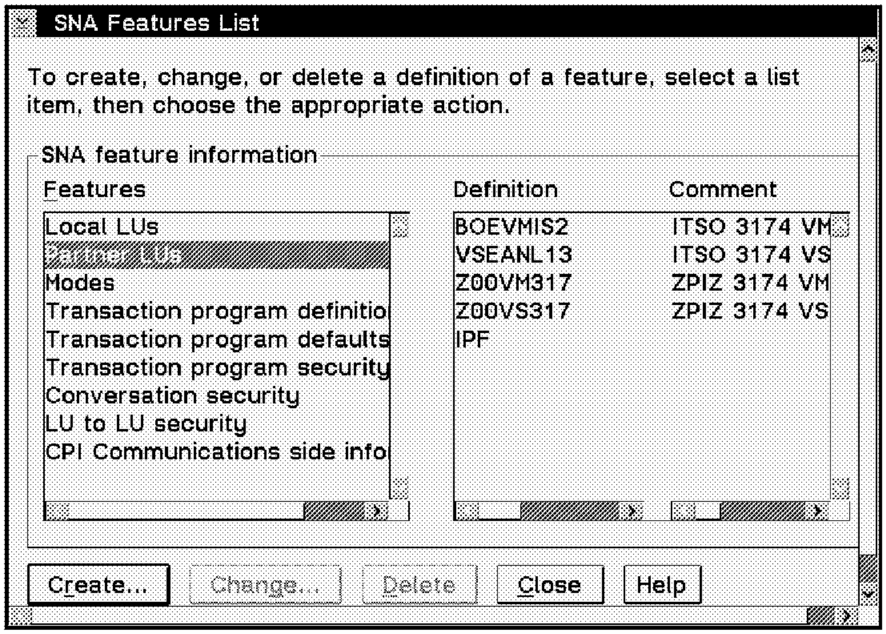


Figure 163. SNA Features List

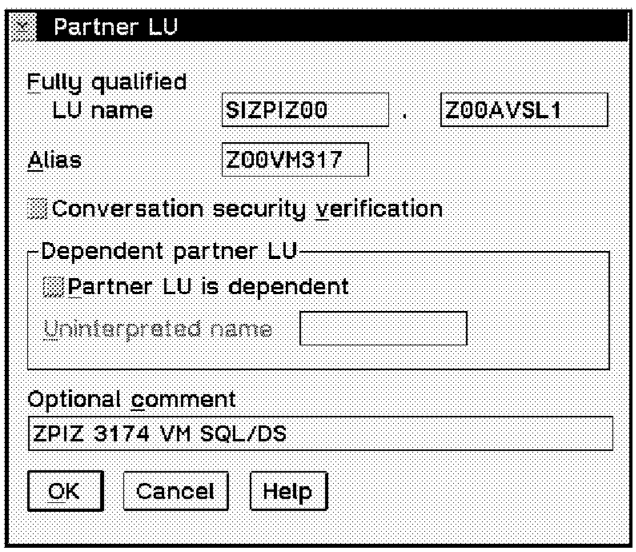


Figure 164. Partner LU - ZPIZ VM SQL/DS

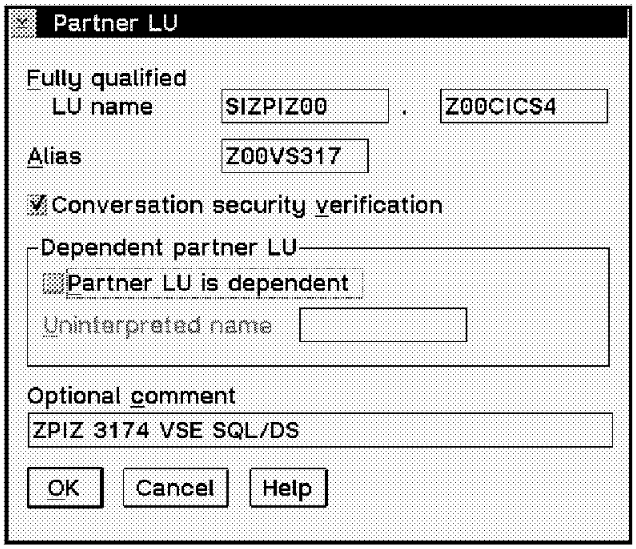


Figure 165. Partner LU - ZPIZ VSE SQL/DS

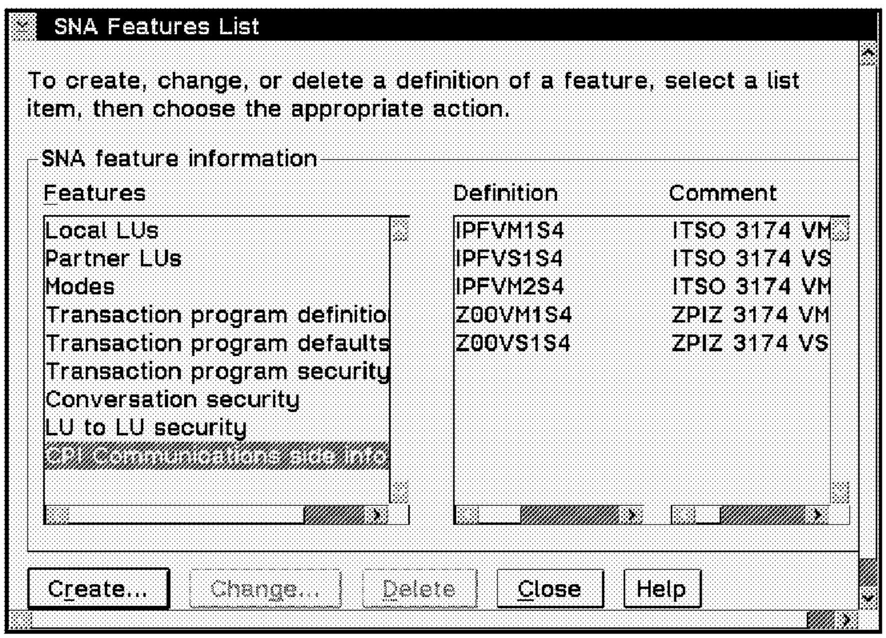


Figure 166. CPI Communications

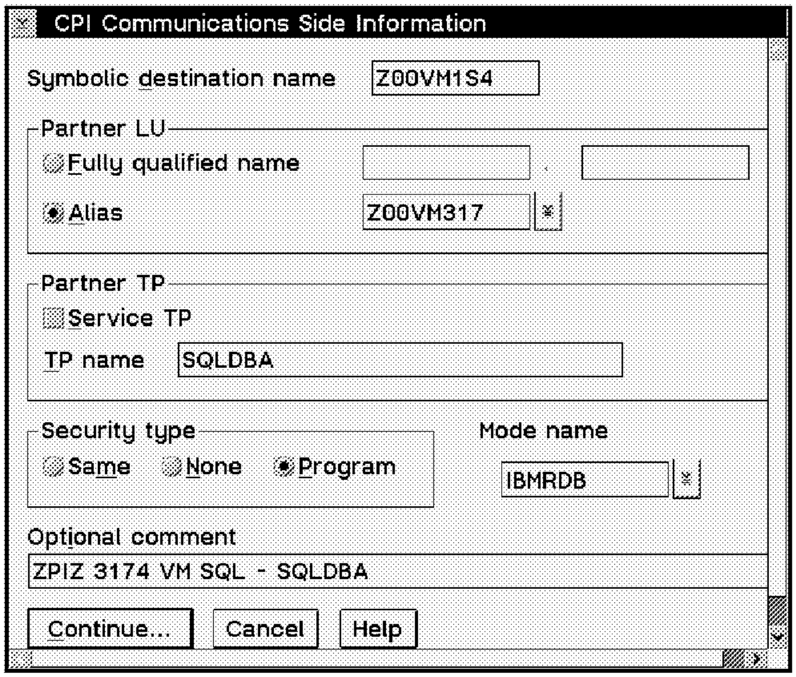


Figure 167. CPI - ZPIZ VM SQL/DS

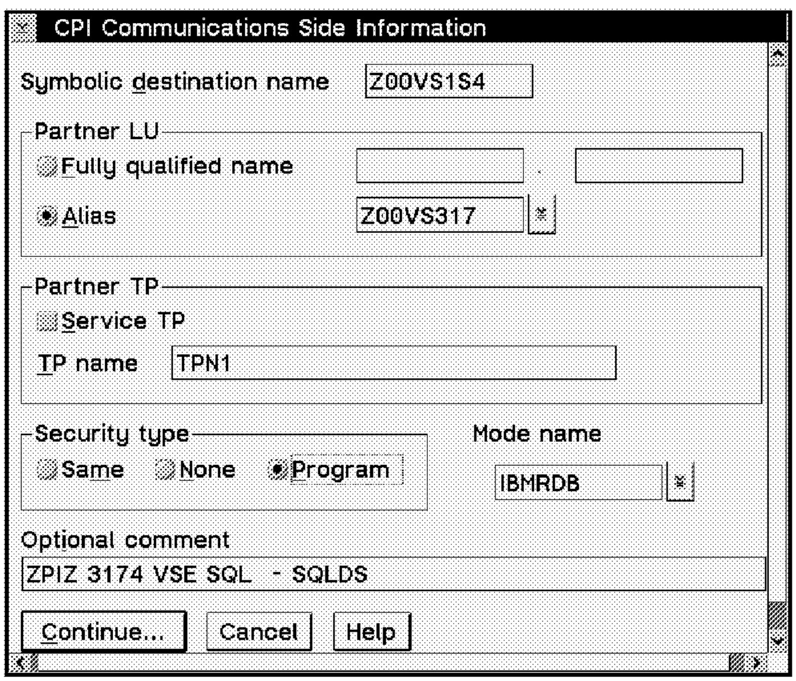


Figure 168. CPI - ZPIZ VSE SQL/DS

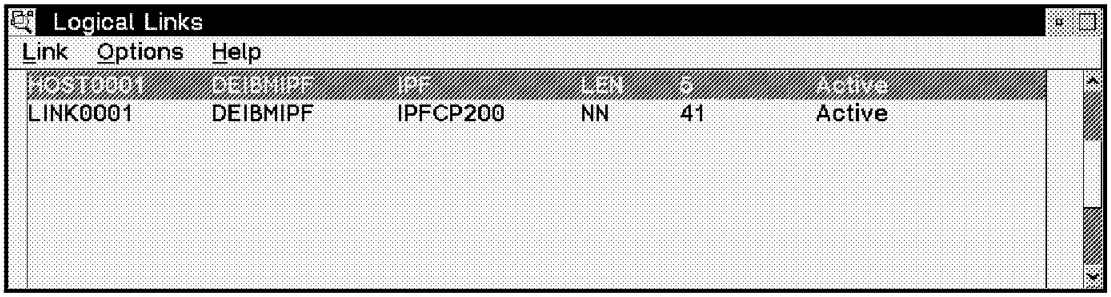


Figure 169. SNA Management - Logical Links

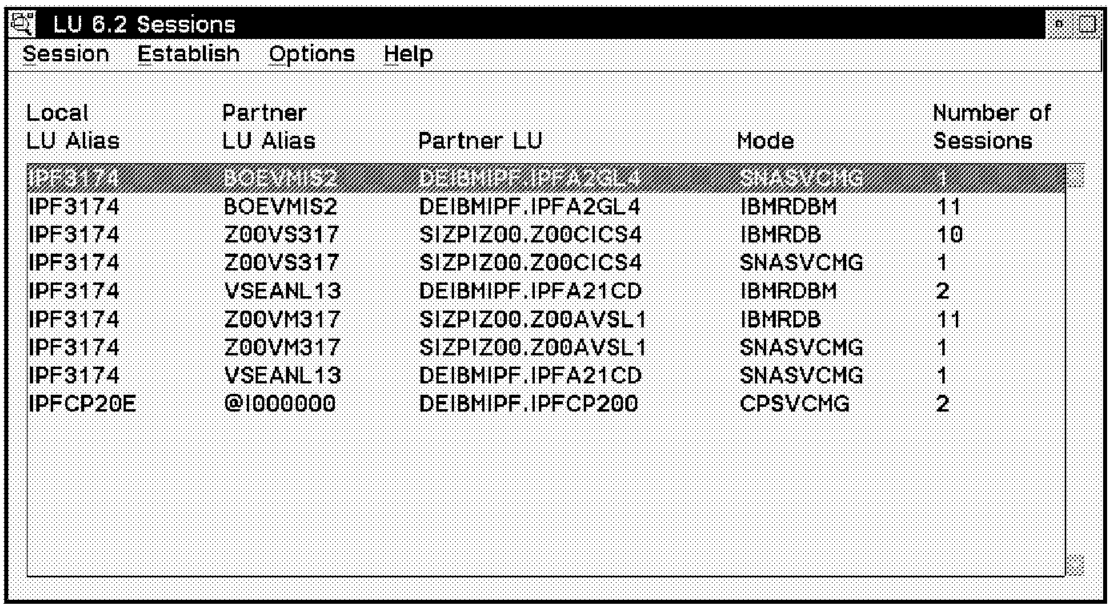


Figure 170. SNA Management - LU 6.2 Sessions

5.3.3 DB2/2 and DDCS/2 Customization via 3174 Control Unit

To create DB2/2 and DDCS/2 directories, open DIRECTORY TOOL or use DB2/2 Command Line. DDCS/2 allows applications running on the workstation to access host data. Customization is very similar to that described in the previous section. Below are panels of the DB2/2 directory from the ZPIZ side.

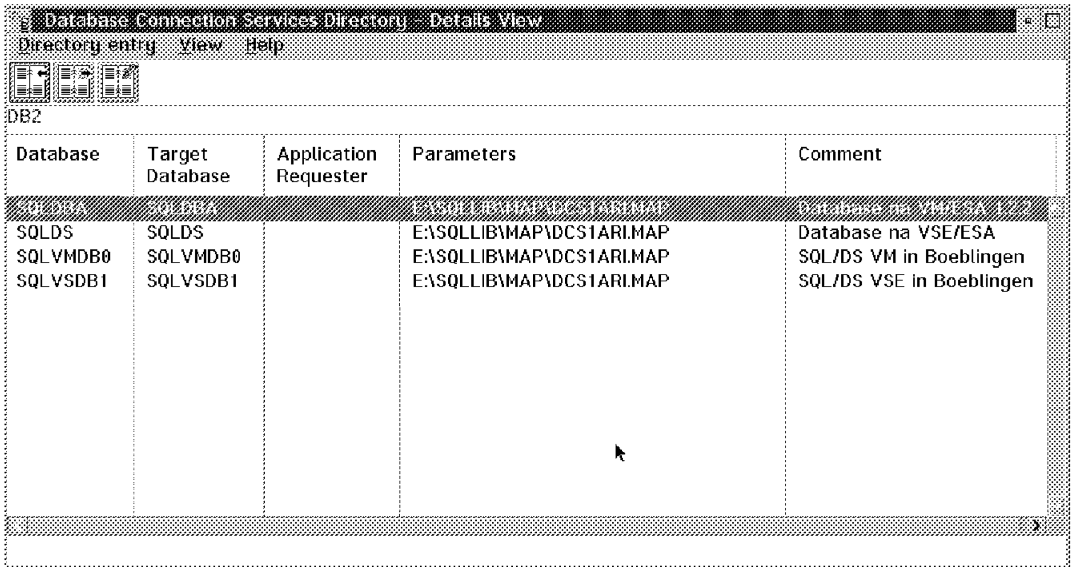


Figure 171. Database Connection Services Directory - ZPIZ

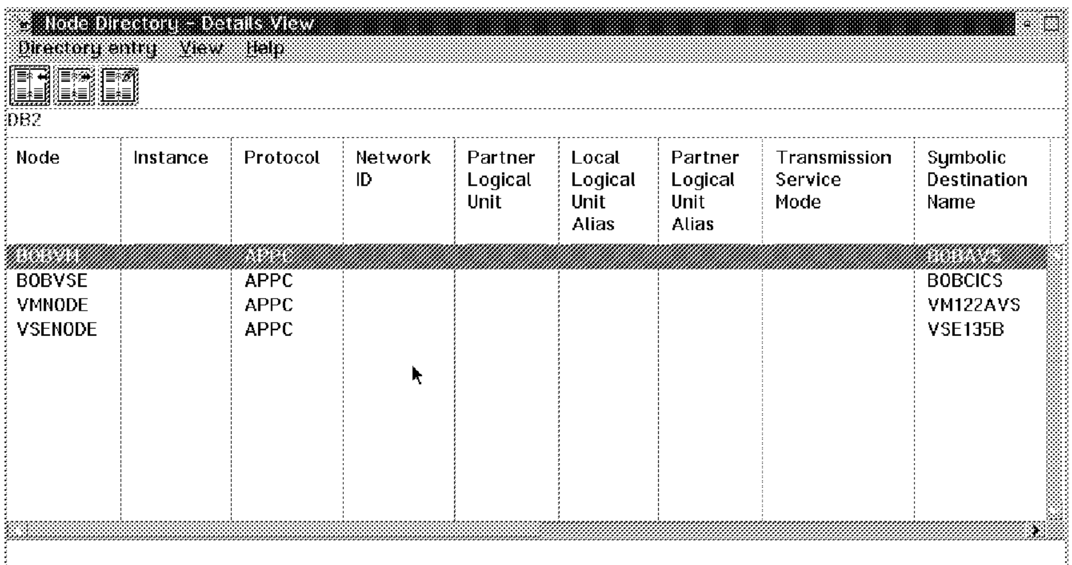


Figure 172. Node Directory - ZPIZ

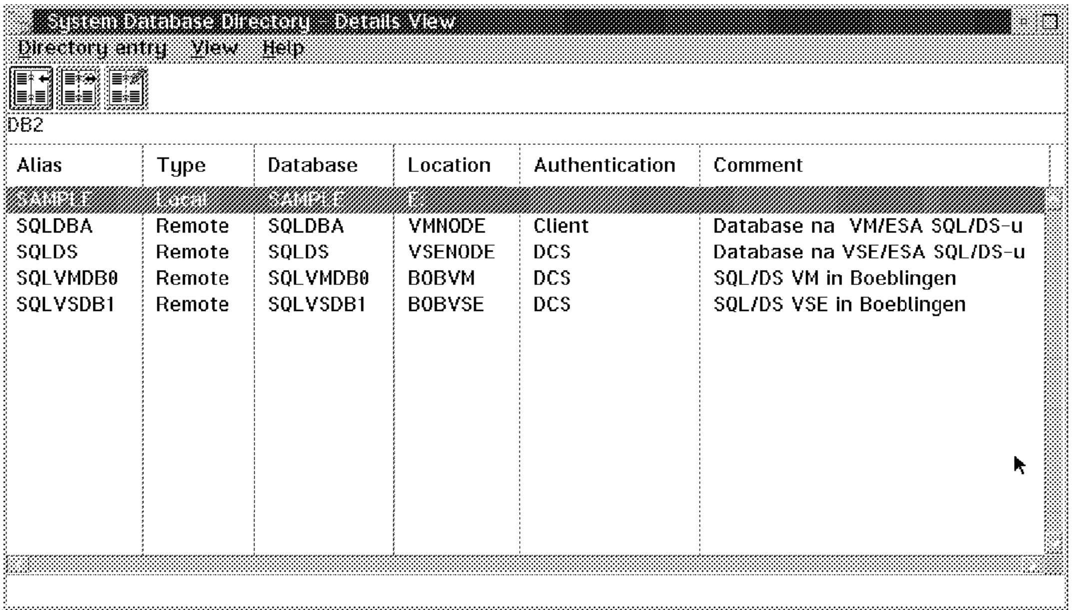


Figure 173. System Database Directory - ZPIZ

Table 12. Database Directory Entries for ITSO and ZPIZ Environment via 3174

	ITSO Böblingen		**ZPIZ Ljubljana**	
	SQL/DS VSE	SQL/DS VM	SQL/DS VSE	SQL/DS VM
Database Directory				
Database alias	IPFVS1A4	IPFVM1A4	Z00VS1A4	Z00VM1A4
Database name	IPFVS1D4	IPFVM1D4	Z00VS1D4	Z00VM1D4
Node name	IPFVS1N4	IPFVM1N4	Z00VS1N4	Z00VM1N4
Authentication	DCS	DCS	DCS	DCS
DCS Database Directory				
local Database	IPFVS1D4	IPFVM1D4	Z00VS1D4	Z00VM1D4
Target Database	SQLVSDB1	SQLVMDB0	SQLDS	SQLDBA
Node Directory				
Node name	IPFVS1N4	IPFVM1N4	Z00VS1N4	Z00VM1N4
Protocol	APPC	APPC	APPC	APPC
Symb.dest.name	IPFVS1S4	IPFVM1S4	Z00VS1S4	Z00VM1S4
Security	Program	Program	Program	Program

Note:
1. IPF - Böblingen,Germany, Z00 for Ljubljana, Slovenia.
2. VM - for VM, VS for VSE.
3. 4 - 3174 communication controller was used.
5. S - Symbolic Destination Name (DCS).
6. N - Node Name.
7. D - Database Name.

5.3.4 CCSID Translation Tables for PS/2 (L1) <----> HOST (L2)

In DB2/2 there is no way to add additional CCSIDs or conversation tables, so the users are limited to the conversion tables which come with the product. Lately, there is enormous pressure to connect to and access different code pages. If the clients in a DRDA environment want to connect within the so-called Latin-1 group there is no problem. However, it is not possible to be connected on a PS platform with L1 code page to a host where SQL/DS is running L2 (870 CP). To activate the PS (L1 CP 850) connection to the host (L2 870), the following steps have been undertaken:

1. The Toronto Lab delivered us a "patch" DB2SYS.DLL which was installed into E:\SQLLIB\DLL\. The code page translation was now enabled, which was not the case in DB2/2 V2.1. CCSID pair 850 (L1 ASCII) and 870 (L2 EBCDIC) is now in use.
2. The SYSTEM.SYSSTRINGS table has been updated. A new pair of translation tables has been inserted. Also note that before you insert new records in a system table from a remote system, make sure that SQL/DS is initiated with the private protocol SQLDS, so that conversion is not enabled.

Let us test DB2/2 client's connection to SQL/DS on VM/ESA or VSE/ESA. The following CCSID (CP/CS) pair is now enabled:
CCSID 850 (L1 ASCII) <----------> (L2 EBCDIC MULTILINGUAL)

ITSO Böblingen Setup

```
CONFIG.SYS ...........COUNTRY=001
                      KEYBOARD=US
                      CODEPAGE=437,850

e:\sqllib\db2 terminate
e:\sqllib\chcp 850
e:\sqllib\set db2codepage=850
e:\sqllib\db2 connect to z00vm1a4 user z00vmu1 using xxxxxx
```

Database Connection Information

```
Database product       = SQL/DS VM 3.5.0

SQL authorization ID   = Z00VMU1

Local database alias   = Z00VM1A4
```

End of ITSO Böblingen Setup

ZPIZ Setup

SQLSTART DB(SQLDBA) PROTOCOL(DRDA) CHARNAME(870)

SQLINIT DB(SQLDBA) PROTOCOL(870)

If you start this type of connection in this specific environment DB2/2 V2.1 will reply

SQL30073. ″119C″ Parameter value ″″ is not supported. SQLSTATE=58017

The CCSD (CP/CS) 870 <----- > 437 is not supported

SQL0332N. There is no available conversion for the source code page ″870″ to the target code page ″437″. Reason Code ″1″. SQLSTATE=57017

End of ZPIZ Setup

Chapter 6. AIX Implementation

6.1 Overview

6.1.1 Our Environment

From the AIX workstation we want to access data in SQL/DS on VM and VSE. The workstation is located in ITSO Böblingen, Germany. SQL/DS on VM and VSE are located both in ITSO Böblingen, Germany, and at the customer site in Ljubljana, Slovenia. We use the 3174 communication controller to establish the connectivity between the workstation and both VM and VSE in Böblingen and Ljubljana. On the AIX workstation we have installed DB2 for AIX V.2.1 and DDCS/6000 V 2.3. The communication manager used is SNA Server/6000 V2.1. Everything is running on operation system AIX 3.2.5 Enhancement 5. This AIX workstation is called ITSO1.

6.1.2 Entries to Database Directories on Workstation ITSO1

To make the database servers on VM and VSE known to DB2 for AIX on ITSO1 we have to put the following information into the database directories of DB2 for AIX. Later on we will discuss this in more detail.

Table 13. Database Directory Entries on ITSO1 for the DRDA Servers

	ITSO Böblingen		**ZPIZ Ljubljana**	
	SQL/DS VSE	SQL/DS VM	SQL/DS VSE	SQL/DS VM
Database Directory				
Database alias	IPFVS1A4	IPFVM1A4	Z00VS1A4	Z00VM1A4
Database name	IPFVS1D4	IPFVM1D4	Z00VS1D4	Z00VM1D4
Node name	IPFVS1N4	IPFVM1N4	Z00VS1N4	Z00VM1N4
Authentication	DCS	DCS	DCS	DCS
DCS Database Directory				
Local Database	IPFVS1D4	IPFVM1D4	Z00VS1D4	Z00VM1D4
Target Database	SQLVSDB1	SQLVMDB0	SQLDS	SQLDBA
Node Directory				
Node name	IPFVS1N4	IPFVM1N4	Z00VS1N4	Z00VM1N4
Protocol	APPC	APPC	APPC	APPC
Symb.dest.name	IPFVS1S4	IPFVM1S4	Z00VS1S4	Z00VM1S4
Security	Program	Program	Program	Program

Note:
1. IPF stands for Böblingen, Germany. Z00 for Ljubljana,Slovenia.
2. VM stands for VM. VS stands for VSE.
3. 4 as the last character means that the 3174 communication controller was used.
4. A - Alias, D - Database, N - Node, S - Symb. Destination Name
5. Symb.dest.name is identical to Side Information Profile name

6.1.3 Entries to SNA Profiles on Workstation ITSO1

Our database connectivity between DB2 for AIX and SQL/DS is based on DRDA. DRDA uses SNA Server/6000 as communication manager. To configure the communication we have to define some SNA profiles. Later we will discuss this in more detail.

Table 14. SNA Profile Entries on ITSO1 for the DRDA Servers

	ITSO Böblingen		**ZPIZ Ljubljana**	
	SQL/DS VSE	SQL/DS VM	SQL/DS VSE	SQL/DS VM
Side Information Profile				
Current profile name	IPFVS1S4	IPFVM1S4	Z00VS1S4	Z00VM1S4
Local LU or CP alias	IPF3174	IPF3174	IPF3174	IPF3174
Fully qualified partner LU name	DEIBMIPF. IPFA21CD	DEIBMIPF. IPFA2GL4	SIZPIZ00. Z00CICS4	SIZPIZ00. Z00AVSL1
Mode name	IBMRDBM	IBMRDBM	IBMRDB	IBMRDB
RPTN	TPN1	SQLVMDB0	TPN1	SQLDBA
Partner LU 6.2 Location Profile				
Profile name	IPFVS4	IPFVM4	Z00VS4	Z00VM4
Fully qualified partner LU name	DEIBMIPF. IPFA21CD	DEIBMIPF. IPFA2GL4	SIZPIZ00. Z00CICS4	SIZPIZ. Z00AVSL1
Partn.LU location method	link_station	link_station	link_station	link_station
Local LU name	IPFBOEDJ	IPFBOEDJ	IPFBOEDJ	IPFBOEDJ
Link Station Profile name	IPFL3174	IPFL3174	Z00L3174	Z00L3174

Note:
1. IPF stands for Böblingen, Germany. Z00 for Ljubljana,Slovenia.
2. VM stands for VM. VS stands for VSE.
3. 4 as the last character means that the 3174 communication controller was used.
4. A - Alias, D - Database, N - Node, S - Symb. Destination Name
5. Symb.dest.name is identical to Side Information Profile name
6. To get the right RPTN names ask SQL/DS administrator.

Table 15 and Table 16 on page 151 give you the SNA Global Information and the Local LU Profile definitions of the ITSO1 workstation.

Table 15. SNA Global Information on ITSO1

SNA Global Information	
Control point (CP) name	DEIBMIPF.IPFP221B
CP alias	IPFP221B
Node ID (for XID)	X′05DE001B′
Node type	End node (EN)

Table 16. Local LU Profile on ITSO1	
6.2 Local LU Profile	
Current profile name	IPF3174
Local LU name	IPFBOEDJ
Local LU alias	IPF3174

6.1.4 Considerations for DRDA AS SQL/DS on VSE

If you connect to SQL/DS on VSE through DRDA be aware that on the VSE side a CICS transaction is invoked to handle the DRDA request. Therefore, some definitions must be added to the CICS environment on VSE to make the workstation known to CICS and enable it to submit the transaction.

See 4.1.1, "CICS/VSE Setup" on page 84 for more information.

6.1.5 VTAM Considerations

Check also the VTAM definitions for the RISC/6000 workstation. These are similar to those for the PS/2. See Appendix A, "System Tables and Sample Programs" on page 191 for more information.

6.2 Connectivity

6.2.1 Overview

From our workstation we want to have a transparent access to data located in SQL/DS on VM and VSE. Therefore, we use the DRDA application requester functionality (DRDA AR) of DDCS/6000. The workstation on which DDCS is installed is called the DDCS workstation. The DRDA AR will handle the connectivity and data transfer to and from the DRDA application server (AS). The DRDA AS can be accessed by local clients on the DDCS workstation and by remote clients. For remote clients the DDCS workstation acts as a multi-user gateway between the client workstation and the DRDA AS database management system. Because DRDA connectivity is based on APPC LU 6.2 connectivity, we use SNA Server/6000 as a communication manager. To make that connectivity work we have to define some SNA profiles. In this part these profiles and the used parameter values are shown.

6.2.2 SNA Profiles Relationship

In the following figure you can see which profiles we have to configure and how these profiles work together.

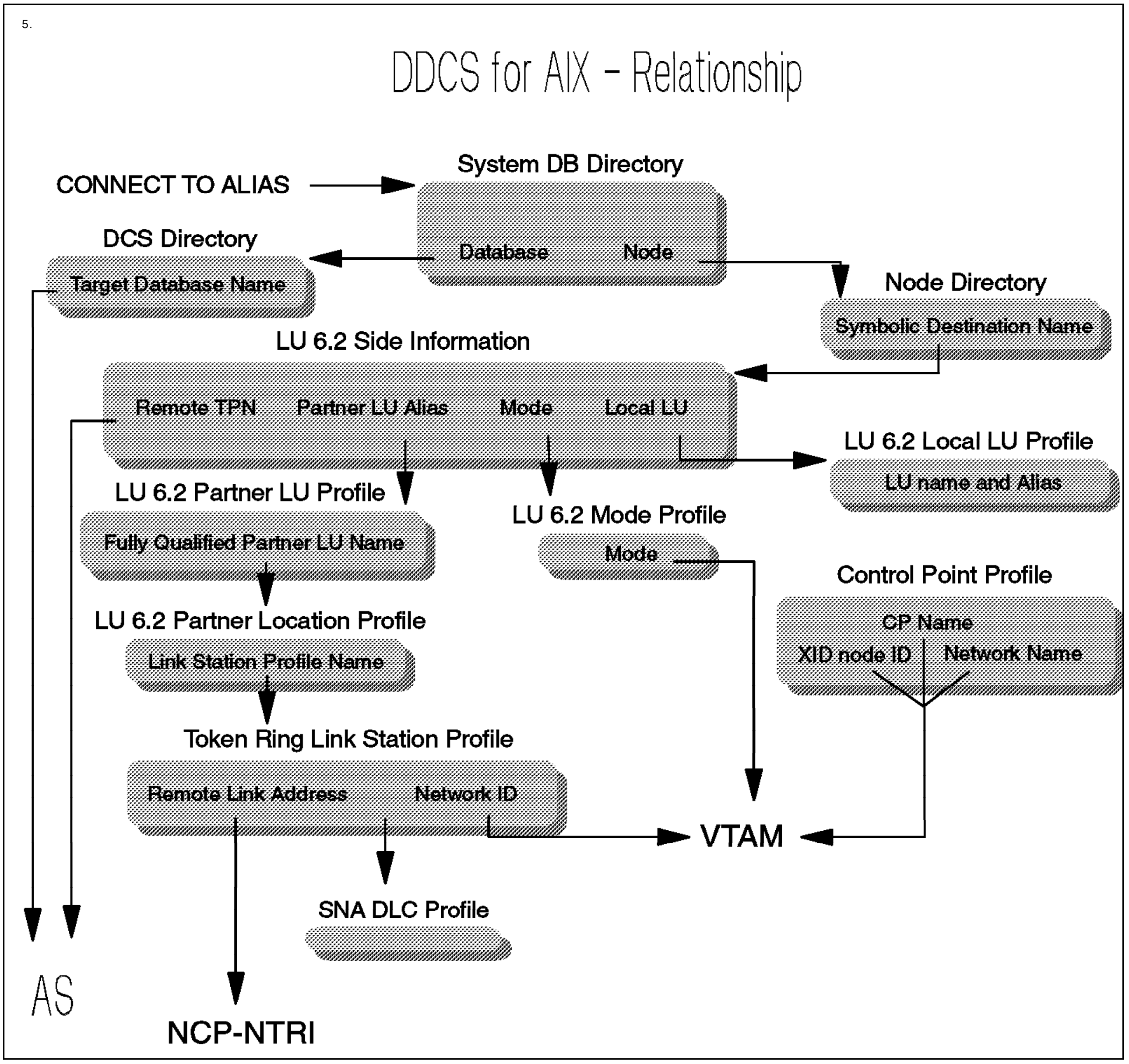


Figure 174. SNA Profiles Relationship

6.2.3 Configuring the System

For our connection in Böblingen we used one Token-Ring Adapter attached to a LAN and connected to a 3174 communication controller.

6.2.3.1 Token-Ring Adapter

Before configuring the SNA profiles you must verify that the Token-Ring Adapter is configured correctly. To do so invoke SMIT and select the following items:

1. Devices
2. Communications
3. Token-Ring Adapter

4. Adapter
5. Change / Show Characteristics of a Token-Ring Adapter

```
                                         †Entry Fields†
Token-Ring Adapter                           tok0
Description                                  Token-Ring High-Performance
Status                                       Available
Location                                     00-05
Receive data transfer OFFSET                 92
TRANSMIT queue size                          30
RECEIVE queue size                           30
STATUS BLOCK queue size                      10
RING speed                                   16
Receive ATTENTION MAC frame                  no
Receive BEACON MAC frame                     no
Enable ALTERNATE TOKEN RING address          yes
ALTERNATE TOKEN RING address                 0x40001010101b
Apply change to DATABASE only                no
```

Figure 175. Show Characteristics of a Token-Ring Adapter

Note: Ensure that Status is set to Available and that the RING speed corresponds to the speed in your LAN.

6.2.3.2 Data Link Control

SNA Server/6000 requires that you supply information about Data Link Control (DLC) characteristics in the SNA DLC Profile. Therefore, you must have DLC configured and available. DLC is a set of communication protocols that supports orderly exchanges of data over a link. Basically, the DLC manages a communications adapter. Before the system can use an installed DLC, you must first add the desired DLC manager. In our case the desired DLC manager is dlctoken because we are using a Token-Ring Adapter. If this step has been done you can check if the DLC manager is available by issuing the following command: lsdev -Cc dlc. As a result you will receive the message:

```
dlctoken Available  Token-Ring Data Link Control
```

Figure 176. Received Message for DLC Manager

6.2.3.3 Configuring SNA Profiles

All the SNA profiles are configured using the advanced configuration. To do so invoke SMIT and select the following items:

1. Communications Applications and Services
2. SNA Server/6000
3. Configure SNA Profiles
4. Advanced Configuration

6.2.3.4 Token-Ring Link Station Profile

```
                                                 †Entry Fields†
Current profile name                               IPFL3174
New profile name
Use Control Point¢s XID node ID?                   yes
   If no, XID node ID                              *
SNA DLC Profile name                               tok0.00001
Stop link station on inactivity?                   no
   If yes, Inactivity time-out (0-10 minutes)      0
LU address registration?                           no
   If yes,
      LU Address Registration Profile name
Trace link?                                        no
   If yes, Trace size                              long

Adjacent Node Address Parameters
   Access routing                                  link_address
   If link_name, Remote link name
   If link_address,
      Remote link address                          400020201001
      Remote SAP address (02-fa)                   04

Adjacent Node Identification Parameters
   Verify adjacent node?                           no
   Network ID of adjacent node                     DEIBMIPF
   CP name of adjacent node
   XID node ID of adjacent node (LEN node only)    *
   Node type of adjacent node                      learn

Link Activation Parameters
   Solicit SSCP sessions?                          yes
   Initiate call when link station is activated?   yes
   Activate link station at SNA start up?          yes
   Activate on demand?                             no
   CP-CP sessions supported?                       yes
   If yes,
      Adjacent network node preferred server?      no
   Partner required to support CP-CP sessions?     no
   Initial TG number (0-20)                        0

Restart Parameters
   Restart on activation?                          no
   Restart on normal deactivation?                 no
   Restart on abnormal deactivation?               no

Transmission Group COS Characteristics
   Effective capacity                              15974400
   Cost per connect time                           0
   Cost per byte                                   0
   Security                                        nonsecure
   Propagation delay                               lan
   User-defined 1                                  128
   User-defined 2                                  128
   User-defined 3                                  128
```

Figure 177. Show Token-Ring Link Station Profile

The first SNA profile we configure is the Token-Ring Link Station. This profile associates the link station with either the CP for APPN services or a user-specified PU. It also associates the link station with an SNA DLC Profile, which defines the hardware adapter and DLC characteristics of the link.

To configure the Token-Ring Link Station invoke SMIT, go to step 4 of 6.2.3.3, “Configuring SNA Profiles” on page 153 (Advanced Configuration of SNA profiles) and select the items:

5. Links
6. Token-Ring
7. Token-Ring Link Station

Following is a brief explanation of the most relevant parameters in Figure 177 on page 154:

Use Control Point′s XID node ID?
Select one of the following values:

- **yes.** Associates the link station with the XID node ID of the APPN CP. Selecting this value places the link station under the control of the APPN CP, with full APPN support.
- **no.** Excludes the link station from the control of the APPN CP and places it under the control of a PU you designate. If you select this value, you must specify the XID node ID of the PU represented by this link station. Specifying link stations with unique XID node ID values enables SNA Server/6000 V2.1 to support dependent LU traffic with multiple hosts.

XID node ID
Enter a hexadecimal value that identifies the PU that controls this link station. The system default value is an asterisk (*), which represents an XID value of 00000000. This indicates that no XID node ID is specified.

LU address registration?
If the value you select is **yes**, generic LU addresses are registered to this link station. Select this value only if this link station is used for generic SNA sessions. You also must create a Generic LU Address Registration Profile containing the registered LUs. Enter the name of the registration profile in the LU Address Registration Profile name field for this link station. If you select **no**, you are not using generic LU address registration and do not require a generic LU address registration profile for this link station.

Remote link address
For a call or selective listen link station, enter the network address of the remote station. The network address must be unique to the specific adapter card of the remote station. If you enter *link address* in the Access routing field but do not enter a remote link address, the link station is a *nonselective listen link station*, that is, it accepts a link activation request from any remote station, but it can handle only one link at a time.

Network ID of adjacent node
Enter the network ID of the adjacent node. If the adjacent node does not supply a CP name in XID exchanges, you must enter a value in this field and define an LU 6.2 Partner Location Profile to define the adjacent node resources.

6.2.3.5 Control Point Profile

The Control Point Profile describes the characteristics of the CP on the local RISC System/6000. A RISC System/6000 workstation must have one (and only one) Control Point Profile. To configure this profile invoke SMIT, go to step 4 of 6.2.3.3, "Configuring SNA Profiles" on page 153 (Advanced Configuration of SNA profiles) and select the item:

5.Control Point

The following information is shown within this profile:

```
                                                    †Entry Fields†
Profile name                                           node_cp
XID node ID                                            05de001b
Network name                                           DEIBMIPF
Control Point (CP) name                                IPFP221B
Control Point alias                                    IPFP221B
Control Point type                                     appn_end_node
Maximum number of cached routing trees                 500
Maximum number of nodes in the TRS database            500
Route addition resistance                              128
Comments
```

Figure 178. Show Control Point Profile

Following is a brief explanation of the most relevant parameters in Figure 178:

Profile name
This field displays the name of the Control Point Profile, *node_cp*. The system uses the profile name to refer to the set of characteristics that you describe in the profile. You must have one (and only one) Control Point Profile in SNA Server/6000 V2.1; you cannot change its name.

XID node ID
Enter a value that the CP on the local system can use to identify itself as a PU. This value is used only for switched links (Token-Ring, Ethernet, X.25, and switched SDLC). Enter one of the following values:

- For a peer network using independent LU 6.2, use the default wildcard value, (*).
- For dependent communication with a host system, specify a unique XID node ID. This value must uniquely identify the local node in the subarea network.

The XID node ID has two parts:

Block Number
The first three hexadecimal digits provide an identifier, or block number, that is unique to each product on the network. The value of *071* identifies the local node as a RISC System/6000 workstation. If you are connecting to a VTAM host, the block number you specify must match the IDBLK parameter value in the VTAM PU definition macro.

ID number The last five digits distinguish a specific piece of equipment from all other similar pieces of equipment on the network. If you are connecting to a VTAM host, the ID number is the IDNUM parameter value in the VTAM PU definition macro.

Network name
Enter the network name of the CP. The network name distinguishes this network from other networks to which it can be connected. The network name, combined with the CP name, forms a unique identifier (the fully qualified name) for this CP.

Control Point name
The Control Point (CP) name distinguishes this CP from other CPs that may be connected to the network. This name, combined with the network name, forms a unique identifier (fully qualified name) for this CP.

Control Point alias
Enter an alias that can be used on the local system in place of the CP name. Using an alias for the control point enables you to change the CP name without having to update all LU 6.2 Side Information Profiles that refer to the CP.

Control Point type
Select one of the following values:

- *appn_network_node* The CP provides NN services.
- *appn_end_node* The CP provides APPN EN services. If you do not provide APPN services, choose *appn_end_node* as your CP type.

6.2.3.6 LU 6.2 Local LU Profile

To configure this profile invoke SMIT, go to step 4 of 6.2.3.3, “Configuring SNA Profiles” on page 153 (Advanced Configuration of SNA profiles) and select the items:

5. Sessions
6. LU 6.2
7. LU 6.2 Local LU

```
                                                    †Entry Fields†
Current profile name                                  IPF3174
New profile name
Local LU name                                         IPFBOEDJ
Local LU alias                                        IPF3174
Local LU is dependent?                                no
   If yes,
      Local LU address (1-255)
      System services control point
         (SSCP) ID (*, 0-65535)                       *
      Link Station Profile name
Conversation Security Access List Profile name
Comments                                              LU used for 3174
```

Figure 179. Show LU 6.2 Local LU Profile

Following is a brief explanation of the most relevant parameters in Figure 179:

Profile name
Enter a name for the new LU 6.2 Local LU profile in this field.

Local LU name
Enter the name of the local LU in this field. This LU name must be unique with respect to all other local LUs defined on this node, and its fully qualified local LU name must be unique across the network. SNA Server/6000 V2.1 generates the fully qualified LU name by concatenating this LU name to the network name defined in the Control Point profile (DEIBMIPF.IPFBOEDJ).

Local LU alias
Enter an alias for the local LU. This alias can be used in an LU 6.2 Side Information Profile or with the *sna* command to refer to this local LU.

Local LU is dependent?
Select one of the following values to specify whether the LU 6.2 is dependent or independent:

- **yes** The local LU 6.2 is dependent, and cannot initiate LU to LU sessions. A host SSCP controls session activation.
- **no** The local LU 6.2 is independent, and sessions can be activated without SSCP.

Local LU address
If the local LU is dependent, enter the local LU address in this field. The value you enter must match the value specified for the LOCADDR parameter in the VTAM/NCP LU resource definition statement.

SSCP ID This field specifies the ID of the SSCP that controls this dependent LU in the subarea network. This identifier is defined on the host system to which this node is connected and is used for host verification.

Link Station Profile name
Select the name of the Link Station Profile shown in Figure 177 on page 154 that describes the characteristics of the link station used for sessions with the remote LU.

Conversation Security Access List Profile name
Enter the name of an LU 6.2 Conversation Security Access List profile. This profile specifies a set of user names with permission to remotely allocate a session with this local LU. If the user name is a valid AIX user, you do not have to configure a Conversation Security Access List profile.

6.2.3.7 LU 6.2 Partner Location Profile

To configure this profile invoke SMIT, go to step 4 of 6.2.3.3, “Configuring SNA Profiles” on page 153 (Advanced Configuration of SNA profiles) and select the items:

5. Sessions
6. LU 6.2
7. Partner LU 6.2 Location

```
                                             †Entry Fields†

Current profile name                            IPFVM4
New profile name
Fully qualified partner LU name                 DEIBMIPF.IPFA2GL4
Partner LU location method                      link_station
If owning_cp,
   Fully qualified owning Control Point (CP) name
   Local node is network server for LEN node?   no
   Fully qualified network node server name
If link_station,
   Local LU name                                IPFBOEDJ
   Link Station Profile name                    IPFL3174
Comments
```

Figure 180. Show Partner LU 6.2 Location Profile

Following is an explanation of the most relevant parameters described in Figure 180:

- **Partner LU location method**

  Two methods are used to support your partner LU 6.2:

  – owning_cp

    Specified when the partner LU is located by first finding the CP in the network specified by the name given in the `Fully qualified owning Control Point (CP) name` field. APPN directory and routing services first locate the CP in the network. Once the CP is located, the BIND is forwarded to the partner LU on the CP over the link on which it was found.

  – link_station

    Specified when the partner LU is assumed to be located on the other side of the link identified by the Link Station profile name specified in the `Link Station Profile name` field. In this case, APPN directory and routing services are completely bypassed; the BIND is immediately sent across the specified link in anticipation of the partner LU being available on the opposite side of the link.

    Note that the entry in the `If link_station` field is valid only if the local node type specified in the Control Point profile is *appn_end_node*. APPN network nodes cannot statically associate a partner LU's location with a particular link.

- **Local LU name**

  Provides the name of the local LU to be matched with the partner LU name specified in the `Fully qualified partner LU name` field to identify which session is to use the link specified in the `Link Station Profile name` field. This value is valid only when the Partner LU location method is *link_station*.

- **Link Station Profile name**

  Provides the name of the Link Station profile which specifies the link over which the session identified by the `Fully qualified partner LU name` and `Local LU name` fields is to activate. This value is valid only when the Partner LU location method is *link_station*.

For the other Partner LU 6.2 Location profiles we defined refer to B.1.6, “Partner LU 6.2 Location” on page 232.

6.2.3.8 LU 6.2 Mode

To configure this profile invoke SMIT, go to step 4 of 6.2.3.3, “Configuring SNA Profiles” on page 153 (Advanced Configuration of SNA profiles) and select the items:

5. Sessions
6. LU 6.2
7. LU 6.2 Mode

```
                                                    †Entry Fields†
Current profile name                                   IBMRDBM
New profile name
Mode name                                              IBMRDBM
Maximum number of sessions (1-5000)                    60
Minimum contention winners (0-5000)                    30
Minimum contention losers (0-5000)                     30
Auto activate limit (0-500)                            0
Upper bound for adaptive receive pacing window         16
Receive pacing window (0-63)                           3
Maximum RU size (128,...,32768: multiples of 32)       4096
Minimum RU size (128,...,32768: multiples of 32)       1024
Class of Service (COS) name                            #CONNECT
Comments
```

Figure 181. Show LU 6.2 Mode Profile

Following is an explanation of the most relevant parameters in Figure 181:

- **Profile name** Enter the name of the new Mode profile. The system uses the profile name to refer to the set of characteristics that you describe in the Mode profile.
- **Mode name** Enter the SNA mode name in this field. The SNA network uses this mode name to identify the set of session parameters associated with the mode.
- **Maximum number of sessions** Enter a value to specify the maximum number of parallel sessions allowed between a local and partner LU using this mode. If the partner LU supports fewer parallel sessions on this mode, the value entered here can be negotiated downward.
- **Minimum contention winners** This field defines the minimum number of contention winner sessions that must be reserved for sessions established by the local LU with a partner LU.
- **Minimum contention losers** This field specifies the minimum number of contention loser sessions that must be reserved for sessions established by the local LU with a partner LU. In conjunction with the `Minimum contention winners` field, this field determines how to resolve contention for a session.
- **Receive pacing window** Enter the number of request units (RUs) that the local LU can receive before it must send a pacing response to the remote LU.
- **Maximum RU size** This field specifies the maximum number of bytes in each request and response unit that are sent and received in sessions using this mode.

- **Minimum RU size** The value in this field defines the minimum size for RUs that can be sent and received in sessions using this mode.
- **Class of Service (COS) name** Enter the class of service (COS) name to specify transmission group characteristics for the mode. You can use an IBM-supplied COS or your own.

  SNA Server/6000 V2.1 includes the following IBM-supplied COS definitions:

  – **#BATCH** Specifies a standard, nonsecure COS with medium transmission priority for batch sessions.

  – **#CONNECT** Specifies a standard, nonsecure COS with medium transmission priority.

  – **#CPSVCMG** Specifies a COS used for establishing CP-CP sessions between two NNs, between two ENs, or between an EN and its NN server in an APPN network.

  – **#INTER** Specifies a standard, nonsecure COS with high transmission priority for interactive sessions.

  – **#INTERSC** Specifies a secure COS with high transmission priority for interactive sessions.

  – **SNASVCMG** Specifies a COS used for negotiating session limits for LU 6.2 independent sessions.

For the other LU 6.2 Mode profiles we defined refer to B.1.7, “LU 6.2 Mode” on page 234.

6.2.3.9 LU 6.2 Side Information Profile

To configure this profile invoke SMIT, go to step 4 of 6.2.3.3, “Configuring SNA Profiles” on page 153 (Advanced Configuration of SNA profiles) and select the items:

5. Sessions
6. LU 6.2
7. LU 6.2 Side Information

```
                                                  †Entry Fields†

Current profile name                                 IPFVM1S4
New profile name
Local LU or Control Point alias                      IPF3174
Provide only one of the following:
   Partner LU alias
   Fully qualified partner LU name                   DEIBMIPF.IPFA2GL4
Mode name                                            IBMRDBM
Remote transaction program name (RTPN)               SQLVMDB0
RTPN in hexadecimal?                                 no
Comments
```

Figure 182. Show LU 6.2 Side Information Profile

Following is an explanation of the most relevant parameters in Figure 182:

- **Profile name** Enter a name for the new Side Information profile. A local TP can specify this name in the OPEN call to identify a set of session parameters. The session parameters include the local LU, partner LU, mode, and remote TP names.

- **Local LU or Control Point alias** Enter the alias of either the local LU or node CP that identifies the local LU for any conversations initiated by TPs using this Side Information profile. If you enter an alias, it must match the alias defined in either the Control Point profile or the LU 6.2 Local LU profile. If you do not enter a value in this field, the node CP serves as the local LU for the Side Information profile.
- **Partner LU alias** Enter or select an alias for the partner LU that specifies the default partner LU for conversations initiated by TPs using this Side Information profile. The TP can override this value. If you enter an alias in this field, you must enter the same alias in the Partner LU alias field of the Partner LU profile.
- **Fully qualified partner LU name** Enter a fully qualified partner LU name similar to that shown in Figure 182 on page 161. It consists of two parts, *network_name.lu_name* (DEIBMIPF.IPFA2GL4).
- **Mode name** Enter or select the name of the mode that specifies the default mode name for conversations initiated by local TPs that use this Side Information profile. Keep in mind that the mode name entered here must correspond to a mode name entered in the LU 6.2 Mode profile. The same mode name must be available to the partner LU.
- **Remote transaction program name (RTPN)** To specify a default TPN for all TPs that use this Side Information profile, enter the TPN to be invoked on the remote system.
- **RTPN in hexadecimal?** Select one of the following values:
  - **yes**. The value entered in the Remote transaction program name (RTPN) field is hexadecimal format.
  - **no**. The value entered in the Remote transaction program name (RTPN) field is ASCII format.

For the other LU 6.2 Side Information profiles we defined refer to B.1.8, "LU 6.2 Side Information Profile" on page 235.

6.2.3.10 Verifying SNA Profiles

After entering or changing values of these SNA profiles you must verify them by invoking SMIT and selecting *Verify Configuration Profiles*.

You should get a screen similar to that shown in Figure 183 on page 163.

```
                         Verify Configuration Profiles

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        (Entry Fields)
Update action if verification successful                     none
If normal_update or dynamic_update,
   Backup file for committed database                      ()
   Backup security file for committed database             ()



F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 183. Verify Configuration Profiles Screen

You have the following options for update actions if the verification is successful:

- none

  Select this option if the verified profiles should remain in the working database. To update the committed database with these profiles, run verification again and select *normal_update* or *dynamic_update* for this field.

- normal_update

  Select this option to update the committed database only if SNA Server/6000 V2.1 is not running. If all profiles are successfully verified, SNA Server/6000 V2.1 promotes them to the committed database. If any errors are found during verification, the new profiles remain in the working database, and you must correct the errors and run verification again to update the committed database.

- dynamic_update

  Select this option to update the committed database while SNA Server/6000 V2.1 is running. If all profiles are successfully verified, SNA Server/6000 V2.1 promotes them to the committed database. If SNA Server/6000 V2.1 is not actually running, this option still updates the committed database if verification is successful. If any errors are found during verification, the new profiles remain in the working database, and you must correct them and run verification again to update the committed database.

6.2.3.11 Testing the Connections

Once verification has successfully completed and the committed database has been updated, proceed to start the SNA Server/6000 V2.1 using SMIT or the command line.

To use SMIT type the following command:

```
smit sna
```

and select the following items:

1. Manage SNA Resources
2. Start SNA Resources
3. Start SNA

From the AIX prompt issue the following command:

```
/usr/bin/sna -start
```

This command automatically starts SNA Server/6000 V2.1, link stations, and LU 6.2 sessions. You can check that the SNA Server/6000 V2.1 started correctly by issuing the following command:

```
lssrc -ssna
```

Test whether you can establish an LU-to-LU session. Although the session establishment is done automatically when you issue the DB2 CONNECT statement, verify it before issuing the first CONNECT statement. If a session cannot be established, the DB2 CONNECT statement will fail.

To establish an LU-to-LU session, invoke SMIT, issue the following command:

```
smit sna
```

and select the following items:

1. Manage SNA Resources
2. Start SNA Resources
3. Start an SNA Session

You should get a screen similar to that shown in Figure 184.

```
                         Start an SNA Session

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    (Entry Fields)
Specify one of the following:
   Local LU name                                       (IPFBOEDJ)
   Local LU alias                                      ()

Specify one of the following:
   Fully qualified partner LU name                     (DEIBMIPF.IPFA2GL4)
   Partner LU alias                                    ()

Mode name                                              (IBMRDBM)

Session type                                            conwinner


F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 184. Start an SNA Session Screen

On the Start an SNA Session screen specify the values you specified when configuring the LU 6.2 Side Information profile.

If the establishment of the session is successful, you receive a message similar to the following:

```
                              COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

Session started: Session ID = EF178F5683653894, Conversation group ID =3




F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit
```

6.3 Database

6.3.1 Overview

The database that a client wants to use has to be known in a database management system. Therefore, the database is cataloged in a database catalog. In DB2 for AIX we use the Database Directory to catalog all the databases to which we want to connect. These databases may be located on the same workstation as the database directory or on a remote system. If the database is on a remote system we have to specify this in the database directory but also in a node directory which gives more information about this remote node. If the database on the remote system is accessed through DRDA we have to give some information within the DCS directory of DB2 for AIX.

To connect to a DRDA database server, DDCS requires information in the database directories. If you want to add, change, or delete entries in the database directories, you must have either SYSADM or SYSCTRL authority.

You must specify entries in the following directories to enable the DDCS workstation to connect to a DRDA AS:

- System database directory
- Database connection services (DCS) directory
- Node directory

6.3.2 DB2 for AIX System Database Directory

The system database directory contains an entry for each database that can be accessed from a workstation. The directory contains information about whether a database is local or remote, the authentication type, the database name, and the database alias. When you create a database, it is automatically cataloged in the system database directory. On the client you may use the CATALOG DATABASE command to explicitly catalog a database in the system database directory. With the command ***db2 "list database directory"*** you will see the entries made to the system database directory.

```
  System Database Directory

Number of entries in the directory = 4

Database 1 entry:

Database alias                  = IPFVS1A4
Database name                   = IPFVS1D4
Node name                       = IPFVS1N4
Database release level          = 6.00
Comment                         =
Directory entry type            = Remote
Authentication                  = DCS
```

Figure 185. Entry in System Database Directory on ITSO1

Refer to B.2.1, "DB2 for AIX Database Directory" on page 236 for the other entries of the system database directory.

- **Database name** is the real name of the remote database.
- **Database alias** is the alias you want to use for this database. If you do not provide an alias, the default is the same as the database name. To reference the cataloged database, you must use the alias.
- **Node name** is the name you specified to define the remote node on which the database is stored.
- **Directory entry type** specifies whether the database is a local or a remote database.
- **Authentication** determines how and where a user who wants to access the database will be verified. If the database is within an instance of a DB2 Version 2 database server this value is ignored, even though the value appears in the database directory. For databases in a DB2 Version 2 instance the authentication type is the same as for the instance. The authentication type is stored in the database manager configuration file on the server. It is initially set when the instance is created. For databases in a DB2 Version 1, every database can have a different authentication type. Then the authentication value specified in the client's system database directory must match the database authentication type on the server. The following authentication types are provided:

  SERVER forces the authentication on the database server. So the user name and password specified during the connection attempt are compared to the valid user name and password combinations on the server.

  CLIENT specifies that authentication occurs on the node where the application is invoked. The user name and password specified during the connection attempt are compared to the valid user name and password combinations on the client node. No further authentication will take place on the database server.

  DCS specifies how authentication will take place for databases accessed using DDCS. The behavior is the same as for AUTHENTICATION SERVER. The difference is in how DDCS interprets the authentication type. When authentication is SERVER, DDCS forces authentication on the DDCS workstation. When authentication type is DCS, DDCS assumes authentication takes place on the DRDA AS.

See the *Installation and Operation Guide* or the *Installing and Using Clients* manual for more information.

6.3.3 DB2 for AIX Database Connection Services Directory

The DCS directory contains an entry for each DRDA AS that can be accessed from the workstation. When the application issues a CONNECT statement specifying a remote database using APPC protocol, the database manager searches the DCS directory for a matching database name entry. If it finds a matching entry, it uses the DRDA protocol for the connection. If it does not find a matching entry, it uses the private protocol. To see all DCS databases issue the following command: *db2 "list dcs directory"*

```
    Database Connection Services (DCS) Directory

Number of entries in the directory = 4


DCS 1 entry:

Local database name                = IPFVS1D4
Target database name               = SQLVSDB1
Application requestor name         =
DCS parameters                     =
Comment                            =
DCS directory release level        = 0x0100
```

Figure 186. Entry in DCS Directory on ITSO1

Refer to B.2.2, "DB2 for AIX DCS Directory" on page 237 for the other entries of the DCS directory.

The following is a short description of the parameters:

- **Local database name**

  Specifies the DCS database name to catalog. This name must match the database name you entered when you cataloged this database in the system database directory.

- **Target database name**

  Specifies the real name (RDB_NAME) of the target host database to catalog.

- **Application requestor name**

  If using the DDCS AR code, you do not have to specify a value for this parameter. The default value causes DDCS to be invoked.

- **DCS parameters**

  You can specify a parameter string defining SQLCODE mapping, interrupt handling, and disconnect events. For further explanation of this string, refer to the *DDCS User's Guide for Common Servers*.

You can use CLP to catalog an entry in the DCS directory by issuing the following command:

```
DB2 catalog dcs database IPFVS1D4 as SQLVSDB1
```

6.3.4 DB2 for AIX Node Directory

You can use CLP to list all entries in the node directory by issuing the following command:

```
db2 † list node directory†
```

For each node you will get the following information:

```
 Node Directory

Number of entries in the directory = 4


Node 1 entry:

Node name                          = IPFVS1N4
Comment                            =
Protocol                           = APPC
Symbolic destination name          = IPFVS1S4
Security type                      = PROGRAM
```

Figure 187. Entry in Node Directory on ITSO1

Refer to B.2.3, “DB2 for AIX Node Directory” on page 238 for the other entries of the node directory.

The following is a short description of the parameters:

- **Node name**

  This is an alias that identifies the remote node. It should be a meaningful name to make it easier to remember.

- **Protocol**

  As we use DRDA, the protocol must be APPC.

- **Symbolic destination name**

  This name must correspond to the LU 6.2 Side Information Profile name. It contains the necessary information to set up an APPC connection to the server (partner LU name, mode name, partner TPN).

- **Security type**

  – None

    Specifies that no security information is sent to the partner LU. This option is not supported by DRDA.

  – Same

    Specifies a user ID together with an indicator that the user has been already verified. Only the user ID is sent to the partner LU. The password is not sent to the partner LU. This implementation requires that the partner LU at the AS be configured to accept already verified security.

– Program

Specifies that a user ID and a password must be included in the allocation request sent to the partner LU for authentication.

You can use CLP to catalog an entry in the node directory by issuing the following command:

```
db2 †catalog appc node IPFVS1N4 remote IPFVS1S4 security program†
```

You have to catalog one entry in the node directory for each AS to which you intend to connect.

6.4 Security

Security is implemented by two separate mechanisms:

- Authentication
- Internal security

In a distributed environment, authentication can take place in one or more locations, depending on the configuration of the DBMSs and network products.

The internal security of the DB2 products then rules what the user can do on DB2 resources, after the user has been authenticated.

On the AIX platform, user ID and password information is managed by the operating system itself, whereas on the OS/2 platform UPM manages user ID and password information. First we discuss the security considerations for a DDCS stand-alone workstation. The considerations are also valid if the DDCS workstation is a gateway workstation used by a local user.

Whether your configuration is stand-alone or client/server, the AUTHENTICATION parameter in the database manager configuration file applies only for local databases. When cataloging a remote database, the value you specify for the AUTHENTICATION parameter overrides the value specified in the database manager configuration file.

6.4.1 DDCS for AIX: Stand-alone or Local Users

When you catalog a database in the system database directory you have three options for the AUTHENTICATION parameter value:

- AUTHENTICATION=CLIENT

  Essentially the authentication takes place in the client workstation. For a local user or a stand-alone configuration, authentication takes place locally.

- AUTHENTICATION=SERVER

  This value is conceptually the same as AUTHENTICATION=CLIENT; the user is authenticated locally. In addition, the authentication process can take place in the DRDA AS. We describe these implementations below when we explain the SECURITY parameter.

- AUTHENTICATION=DCS

  Authentication takes place on the DRDA AS.

When you catalog the APPC node for the DRDA connection you have two options for the SECURITY parameter value:

- SECURITY=SAME

  A password is not sent to the DRDA AS.

- SECURITY=PROGRAM

  A password is sent to the DRDA AS.

If you specify AUTHENTICATION=CLIENT, you have to specify SECURITY=SAME. In this case the user ID is authenticated locally, and only the user ID is sent to the DRDA AS. If the DRDA AS uses VTAM, the APPL statement for the RDBMS must specify SECACPT=ALREADYV. If the DRDA AS is DB2 for OS/400, the SEC parameter in the remote configuration list must be set to *YES for the LU that represents this DDCS workstation. Refer to the DRDA AS security sections for the platform specifics.

If you specify AUTHENTICATION=SERVER and SECURITY=SAME, the user ID is authenticated locally, and only the user ID is sent to the DRDA AS.

If you specify AUTHENTICATION=SERVER and SECURITY=PROGRAM, the user ID is first authenticated locally. If the authentication process is successful, the user ID and the password are sent to the DRDA AS, where the user is authenticated again.

If you specify AUTHENTICATION=DCS, the only option you have is SECURITY=PROGRAM. The user ID is not authenticated locally; the user ID and the password combination are sent to the DRDA AS, where the user is authenticated.

Table 17 summarizes the authentication scheme for a DDCS workstation without remote clients.

Table 17. Security Matrix: DDCS without Remote Clients

DDCS Workstation			**DRDA AS**
Authentication	**Security**		
CLIENT	SAME	userid/pwd	userid
SERVER	SAME	userid/pwd	userid
SERVER	PROGRAM	userid/pwd	userid/pwd
DCS	PROGRAM	userid/pwd	userid/pwd

The reverse highlighted userid/pwd shows the user ID and password combination that is verified and where authentication takes place.

6.4.2 DDCS for AIX: Remote Client Support

For remote clients, security in AIX differs slightly from security in OS/2. The following three options are available when you catalog a database in the client workstation:

- AUTHENTICATION=CLIENT

  Requires that the database is cataloged with AUTHENTICATION=CLIENT on the DDCS gateway as well. However, it is not necessary to define in AIX the same user ID as used on the client.

- AUTHENTICATION=SERVER or DCS

  Whichever of these two options is specified in the client workstation, the user ID and password are sent to the DDCS gateway. The security is then managed by the AUTHENTICATION type specified at the DDCS gateway workstation:

  – AUTHENTICATION=CLIENT This specification is incompatible with the type of authentication specified at the client.

  – AUTHENTICATION=SERVER and SECURITY=SAME The user ID and password are sent to the DDCS gateway workstation and authenticated there.

  – AUTHENTICATION=SERVER and SECURITY=PROGRAM The user ID and password are sent to the DDCS gateway workstation, where authentication takes place. If successful, the user ID and password are routed to the DRDA AS where they are authenticated again.

  – AUTHENTICATION=DCS and SECURITY=PROGRAM The user ID and password are sent to the DDCS gateway workstation and then sent to the DRDA AS. Authentication is performed at the DRDA AS only.

The communications protocol used to connect clients to the DDCS workstation has no impact on the authentication scheme. However, as a general guideline, if the connection is APPC, we recommended that you use SECURITY=NONE on the client (note that SECURITY=NONE cannot be used on the DDCS gateway workstation).

The overall process is not influenced by the AUTHENTICATION parameter set at the DDCS database manager instance level, which would have led to less granularity in the specification of the security policies to access different DRDA ASs.

For a DDCS workstation serving remote clients, the authentication mechanism works in exactly the same way as shown in the following table.

Table 18 summarizes the authentication scheme for a DDCS workstation serving remote clients. Note how we have essentially nested Table 17 on page 170 in a broader scenario to account for the clients.

Table 18. Security Matrix: DDCS with Remote Clients

Client Workstation		**DDCS Workstation**			**DRDA AS**
Auth		**Auth**	**Sec**		
CLIENT	userid/pwd	CLIENT	SAME	userid	userid
SERVER or DCS	userid/pwd	SERVER	SAME	userid/pwd	userid
SERVER or DCS	userid/pwd	SERVER	PROGRAM	userid/pwd	userid/pwd
SERVER or DCS	userid/pwd	DCS	PROGRAM	userid/pwd	userid/pwd

Authentication takes place wherever userid/pwd is reverse highlighted.

6.4.3 User ID and Password Folding

AIX is a case-sensitive environment, and user IDs and passwords are commonly stored in lowercase format. Before a user ID and password can be sent to the DRDA AS, they must be folded to uppercase.

If the SECURITY type specified at the DDCS workstation is SAME, only the user ID is sent to the AS. To have the user ID folded to uppercase before it is sent:

- For local clients, the login user must belong to the AIX system group. If the login user issues a CONNECT in the form:

  ```
  db2 connect to database user another_user using password
  ```

  The user specified in the CONNECT statement does not have to necessarily belong to the AIX system group.

- For remote clients, the DDCS instance owner must belong to the AIX system group, otherwise the instance owner user ID, which is not folded to uppercase, is sent to the DRDA AS.

Note that these problems can be circumvented when the DRDA AS is DB2 for MVS, because APAR PN70160 provides user ID and password folding on the MVS platform before RACF facilities are invoked. Another way to circumvent these problems is to perform inbound translation at the DRDA AS location.

The following redbooks are useful sources of information about security: *DB2 for MVS DRDA Server: Security Consideration, GG24-2500* and *DB2 for MVS Connections with AIX and OS/2, SG24-4558*.

6.4.4 Database Considerations

A valid user ID must be presented to the DRDA AS. Where authentication is performed is of no relevance. Once the DRDA AS has the valid user ID, its internal security facility controls any attempts to access its resources (tables, indexes, packages). The minimum requirement is that the user ID be granted the privilege to connect to the DRDA AS.

6.4.5 DB2 for AIX Privileges

With regard to privileges, the considerations for DB2 for OS/2 also apply to DB2 for AIX. Although UPM is not available in AIX, similar facilities are offered by the operating system itself. Individual AIX user IDs can be assigned to groups, which are valid subject to SQL GRANT and REVOKE statements. Group identifiers are not valid specifications for connections to DB2 for AIX. A user ID and a group can share the same name.

The internal process of checking authorizations and the privileges needed to bind and execute packages are the same for both platforms.

6.4.6 Session Level Security

The DB2 for AIX AS supports the use of session level security. To implement session level security with a specific remote LU select **yes** in the `Session security supported?` field in the LU 6.2 Partner LU profile and create a corresponding LU 6.2 Session Security profile.

To implement session level security in VTAM, specify VERIFY=REQUIRED in the APPL statement.

6.5 Accessing the Databases

6.5.1 SNA Resources

Before connecting to the database you should check if the SNA system is ready to support the communication between the database client on the AIX system and the SQL/DS database server. You can use SMIT to start, stop and display the SNA resources. To display SNA resources start SMIT and select:

1. Communications Applications and Services
2. SNA Server/6000
3. Manage SNA Resources
4. Display SNA Resources

If you ask for the status of SNA you hopefully will get this screen.

```
                                 COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

Subsystem         Group            PID     Status
 sna              sna              6411    active
```

Figure 188. Display the Status of SNA

Also have a look which link is active and which sessions are available. Using SMIT you will see this screen with the active link information.

```
                                 COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

    Link           Adjacent      Node    Device              # of local  In
   station         CP name       type     name      State     sessions   use
-------------- ----------------- ----- --------- ---------- ---------- -----
@tok0.4                                tok0      Starting            0 No
IPFL3174      DEIBMIPF.IPFCP200 NN     tok0      Active              2 Yes


F1=Help            F2=Refresh         F3=Cancel          F6=Command
F8=Image           F9=Shell           F10=Exit
```

Figure 189. Display Active Link Information

If you use SMIT to show the short report of the LU 6.2 Session Information you will see the following screen.

```
                          COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

           Local            Partner        Mode      Link
 CGID     LU name           LU name        name     station     State
------ ----------------- ----------------- -------- ----------- ----------
2      DEIBMIPF.IPFP221B DEIBMIPF.IPFCP200 CPSVCMG  IPFL3174    Available
1      DEIBMIPF.IPFP221B DEIBMIPF.IPFCP200 CPSVCMG  IPFL3174    Available


F1=Help            F2=Refresh          F3=Cancel          F6=Command
F8=Image           F9=Shell            F10=Exit
```

Figure 190. Display LU 6.2 Session Information

For more information see appendix B.3, “Database Connection from ITSO1 to SQL/DS on VM/VSE” on page 239.

6.5.2 Connection to Databases

We now want to connect from ITSO1 client to the SQL/DS database on VSE. On ITSO1 we use code page IBM-850 and on SQL/DS we use CCSID 500.

```
itso1(db2inst1) $ db2 † connect to ipfvs1a4 user demo using sql34n†

   Database Connection Information

 Database product       = SQL/DS VSE 3.5.0
 SQL authorization ID   = DEMO
 Local database alias   = IPFVS1A4

itso1(db2inst1) $
```

Figure 191. Connect with Corresponding CCSIDs

Now we change the CCSID for the SQL/DS database on VSE from 500 to 870. Then we try again to connect from the ITSO1 client to the SQL/DS database on VSE. We get an error message because there is no automatic transformation from CCSID 870 to code page 850 and vice versa.

```
itso1(db2inst1) $ db2 † connect to ipfvs1a4 user demo using sql34n†

SQL0332N  There is no available conversion for the source code page †870† to
the target code page †850†.  Reason Code †1†.  SQLSTATE=57017

itso1(db2inst1) $
```

Figure 192. Connect with not Corresponding CCSIDs

One of the solutions for connecting with an incompatible CCSIDs is as following:

In order to connect to a data base with an incompatible code page, the following steps need to be performed. Although this information is correct at this time (Nov '96), you should always check your DB2 manuals first, to see if there is more up-to-date information applicable to the same topic.

1. Make sure you are using DB2 server V2.1.2 or above.
2. Ask your IBM representative to obtain two conversion files for the pair of code pages that you want to support. For example, if you want to support connection between code page 819 and code page 915, you need 08190915.cnv and 09150819.cnv tables.
3. Place these binary files in the *<DB2Path>/sqllib/conv* directory on UNIX platforms such as AIX, and in the *<DB2Path>/sqllib/conv* directory on various Intel platforms such as OS/2 or Windows.
4. If you are only using Common Server DB2 C/S products, you need to place these files only on the server.
5. For connections between Common Server DB2 C/S and mainframe DB2 implementations, such as SQL/DS or DB2/MVS, you need to first ensure that the corresponding DB2 host platform supports the same conversion. And you only need the Host-to-Workstation conversion table on the workstation side.
6. Once the connection between incompatible code page environments is established, you will get a warning message:

```
SQL0863W  A successful connection was made, but only single byte
characters should be used.  SQLSTATE=0153.
```

Figure 193. Connection Information

This is normal, and intended to warn you that you should only use the characters that exist in both code pages.

Chapter 7. Multi-Language Solutions

This chapter focuses on the ZPIZ side, and gives information about the connectivity parameters used at ZPIZ and ITSO, and related installation and customization steps.

7.1 ITSO - ZPIZ Multi-Language Solutions

7.1.1 ZPIZ DRDA Objectives

We wanted to set up a prototyping environment which would be based on our existing I/T infrastructure and would use the existing systems and applications that can be integrated with each other. In addition, it must facilitate the evolution of information systems while protecting our investments.

Short term, it must enable us to establish a DRDA connectivity environment and provide a framework for enabling new types of applications.

Long term, the prototype must ensure that the quality of data, data integrity, and security remains on the same level as today, if not higher.

The prototyping must establish an environment that is manageable by our existing personnel on existing equipment and must point out all the possible unsolved issues and expected difficulties.

7.1.2 ZPIZ DRDA Environment

Figure 195 on page 180 illustrates the main software and hardware components involved in DRDA connectivity at ZPIZ. Some of them are running on our production environment, the others are installed and implemented on a test system.

VM/ESA 1.2.2 system is installed as a second level system and is connected to the first level VM/ESA 1.1.1 via Virtual Channel-to-Channel Adapter (VCTCA).

The SQL/DS 3.5.0 for VM database runs on a VM/ESA second level system.

SQL/DS 3.5.0 for VSE runs on VSE/ESA 1.3.5 which is connected via VCTCA to the VM/ESA first level system.

VM/VTAM on the VM/ESA first level system is connected via 3745 to the Token-Ring and other SNA networks, such as IGN, and via VCTCA to the other VTAM systems in our environment.

7.1.3 ZPIZ Installation and Customization Steps

For the purpose of establishing the ZPIZ DRDA environment we have implemented the following hardware and software environment.

The current production system runs VM/ESA 1.1.1 on an IBM 9672-R22 in basic mode (no LPAR). Under VM/ESA 1.1.1, there are three VSE/ESA guest operating systems running, all in a full production environment.

In order to influence our existing production environment as little as possible, and considering that our processor is not CPU or storage constrained, we have installed and implemented the following additional environment:

1. **Install and implement VM/ESA 1.2.2 second level**

   The second level virtual machine was called VM122.

2. **Implement AVS/VM environment**

   AVSVM virtual machine is running under second level VM122.

3. **Install and implement a new CICS/VSE region**

   We installed a new CICS/VSE on one of the existing VSE/ESA guest operating systems and named it Z00CICS4. For the VSE system, we have chosen the VSE3 virtual machine, which was already running under VM/ESA first level.

   The new CICS region must have the capability of establishing LU 6.2 sessions, and with the ability to connect to our SQLDS application server.

4. **Customize VM/VTAM first level, VM/VTAM second level and VSE/VTAM.**

5. **Install and implement the most current SQL/DS 3.5.0 for VM**

   This SQL was installed on the VM/ESA 1.2.2 second level system. We have prepared the SQLMACH virtual machine as an application server and defined a sample SQL database, called SQLDBA. The second user, which was defined for the purpose of acting as an application requester, was named SQLUSER. The SQLUSER was granted SQL DBA authority.

6. **Install and implement the most current SQL/DS 3.5.0 for VSE**

   This SQL was installed on the VSE/ESA 1.3.5 system. We have defined a sample SQL database, called SQLDS.

7. **Connect to IBM Global Network (IGN) and ITSO Böblingen**

   This was the most time consuming job, especially as we were connecting to the IGN for the first time.

8. **APPC connection to ITSO**

9. **Customize ACF/NCP**

   Some special APPC considerations were taken into account when changing NCP parameters.

10. **Install and customize DB2/2 V2.1.1**

    We have installed a single-user version of DB2/2 under OS/2 Warp.

    There are some special considerations that have to be taken into account before trying to install DB2/2 V2.1. These considerations apply to the workstations customized to use the Slovenian code page only.

11. **Install and customize DDCS/2 V2.3**

    We have installed a single-user version of DDCS/2 under OS/2 Warp.

12. **If need to support an incompatible code page conversions**

    The following steps need to be performed. Although this information is correct at this time (Nov ′96), you should always check your DB2 manuals first, to see if there is more up-to-date information applicable to the same topic.

a. Make sure you are using DB2 server V2.1.2 or above.

b. Ask your IBM representative to obtain two conversion files for the pair of code pages that you want to support. For example, if you want to support connection between code page 819 and code page 915, you need 08190915.cnv and 09150819.cnv tables.

c. Place these binary files in the *<DB2Path>/sqllib/conv* directory on UNIX platforms such as AIX, and in the *<DB2Path>/sqllib/conv* directory on various Intel platforms such as OS/2 or Windows.

d. If you are only using Common Server DB2 C/S products, you need to place these files only on the server.

e. For connections between Common Server DB2 C/S and mainframe DB2 implementations, such as SQL/DS or DB2/MVS, you need to first ensure that the corresponding DB2 host platform supports the same conversion. You only need the Host-to-Workstation conversion table on the workstation side.

f. Once the connection between incompatible code page environments is established, you will get a warning message:

```
SQL0863W  A successful connection was made, but only single byte
characters should be used.  SQLSTATE=0153.
```

Figure 194. Connection Information

This is normal, and intended to warn you that you should only use the characters that exist in both code pages.

7.1.4 Schematic View of DRDA Environment at ZPIZ

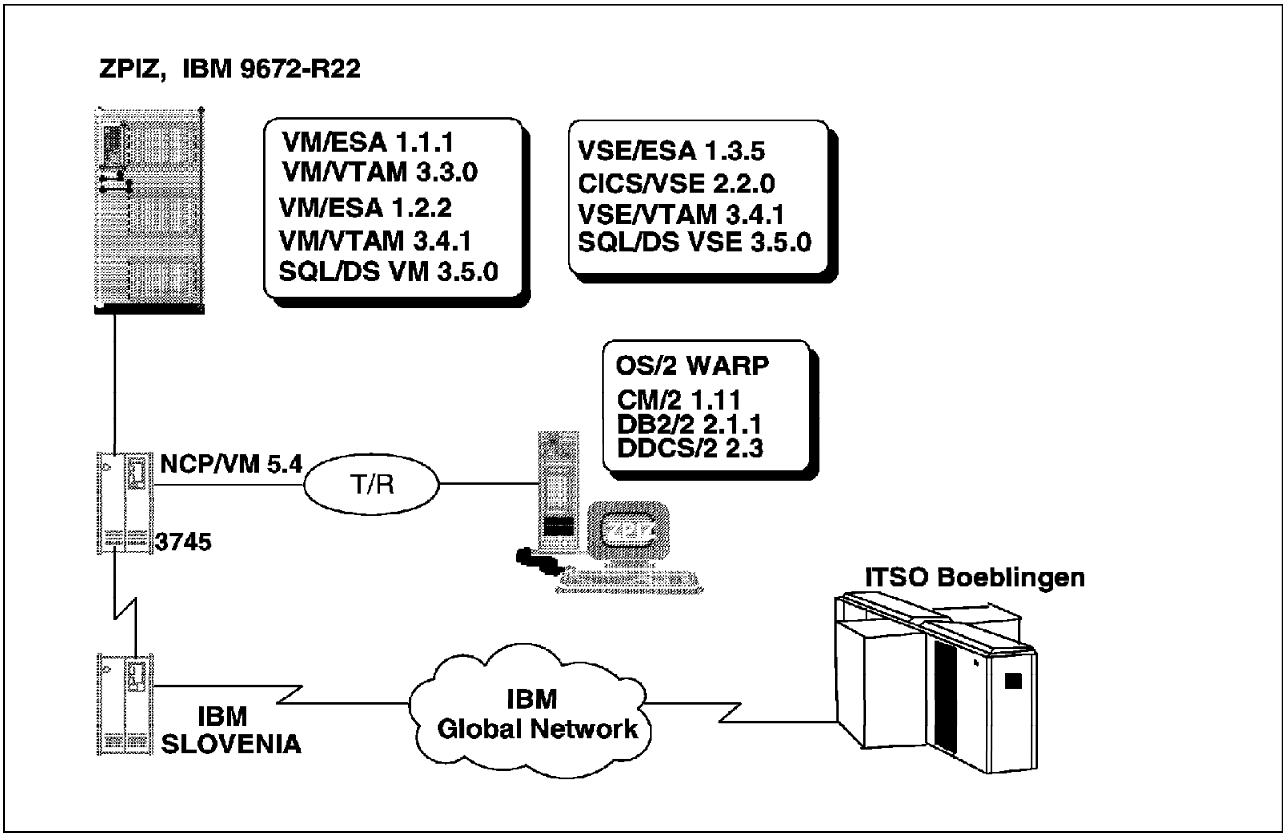


Figure 195. ZPIZ DRDA Hardware and Software

7.1.5 ZPIZ DRDA Connections

Figure 196 shows the overall connectivity scheme from the ZPIZ point of view.

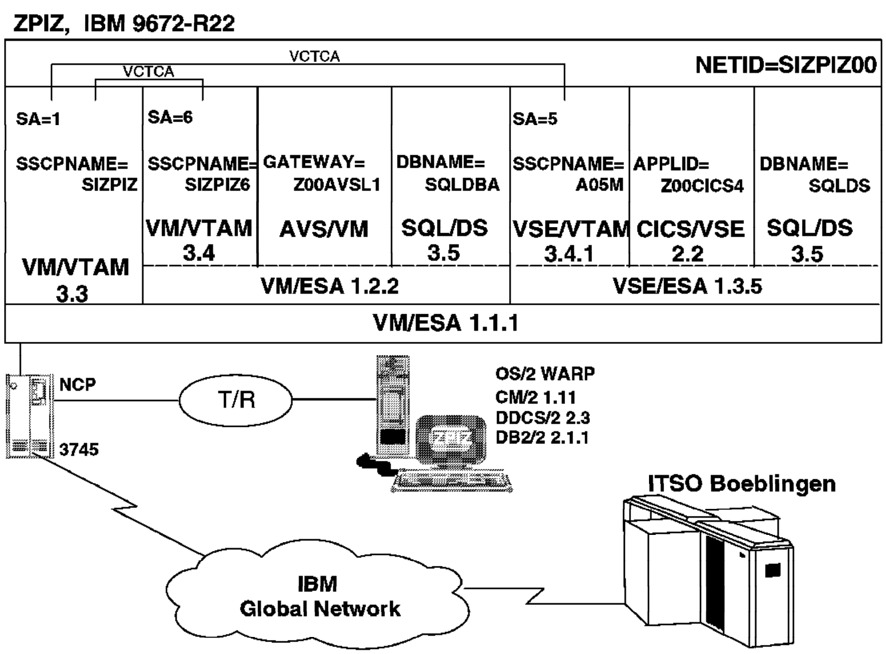


Figure 196. DRDA Configuration at ZPIZ

From the ZPIZ side, we will be connected as application requesters to the following application servers:

Table 19. ZPIZ DRDA Connections. ZPIZ application requester connection possibilities

ZPIZ AR	ZPIZ LU	Partner LU	Partner AS
ZPIZ VM	-	-	ZPIZ VM
	Z00AVSL2	Z00CICS4	ZPIZ VSE
	Z00AVSL1	IPFA2GL4	ITSO VM
	Z00AVSL1	IPFA21CD	ITSO VSE
ZPIZ OS/2	Z00T010A	Z00CICS4	ZPIZ VSE
	Z00T010A	Z00AVSL1	ZPIZ VM
	Z00T010A	IPFA2GL4	ITSO VM
	Z00T010A	IPFA21CD	ITSO VSE

ZPIZ VM application requester will be connected to:

- Local AS running on VM and VSE platforms
- Remote AS running on VM and VSE platforms in ITSO Böblingen

ZPIZ OS/2 application requester will be connected to:

- Local AS running on VM and VSE platforms
- Remote AS running on VM and VSE platforms in ITSO Böblingen

7.1.6 ZPIZ Connectivity Parameters

Before setting up the communications between the local and partner systems we have collected the SNA and DRDA parameters in a table as shown below.

Table 20. ZPIZ SNA and DRDA Connectivity Parameters

Parameter	DB2/2 ZPIZ	SQL/DS VM ZPIZ	SQL/DS VSE ZPIZ	SQL/DS VM Böblingen	SQL/DS VSE Böblingen
Node	DB2	VMNODE	VSENODE	BOBVM	BOBVSE
Comm.type	APPC	APPC	APPC	APPC	APPC
Symbolic dest.name	-	VM122AVS	VSE135B	BOBAVS	BOBCICS
Security	PGM	SAME	PGM	PGM	PGM
Authentication	DCS	DCS	DCS	DCS	DCS
RDB_NAME	Z00DB200	SQLDBA	SQLDS	SQLVMDB0	SQLVSDB1
CP	-	SIZPIZ6	A05M	IPFV2	IPFV2A
LU name	Z00T010A	Z00AVSL1	Z00CICS4	IPFA2GL4	IPFA21CD
LU Alias	Z00T01	VM122AVS	VSE135	BOBAVS	BOBCICS
TPN	DB2DUMMY	SQLDBA	TPN1	SQLVMDB0	TPN1
NETID	SIZPIZ00	SIZPIZ00	SIZPIZ00	DEIBMIPF	DEIBMIPF
Modeent	IBMRDB	IBMRDB	IBMRDB	IBMRDBM	IBMRDB

Chapter 8. REXX SQL Application Enabling and Usage

8.1 Application Enabling and Usage

The intent of this chapter is to focus on specific OS/2 REXX application capabilities and interfaces to the DB2 family of products. Specifically, the interfaces to remote databases residing on VM/ESA and VSE/ESA were tested.

The REXX examples that follow assume that the DRDA environment was already established.

This chapter is based on the IBM publication *GG24-4199 OS/2 REXX From Bark to Byte*, which concentrates on the OS/2 REXX interface to other OS/2 based products, such as DB2/2, CM/2 and MMPM/2.

8.1.1 OS/2 REXX Interface to DB2 Family of Products

OS/2 REXX has the capability of using dynamic link libraries (DLL) to interface with other software applications, for example DB2/2. DB2/2 provide application programming interfaces (APIs) that allow REXX programs to interface with them.

DB2/2 provides an interface to REXX programs in three basic ways:

1. The SQLDBS DB2/2 API allows REXX programs to invoke command like versions of DB2/2′s API set
2. The SQLEXEC DB2/2 API allows REXX programs to invoke SQL statements.
3. DB2/2 provides data structures that are accessible by REXX programs

We will discuss some of these interfaces and how they can be used to create useful REXX programs. Example programs are provided and explained.

If coded properly, the same REXX program can be used to access and manipulate databases on either the local or remote site. The remote database can reside on any platform that is connected to the local workstation using DRDA protocol. That is, it is transparent to the program which database is being accessed if cataloged, DB2/2 will find the database′s location.

8.1.1.1 Presentation Manager Interface to REXX

Some Presentation Manager (PM) features can be included in the REXX program by using PMREXX. PMREXX is a windowed Presentation Manager application, included with OS/2, that enables the REXX program to input data, and to browse the REXX output in a PM window.

8.1.1.2 Invocation of REXX Programs

The REXX programs can be started either by typing the REXX program name in an OS/2 window, or by typing ***PMREXX rexx_prog_name***, which will start the program using PMREXX.

8.1.1.3 Register DB2/2 Functions to REXX

The DB2/2 APIs provided by REXX are external function packages. They must be registered by the REXX program by using RxFuncAdd. Just like other external function packages, once registered these functions are available to all REXX programs running on the system until they are dropped. When they are dropped, all REXX programs running on the system lose access to these functions. Therefore, we recommend not to drop these functions at the end of the programs.

8.1.1.4 Error Handling

DB2/2 provides information in the SQLCA data structure that is useful in return code checking and error determination. SQLCA.SQLCODE is filled with a return code value after every call to the SQLEXEC and SQLDBS APIs. A way to get a more detailed explanation of the SQL error is to bring up the DB2/2 online documentation. An example of this is provided in Figure 197, which is taken from SQLERR.CMD.

```
/*
     SQLERR

     This routine is called when there is an SQL error.
     The view command is used to bring up the DBMSG.INF
     information for the error.

     Input arguments:
          SQLCODE    - SQL error code    (SQLCA.SQLCODE)
          TABLE NAME - database table name
*/
arg sqlcode
sqlcode = STRIP(sqlcode,¢L¢,¢-¢)                /* remove a sign and */
                                            /* pad with zeros on the left */
If Length(sqlcode) < 4 then sqlcode = RIGHT(sqlcode,4,¢0¢)
¢START /PM VIEW DBMSG.INF SQL¢]]sqlcode
return
```

Figure 197. SQL Error Handling

8.1.1.5 REXX SQL Statement Type

There are two types of SQL statements, static and dynamic. Static SQL statements are statements that are prepared prior to the execution of the program that contains them. The complete SQL statement must be known prior to the compilation of the program. During compilation, an executable form of the SQL statement is created. Having said that, since OS/2 REXX is an interpreted language and not a compiled language, static SQL statements are not possible in REXX. All SQL statements in REXX programs are prepared when the program runs. The term used for this type of statement is dynamic SQL statement.

Both static and dynamic SQL statements have advantages. Static SQL statements often process faster than dynamic SQL statements because the overhead of preparing the statement is done before the program actually runs. However, dynamic statements offer more flexibility because the actual form of the SQL statement does not need to be known before the program runs. For example, if you want to use the SELECT statement to query a table and load the results into variables, in a static SQL situation you would need to know the number of columns in the table and their names before your program is compiled. If there are changes to the table, chances are your program would have to be recompiled. Using a dynamic SQL statement in this situation, the number of columns, their names, lengths, and data types do not need to be known in advance. They can be obtained during the execution of a program.

8.1.1.6 SQL SELECT Example

The steps used to code the SELECT type of program:

1. CONNECT to the database
2. Create the SELECT statement
3. Use the DECLARE statement to define a cursor and associate the cursor to the SELECT statement
4. Use the PREPARE statement with the INTO clause to dynamically build the SELECT statement and specify a REXX variable into which column information from the SQLDA will be loaded
5. Use the DESCRIBE statement to load column information from the SQLDA into a REXX array variable
6. Use the OPEN statement to initialize the cursor pointing to the first row
7. Use the FETCH statement with the USING DESCRIPTOR in a loop to retrieve rows into REXX array variable
8. Use the CLOSE statement to release the cursor
9. Use the COMMIT statement to complete the unit of work

```
/*
     SELECT.CMD

     This routine prompts user to select a database and then a table
     from the database. A dynamic SQL query is run on the table
     and the results are loaded into a flat file called TABLE.DAT.
*/
¢@echo off¢
  /* load SQLDBS and SQLEXEC external function packages if necessary */
if Rxfuncquery(¢SQLDBS¢) <> 0 then do
  rc = RxFuncAdd(¢SQLDBS¢ , ¢SQLAR¢ , ¢SQLDBS¢)
  if RC \= 0 then do
    say †Error registering SQLDBS, rc = † rc
    return
  end  /* if RC */
end
if Rxfuncquery(¢SQLEXEC¢) <> 0 then do
  rc = RxFuncAdd(¢SQLEXEC¢, ¢SQLAR¢, ¢SQLEXEC¢)
  if RC \= 0 then do
    say †Error registering SQLEXEC, rc = † rc
    return
  end  /* if RC */
end
                                     /* disconnect from any database */
call sqlexec ¢CONNECT RESET¢;           /* -1024 means not connected */
 if ( SQLCA.SQLCODE <> 0 & SQLCA.SQLCODE <> -1024) then do
  call sqlerr SQLCA.SQLCODE
   say ¢CONNECT RESET Error, SQLCODE = ¢ SQLCA.SQLCODE;
  return
 end

say ¢Enter DBNAME, USERID and PASSWORD:¢;
pull dbname userid password;
                                     /* connect to a remote database */
call sqlexec ¢CONNECT TO ¢ dbname ¢USER ¢ userid ¢USING ¢ password;
 if ( SQLCA.SQLCODE <> 0) then do
   call sqlerr SQLCA.SQLCODE
   say ¢CONNECT Error, SQLCODE = ¢ SQLCA.SQLCODE;
   return
 end
                                    /* prepare dynamic SQL statement */
say ¢Enter table name:¢;            /* ask the user for a table name */
pull table;
st1 = †SELECT * FROM † table                     /* select statement */
call sqlexec ¢DECLARE c1 CURSOR FOR s1¢            /* declare cursor */
 if ( SQLCA.SQLCODE <> 0) then do
   call sqlerr SQLCA.SQLCODE
   say ¢DECLARE Error, SQLCODE = ¢ SQLCA.SQLCODE;
   return
 end
```

Figure 198 (Part 1 of 3). SQL SELECT Example

```
                                                       /* prepare statement */
call sqlexec ¢PREPARE s1 INTO :sqlda FROM :st1¢
 if ( SQLCA.SQLCODE <> 0) then do
   call sqlerr SQLCA.SQLCODE
   say ¢PREPARE Error, SQLCODE = ¢ SQLCA.SQLCODE;
   return
 end
                                                      /* describe statement */
call sqlexec ¢DESCRIBE s1 INTO :sqlda¢
 if ( SQLCA.SQLCODE <> 0) then do
   call sqlerr SQLCA.SQLCODE
   say ¢DESCRIBE Error, SQLCODE = ¢ SQLCA.SQLCODE;
   return
 end
                                                           /* open cursor */
call sqlexec ¢OPEN c1¢
 if ( SQLCA.SQLCODE <> 0) then do
   call sqlerr SQLCA.SQLCODE
   say ¢OPEN Error: SQLCODE = ¢ SQLCA.SQLCODE;
   return
 end
                                                             /* fetch row */
call sqlexec ¢FETCH c1 USING DESCRIPTOR :sqlda¢
 if ( SQLCA.SQLCODE <> 0) then do
   call sqlerr SQLCA.SQLCODE
   say ¢OPEN Error: SQLCODE = ¢ SQLCA.SQLCODE;
   return
 end

                                                    /* prepare output file */
  ¢@ERASE TABLE.DAT 2>nul:¢  /* erase TABLE.DAT if it already exists */
  header = ¢Query results from DB2/2 to database ¢ dbname ¢ , table ¢ table
  blankheader = ††
  dateheader = date(¢o¢);
  timeheader = time();
  dtheader = dateheader timeheader;
                              /* write header records to TABLE.DAT */
  call LINEOUT ¢TABLE.DAT¢,blankheader
  call LINEOUT ¢TABLE.DAT¢,dtheader;
  call LINEOUT ¢TABLE.DAT¢,header
  call LINEOUT ¢TABLE.DAT¢,blankheader
do while (SQLCA.SQLCODE = 0)                       /* loop through rows */
  outrec = ¢¢                                   /* retrieved with SELECT */
  do col = 1 to sqlda.sqld                  /* loop through columns and */
                                                  /* build output record */
     if  (sqlda.col.sqlind) = -1  then             /* is it a null field */
        outrec = outrec ¢-¢
     else
     outrec = outrec sqlda.col.sqldata
  end /* do col */
  call LINEOUT ¢TABLE.DAT¢,outrec                /* write output record */
  call sqlexec ¢FETCH c1 USING DESCRIPTOR :sqlda¢    /* get next row */
end /* do while */
```

Figure 198 (Part 2 of 3). SQL SELECT Example

```
call LINEOUT ¢TABLE.DAT¢                                  /* close file */
                         /* clean up - close cursor and commit work */
call sqlexec ¢CLOSE c1¢;
 if ( SQLCA.SQLCODE <> 0) then do
  call sqlerr SQLCA.SQLCODE
   say ¢CLOSE Error, SQLCODE = ¢ SQLCA.SQLCODE;
  return
 end

call sqlexec ¢COMMIT¢
 if ( SQLCA.SQLCODE <> 0) then do
  call sqlerr SQLCA.SQLCODE
   say ¢COMMIT Error, SQLCODE = ¢ SQLCA.SQLCODE;
  return
 end
                                        /* disconnect from database */
call sqlexec ¢CONNECT RESET¢;
 if ( SQLCA.SQLCODE <> 0) then do
  call sqlerr SQLCA.SQLCODE
   say ¢CONNECT RESET Error, SQLCODE = ¢ SQLCA.SQLCODE;
  return
 end
       /* open E editor with TABLE.DAT, which contains query results */
¢START /PM E TABLE.DAT¢

return
```

Figure 198 (Part 3 of 3). SQL SELECT Example

Notes:

1. The program uses the cursor to traverse through the rows. The cursor points to the row currently being processed, beginning with the first row. The cursor is incremented to point to the next row by DB2/2 automatically when you request the next row using the FETCH statement.
2. The PREPARE statement validates the SELECT statement and prepares it for execution. It names the REXX variable into which the SQLDA information will be loaded.
3. The DESCRIBE statement loads column information into the REXX array variable.
4. The OPEN statement initializes the cursor to point to the first row that satisfies the SELECT conditions.
5. The FETCH statement with the USING DESCRIPTOR clause loads the row data into the REXX array variable.

8.1.1.7 SQL INSERT Example

```
                                      /* prompt user to enter values */
fields=¢¢
values=¢¢
do i = 1 to sqlda.sqld
  say ¢Enter  a ¢ sqlda.i.sqllen ¢ character ¢ sqlda.i.sqlname
  pull data
  if (data = ¢¢) & (sqlda.i.sqltype //2 = 0) then  do
    say ¢No nulls allowed in this field. Please enter a ¢ sqlda.i.sqllen,
        ¢ character ¢ sqlda.i.sqlname
    pull data
    if data = ¢¢  then do
       say ¢leaving program - user error¢
       return
    end  /* if data */
  end /* if data */
  fields = fields]]sqlda.i.sqlname]]¢,¢  /* build column name string */
  values = values]]†¢†]]data]]†¢,†     /* build column values string */
end /* do i = 1 */

fields = STRIP(fields,t,¢,¢)                   /* remove last comma */
values = STRIP(values,t,¢,¢)

st1 = †INSERT INTO † table †(†fields†) VALUES (†values†)†
call SQLEXEC ¢EXECUTE IMMEDIATE :st1¢
 if ( SQLCA.SQLCODE <> 0) then do
   call sqlerr SQLCA.SQLCODE
   say ¢UPDATE Error, SQLCODE = ¢ SQLCA.SQLCODE;
   return
 end
 else
   call SQLEXEC ¢COMMIT¢
 if ( SQLCA.SQLCODE <> 0) then do
   call sqlerr SQLCA.SQLCODE
   say ¢COMMIT Error, SQLCODE = ¢ SQLCA.SQLCODE;
   return
 end
```

Figure 199. SQL INSERT Example

In an application that is adding rows to a table, the techniques discussed in the previous section on SELECT can be very helpful. The example in Figure 199 shows the usage of the SQL INSERT statement.

Notes: In the figure above only a section of a program is shown.

- the SQLDA.SQLD contains the number of columns in the row.
- the SQLDA.SQLLEN contains the length of a column.

Appendix A. System Tables and Sample Programs

A.1 VM/VTAM Definitions - BOEVMIS2

```
ATCCON02 VTAMLST

         PF2PATH,              ** PATH TABLES                        ** C
         PFNV5,                ** NCP FOR 3725                       ** C
         PFNV3,                ** NCP FOR 3745                       ** C
         PF2VSCS,              ** HOST APPLICATIONS FOR VM ONLY      ** C
         PF2AVS,               ** AVS                                ** C
         PF2CDRM,              ** CDRMS:                             ** C
         PF2CDRS,              ** RESOURCES:                         ** C
         DRSDRDA,              ** DRDA RESUORCES:                    ** C
         PF2SNA,               ** LOCAL SNA, 3174-11L                ** C
         PF2LOC,               ** LOCAL NON SNA, 4361-WSA            ** C
         PF2CTCA,              ** VCTCA TO 2ND LVL VSE/ESA 1.3       ** C
         VMG,                  ** PVM GATEWAY DEFINITIONS            ** C
         PF2SSCP,              ** ADJACENT SSCP FOR DYNAMIC          ** C
         PNSAPPL,              ** TPNS APPLS                         ** C
         PNSDM,                ** TPNS DISPLAY MONITOR APPL          ** C
         PNSCDRM,              ** TPNS CROSS DOMAIN RESOURCE MAN.    ** C
         PNSCDRS,              ** TPNS CROSS DOMAIN RESOURCES        ** C
         390LAN,               ** P390 GROUP                         ** C
         390CDRM,              ** P390 CROSS DOMAIN RESOURCE MANAGER** C
         390CDRS,              ** P390 CROSS DOMAIN RESOURCE         ** C
         390AVS,               ** P390 GATEWAY LU                    ** C
         PF2RSCS,              ** RSCS APPLICATION                   ** C
         PFA2NP,               ** NPM APPLICATION MAJOR NODE         ** C
         PFA2NV,               ** NETVIEW APPLICATION MAJOR NODE     ** C
         DRSSLOV,              ** ZPIZ SLOVENIA CONNECTIVITY         ** C
         PF2SWI0               ** SWITCHED MAJOR NODE VIA 3745       **
         RAPHOPT BLOCK   NAME IPF*
         RAPHOPT FORWARD NAME IPFA2NV
         RAPHOPT FORWARD NAME IPFCP200
         RAPHOPT FORWARD NAME IPFA21CD
         RAPHOPT FORWARD NAME IPF2AGL4
```

Figure 200. ATCCON02 VTAMLST

```
ATCSTR02 VTAMLST

        *  DEFAULT BUFFER SIZE ARE USED
        *APBUF=(16,60,2,,1,3)
        *BSBUF=(28,142,0,,1,27)
        *LFBUF=(2,80,0,,1,1)
        *LPBUF=(12,1344,0,,1,6)
        *SFBUF=(51,72,0,,1,1)
        *SPBUF=(2,112,0,,1,1)
        *WPBUF=(19,208,0,,1,19)
        *XDBUF=(6,665,0,,1,5)
        *CRPLBUF=(60,116,0,,1,60)
        *
        IOBUF=(200,384,40,,10,80,0),                               C
        CSALIMIT=0,                                                C
        CDRSCTI=480,                                               C
        CONFIG=02,                                                 C
        DLRTCB=32,                                                 C
        DYNASSCP=NO,                                               C
        GWSSCP=NO,                                                 C
        HOSTPU=IPFVM2,                                             C
        HOSTSA=02,                                                 C
        IOINT=180,                                                 C
        MAXSUBA=255,                                               C
        MSGMOD=NO,                                                 C
        NETID=DEIBMIPF,                                            C
        NMVTLOG=NEVER,                                             C
        NMVTLOG=NEVER,                                             C
        PPOLOG=NO,                                                 C
        SDLCMDRS=NO,                                               C
        SSCPDYN=YES,                                               C
        SSCPID=02,                                                 C
        SSCPNAME=IPFV2,                                            C
        SSCPORD=PRIORITY,                                          C
        SUPP=NOSUP,                                                C
        TNSTAT,                                                    C
        TIME=1440,                                                 C
        NOTRACE,TYPE=VTAM,                                         C
        USSTAB=ISTINCNO,                                           C
        XNETALS=NO
```

Figure 201. ATCSTR02 VTAMLST

```
IPF2CDRM VTAMLST

        VBUILD TYPE=CDRM
IPF     CDRM   SUBAREA=01,CDRSC=OPT   ** SA=01 BOEVMISC
*              STATOPT=¢BOEVMISC¢
IPFV2   CDRM   SUBAREA=02,CDRDYN=YES  ** SA=02 BOEVMIS2
*              STATOPT=¢BOEVMIS2¢
IPFV2A  CDRM   SUBAREA=21,CDRSC=OPT   ** SA=21 VSEANL13
*              STATOPT=¢VSEANL13¢
IPFV2B  CDRM   SUBAREA=22,CDRSC=OPT   ** SA=22 V123A80K
```

Figure 202. IPF2CDRM VTAMLST

```
 IPF2CDRS VTAMLST

          VBUILD TYPE=CDRSC
WCVM122  CDRSC  CDRM=SSCP04       ** SA=04 BOEVMIS1 WCVM122
RSCS122  CDRSC  CDRM=SSCP04       ** SA=04 BOEVMIS1 RSCS
IPFA1RSC CDRSC  CDRM=IPF          ** SA=01 BOEVMISC
IPFA1VSC CDRSC  CDRM=IPF          ** SA=01 BOEVMISC
IPFA21CD CDRSC  CDRM=IPFV2A       ** SA=21 VSEANL13
*               STATOPT=¢VSEANL13 CICS¢
IPFAN    CDRSC  CDRM=IPFV2        ** SA=2  IPFA2NV (NETVIEW)
*               STATOPT=¢MANFREDS PWS¢
IPFCL0E0 CDRSC  ALSLIST=(IPFCP20E,IPFCP200),                           C
                ALSREQ=YES,                                            C
                DLOGMOD=IBMRDBM,                                       C
                MODETAB=DRDAMOD
IPFBOEDJ CDRSC  ALSLIST=(IPFP211B,IPFCP200),                           C
                ALSREQ=YES,                                            C
                DLOGMOD=IBMRDBM,                                       C
                MODETAB=DRDAMOD
```

Figure 203. IPF2CDRS VTAMLST

```
 IPF2SNA  VTAMLST

          VBUILD TYPE=LOCAL
*----------------------------------------------------------------------
*         DEFINE THE GATEWAY CONTROLLER PU
*----------------------------------------------------------------------
IPFCP200 PU     CUADDR=200,                                            C
                DELAY=0.2,                                             C
                ISTATUS=ACTIVE,                                        C
                MAXBFRU=29,                                            C
                PUTYPE=2,                                              C
                XID=YES,                                               C
                DYNLU=YES,                                             C
                USSTAB=ISTSNA,MODETAB=ISTINCLM,DLOGMOD=D4A32782
*----------------------------------------------------------------------
*         DEFINE THE DOWNSTREAM PUS ATTACHED TO THE GATEWAY
*----------------------------------------------------------------------
IPFCP20E PU     CUADDR=20E,                                            C
                DELAY=0.2,                                             C
                ISTATUS=ACTIVE,                                        C
                MAXBFRU=29,                                            C
                PUTYPE=2,                                              C
                XID=NO,                                                C
                USSTAB=ISTSNA,MODETAB=AMODETAB,DLOGMOD=MSDLCQ,         C
                PACING=1,VPACING=1,                                    C
```

Figure 204. IPF2SNA VTAMLST

```
 IPF2SSCP VTAMLST

         VBUILD TYPE=ADJSSCP
*
         NETWORK
IPF      ADJCDRM
*
DEIBMIPF NETWORK NETID=DEIBMIPF
*
IPF      ADJCDRM
*               STATOPT=¢BOEVMISC¢
IPFV2    ADJCDRM
*               STATOPT=¢BOEVMIS2 VTAM¢
IPFVA    ADJCDRM
*               STATOPT=¢BOEVMIS1¢
IPFV2A   ADJCDRM
*               STATOPT=¢VSEANL13 VTAM¢
IPFV2B   ADJCDRM
*               STATOPT=¢V132A80K VTAM¢
```

Figure 205. IPF2SSCP VTAMLST

```
 CDRSSLOV VTAMLST

* Z00AVSL1 .... VM/ESA AR ITSO -------- VM/ESA  AS ZPIZ
* Z00CICS4 .... VM/ESA AR ITSO -------- VSE/ESA AS ZPIZ
         VBUILD TYPE=CDRSC
         NETWORK NETID=SIZPIZ00
Z00AVSL1 CDRSC
Z00CICS4 CDRSC
```

Figure 206. CDRSSLOV VTAMLST

```
 IPF2SWI0 VTAMLST

IPF2SWI0 VBUILD TYPE=SWNET,                                           C
               MAXGRP=5,          NUMBER OF UNIQUE PATH GROUPS        C
               MAXNO=20,          NUMBER OF UNIQUE PHONE NUMBERS
*************************************************************
* PS/2 FOR DRDA
*************************************************************
IPFCPS0E PU    ADDR=0E,           COULD BE ANYTHING (NOT USED)        C
               IDBLK=05D,         IDBLK OF OS/2 COMM. MANAGER         C
               IDNUM=E000E,       IDENTIFICATION NUMBER               C
               DISCNT=NO,         VTAM DOES NOT HANG UP               C
               MAXOUT=1,          MAXIMUM NUBER OF PIUS               C
               MAXDATA=1920,      MAXDATA FOR OS/2 COMM. MANAGER      C
               USSTAB=ISTSNA,                                         C
               MODETAB=AMODETAB,                                      C
               DLOGMOD=MSDLCQ,                                        C
               MAXPATH=1,         NUMBER OF DIAL OUT PATHS TO PU      C
               VPACING=2,                                             C
               PUTYPE=2,                                              C
               CPNAME=CPS0E,                                          C
               ISTATUS=ACTIVE
         PATH  DIALNO=020440001010100E,                               C
               GRPNM=IPFG3L89,    LOGICAL GROUP NAME OF TIC 2         C
               GID=1,                                                 C
               PID=2
IPFSL0E0 LU    LOCADDR=00,MODETAB=DRDAMOD,DLOGMOD=IBMRDBM
```

Figure 207. IPF2SWI0 VTAMLST

```
 IPF2AVS  VTAMLST

AVS run as VTAM application (defined to VM/VTAM)

IPF2AVS  VBUILD TYPE=APPL
IPFA2GL4 APPL  APPC=YES,                                              C
               AUTHEXIT=YES,                                          C
               AUTOSES=20,                                            C
               DSESLIM=200,                                           C
               DMINWNL=100,                                           C
               DMINWNR=100,                                           C
               EAS=3999,                                              C
               MAXPVT=200K,                                           C
               SECACPT=ALREADYV,                                      C
               VERIFY=NONE,                                           C
               VPACING=2,                                             C
               MODETAB=DRDAMOD,                                       C
               DLOGMOD=IBMRDBM,                                       C
               SYNCLVL=CONFIRM,                                       C
               OPERCNOS=ALLOW,                                        C
               PARSESS=YES
*              STATOPT=¢AVSVM LU¢
```

Figure 208. IPF2AVS VTAMLST

A.2 VSE/VTAM Definitions - BOEVMIS2

```
 CATALOG ATCSTR00.B

SSCPID=21,                                                          C
HOSTSA=21,                                                          C
SSCPNAME=IPFV2A,                                                    C
HOSTPU=IPFVM21,                                                     C
NOPROMPT,                                                           C
NETID=DEIBMIPF,                                                     C
MAXSUBA=255,                                                        C
CONFIG=00,                                                          C
DYNLU=YES,                                                          C
IOINT=0,                                                            C
SGALIMIT=0,                                                         C
BSBUF=(28,,,,1),                                                    C
CRPLBUF=(60,,,,1),                                                  C
LFBUF=(70,,,,11),                                                   C
IOBUF=(70,288,,,11),                                                C
LPBUF=(12,,,,6),                                                    C
SFBUF=(20,,,,20),                                                   C
SPBUF=(210,,,,32),                                                  C
XDBUF=(6,,,,1)
```

Figure 209. ATCSTR00.B

```
 CATALOG ATCCON00.B
         APPLDRD,                                                   C
         VTMSNA,                                                    C
         VTMNSNA,                                                   C
         VTMPATH,                                                   C
         VTMCA1,                                                    C
         VTMCA2,                                                    C
         VTMCA3,                                                    C
         VTMCTCA,                                                   C
         CDRMDRDA,                                                  C
         ADJDRDA,                                                   C
         CDRSSLOV,                                                  C
         CDRSSTB3,                                                  C
         EXPVAPPL,                                                  C
         VTMCDRS,                                                   C
         VTMSW1
```

Figure 210. ATCCON00.B

```
 CATALOG ADJDRDA.B

ADJDRDA  VBUILD TYPE=ADJSSCP
         NETWORK
IPF      ADJCDRM
DEIBMIPF NETWORK NETID=DEIBMIPF
IPFV2A   ADJCDRM
IPFV2    ADJCDRM
IPF      ADJCDRM
```

Figure 211. ADJDRDA.B

```
IPFNV3OP OPTIONS NEWDEFN=(YES,ECHO)

**********************************************************************
*    NEWNAME = IPFNV3        *                                       *
******************************                                       *
*                            *           SUBAREA = 03                *
* ACF/NCP FOR 3745-170       *           10/10/94  BY WILLEM         *
*                            *           MM/DD/YY                    *
**********************************************************************
*       PCCU¢S STATEMENT SPECIFICATIONS (V6R2M0)                     *
**********************************************************************
*                                                                    *
*       PCCU SPECIFICATIONS - ACF/VTAM  BOEVMISC   3745 (SUBAREA 01) *
*                                                                    *
**********************************************************************
IPFPCC   PCCU AUTODMP=NO,          AUTOMATICALLY TAKE DUMP            C
              AUTOIPL=NO,          LOAD NCP AFTER FAILURE OR DUMP     C
              AUTOSYN=YES,                                            C
              BACKUP=NO,           NO BACKUP HOST                     C
              CHANCON=COND,        DEFAULT                            C
              CUADDR=560,          CHANNEL-ATTACHMENT ADDRESS         C
              DELAY=.2,                                               C
              DUMPDS=NCPDUMP,      DUMP DATA SET                      C
              DUMPLD=NO,           DEFAULT                            C
              GWCTL=SHR,           DEFAULT                            C
              LOADFROM=HOST,       DEFAULT                            C
              MAXDATA=7000,        MAXIMUM PIU SIZE                   C
              NETID=DEIBMIPF,      SNI ONLY                           C
              OWNER=IPFHOS,                                           C
              SAVEMOD=NO,          DEFAULT                            C
              SUBAREA=01,          HOST SUBAREA ADDRESS               C
              TGN=1,               DEFAULT                            C
              VFYC=IGNORE,                                            C
              VFYLM=YES            VERIFY NCP LOAD MODULE NAME
```

Figure 212. IPFNV3

```
 CATALOG  CDRMDRDA.B

CDRMDRDA VBUILD TYPE=CDRM
         NETWORK NETID=DEIBMIPF
IPFV2A   CDRM  SUBAREA=21,CDRDYN=YES,CDRSC=OPT,ISTATUS=ACTIVE
IPFV2    CDRM  SUBAREA=02,CDRDYN=YES,CDRSC=OPT,ISTATUS=ACTIVE
IPF      CDRM  SUBAREA=01,CDRDYN=YES,CDRSC=OPT,ISTATUS=ACTIVE
```

Figure 213. CDRMDRDA.B

```
 CATALOG VTMCDRS.B

VTMCDRS  VBUILD TYPE=CDRSC
IPFBOEDJ CDRSC  CDRM=IPFV2,ISTATUS=ACTIVE
IPFCL0E0 CDRSC  CDRM=IPFV2,ISTATUS=ACTIVE
IPFSL0E0 CDRSC  CDRM=IPFV2,ISTATUS=ACTIVE
IPFA2GL4 CDRSC  CDRM=IPFV2,ISTATUS=ACTIVE
```

Figure 214. VTMCDRS.B

```
 CATALOG CDRSSLOV.B

VTMCDRS  VBUILD TYPE=CDRSC
         NETWORK NETID=SIZPIZ00
Z00AVSL1 CDRSC
Z00T010A CDRSC
```

Figure 215. CDRSSLOV.B

```
scale=¢1.0¢.
 CATALOG APPLDRD.B
APPLDRD  VBUILD TYPE=APPL
IPFA21CD APPL  ACBNAME=IPFA21CD,                                       C
               AUTH=(ACQ,PASS,VPACE),                                  C
               APPC=NO,                                                C
               SONSCIP=YES,                                            C
               EAS=30,                                                 C
               PARSESS=YES,                                            C
               MODETAB=DRDAMOD,                                        C
               DLOGMOD=IBMRDBM,                                        C
               VPACING=0
POWER    APPL  AUTH=(ACQ)
PNET     APPL  AUTH=(PASS,ACQ),VPACING=3,MODETAB=VTMLOGTB,DLOGMOD=PNET
IESWAITT APPL  AUTH=(NOACQ)
```

Figure 216. Application Definition at ITSO

```
Problem

db2=>connect to ipfvs1a4 user cicsus1 using cicsus1

SQL30081N A communication error has been detected.Communication protocol
being used †APPC†. Communication API being used: †CPI-C†. Location where
the error was detected :††. Communication function detecting the error:
†cmallc†. Protocol specific error code(s): †24†, †*†, †*†. SQLSTATE-0800

send vtam vtam v net,act,id=ipfa2gl4,scope=all

VTAM    : Ready;
VTAM    : IST097I VARY ACCEPTED
VTAM    : IST093I IPFA2GL4 ACTIVE
VTAM    : IST663I BFINIT REQUEST FROM IPFNV3 FAILED, SENSE=08570003
VTAM    : IST664I REAL  OLU=DEIBMIPF.IPFSL0E0   REAL  DLU=DEIBMIPF.IPFA2GL4
VTAM    : IST889I SID = E5570B05CC3A82F6
VTAM    : IST264I REQUIRED RESOURCE IPFA2GL4 NOT ACTIVE
```

Figure 217. Problem -- Connect Failed

```
SOLUTION :

MSG F2

2 cemt set vtam open
```

Figure 218. Solution for Connection Failed

A.3 Field Procedure Sample

```
--------------------------------------------------------------------
FIELDPROC-THE SAME ASSEMBLER PROGRAM WAS CUSTOMIZED ON SQL/DS VSE/ESA
--------------------------------------------------------------------
*   ACC 193 V
*
*   &TRACE ALL
*   GLOBAL MACLIB DMSOM DMSGPI OSMACRO FLDPROC
*   HASM &1.......ASSEMBLE (24 BIT)
*   LOAD &1 ( RLD
*   GENMOD &1
```

Figure 219. Field Procedure Installation under VM/ESA

```
*
**************************************************************
* SLOSORT ASSEMBLE                                           *
* COLLATING SEQUENCE FOR SLOVENIA WAS IMPLEMENTED.           *
*                                                            *
**************************************************************
**************************************************************
*                                                            *
* EXAMPLE OF SORTING SEQUENCES FOR SLOVENIA:                 *
*                                                            *
* CODING FOR DOUBLE BYTE VOWEL,HAS NOT BEEN REMOVED, BUT JUST *
* SET TO 00.                                                 *
* THE CUSTOMER HAS ALTERNTIVE, TO EXTENT SORT REQUESTS.      *
*                                                            *
* MODULE NAME = SLOSORT                                      *
**************************************************************
**************************************************************
870      TITLE ¢870¢
         EJECT
SLOSORT  START 0
SLOSORT  AMODE 31
SLOSORT  RMODE ANY
*
         PRINT GEN
*
         USING SLOSORT,R12           Base register
         SAVE  (14,12),,SLOSORT      SAVE REGS
         LR    R12,R15               LOAD BASE REGISTER
         ST    R13,SAVE
         USING FPPL,R1               Parameter List
         L     R2,FPPWORK            WORK AREA ADDRESS
         L     R3,FPPFPIB            COMMON INFORMATION BLOCK
         USING FPIB,R3               Common Information Block
         L     R4,FPPCVD             COLUMN DESCRIPTOR
         L     R5,FPPFVD             FIELD  DESCRIPTOR
         L     R6,FPPPVL             PARAMETER VALUE LIST
         USING FPPVL,R6              PARAMETER VALUE LIST
*
**************************************************************
*    MAIN LINE                                               *
**************************************************************
*
*        LH    R7,FPBFCODE           GET SQL FUNCTION CODE
*        L     R15,FDFLC(R7)         SELECT APPROPRIATE ROUTINE
*        B     R15
         CLC   FPBFCODE,=AL2(FPBFDEF)
         BE    DEFINE
         CLC   FPBFCODE,=AL2(FPBFENC)
         BE    ENCODE
         CLC   FPBFCODE,=AL2(FPBFDEC)
         BE    DECODE
*
```

Figure 220 (Part 1 of 7). Field Procedure Sample

```
         MVC   FPBRTNC,=AL2(FPBRC12)   SET RETURN CODE
         MVC   FPBRSNC,=C¢CODE¢        SET REASON CODE
         L     R13,SAVE                RESTORE R13
         RETURN (14,12),RC=12
         LTORG
* FDFLC    DC    A(ENCODE,DECODE,DEFINE)
*
*****************************************************************
DEFINE   EQU   *                       FUNCTION
*****************************************************************
*
         LH    R8,FPVDTYPE-FPVD(R4)    COLUMN DATA TYPE
         CH    R8,=AL2(FPVDTVCH)       IS IT FIX CHARACTER
         BE    CDTOK
         CH    R8,=AL2(FPVDTCHR)       IS IT VAR CHARACTER
         BE    CDTOK
*
         MVC   FPBRTNC,=AL2(FPBRC4)    SET RETURN CODE
         MVC   FPBRSNC,=C¢TYPE¢        SET REASON CODE
         L     R13,SAVE                RESTORE R13
         RETURN (14,12),RC=8
*
CDTOK    EQU   *
         MVC   FPBWKLN,=H¢256¢         SET WORK AREA LENGTH (NOT USED)
         MVC   FPPVLEN,=H¢0¢           SET PARM VALUE LIST  (NOT USED)
         MVC   FPVDTYPE-FPVD(L¢FPVDTYPE,R5),FPVDTYPE-FPVD(R4)
         MVC   FPVDVLEN-FPVD(L¢FPVDVLEN,R5),FPVDVLEN-FPVD(R4)
*
         MVC   FPBRTNC,=AL2(FPBRC0)    SET RETURN CODE
         MVC   FPBRSNC,=C¢    ¢        SET REASON CODE
         L     R13,SAVE                RESTORE R13
         RETURN (14,12),RC=0
*
*****************************************************************
ENCODE   EQU   *                       FUNCTION
*****************************************************************
*
         LH    R8,FPVDTYPE-FPVD(R4)    GET COLUMN DATA TYPE
         CH    R8,=AL2(FPVDTVCH)       IS IT VAR CHARACTER ?
         BE    VCHE                    YES
*
         LH    R7,FPVDVLEN-FPVD(R4)    GET FIX CH COLUMN LENGTH
         BCTR  R7,0                    L-1
         B     ENCP                    ENCODE PROCEDURE
*
VCHE     EQU   *
         LH    R7,FPVDVALE-FPVD(R4)    GET VAR CH COLUMN LENGTH
         STH   R7,FPVDVALE-FPVD(R5)    STORE LENGTH INTO FVD
         BCTR  R7,0                    L-1
         LA    R4,2(R4)                CONSIDER 2 BYTE FIELD  LENGTH
         LA    R5,2(R5)                CONSIDER 2 BYTE COLUMN LENGTH
*        B     ENCP                    ENCODE PROCEDURE
*
```

Figure 220 (Part 2 of 7). Field Procedure Sample

```
ENCP     EQU   *
         EX    R7,MOVEE                MOVE COLUMN TO FIELD
         EX    R7,TRANE                ENCODE XLATE FIELD
EXTRE    EX    R7,TRTLE                1ST BYTE MIXED & 2ND LOWER CASE
         BC    8,NEXTE                 NO HIT, GO & GET NEXT FIELD
         LR    R9,1                    SAVE HIT ADDRESS WITHIN FIELD
         STC   R2,CLILE+1              FUNCTION BYTE TO CLI MASK
CLILE    CLI   1(R9),X¢00¢             X¢00¢ = EXPECTED 2ND BYTE
         BE    FINDE                   CONTINUE WITH THE NEXT BYTE
*
         TRT   0(0,R9),TBLUE           1ST BYTE UPPER & 2ND BUTE UPPER
         BC    8,LOOPE                 NO HIT, GO FOR LOWER CASE
         STC   R2,CLIUE+1              FUNCTION BYTE TO CLI MASK
CLIUE    CLI   1(R9),X¢00¢             X¢00¢ = EXPECTED 2ND BYTE
         BNE   LOOPE                   GO FOR 2N BYTE IN LOWER CASE
*
FINDE    TR    0(0,R9),DBWTE           GET WEIGHT FOR 1ST BYTE
         LA    R9,1(R9)                NO. TRANSLATED
*
LOOPE    SR    R9,R5                   NO. OF BYTES TRANSLATED-1
         LA    R9,1(R9)                NO. TRANSLATED
         AR    R5,R9                   POSITION TO THE NEXT BYTE
         SR    R7,R9                   REMAIN TO TRANSLATE
         BP    EXTRE                   CONTINUE TO TRANSLATE
*
*-------------------------------------------------------*
NEXTE    L     13,SAVE                 RESTORE 13
         RETURN (14,12),RC=0
*-------------------------------------------------------*
*
MOVEE    MVC   FPVDVALE-FPVD(0,R5),FPVDVALE-FPVD(R4)
TRANE    TR    FPVDVALE-FPVD(0,R5),CPLAE
TRTLE    TRT   FPVDVALE-FPVD(0,R5),TBLLE
*
***************************************************************
DECODE   EQU   *                       FUNCTION
***************************************************************
*
         LH    R8,FPVDTYPE-FPVD(R5)    GET FIELD  DATA TYPE
         CH    R8,=AL2(FPVDTVCH)       IS IT VAR CHARACTER ?
         BE    VCHD                    YES
*
         LH    R7,FPVDVLEN-FPVD(R5)    GET FIX CH FIELD  LENGTH
         BCTR  R7,0                    L-1
         B     DECP                    DECODE PROCEDURE
*
VCHD     EQU   *
         LH    R7,FPVDVALE-FPVD(R5)    GET VAR CH FIELD  LENGTH
         STH   R7,FPVDVALE-FPVD(R4)    STORE LENGTH INTO CVD
         BCTR  R7,0                    L-1
         LA    R4,2(R4)                CONSIDER 2 BYTE FIELD  LENGTH
         LA    R5,2(R5)                CONSIDER 2 BYTE COLUMN LENGTH
*        B     DECP                    ENCODE PROCEDURE
*
```

Figure 220 (Part 3 of 7). Field Procedure Sample

```
DECP     EQU  *
         EX   R7,MOVED               MOVE FIELD TO COLUMN
         EX   R7,TRAND               DECODE XLATE COLUMN
*
EXTRD    EX   R7,TRTLD               1ST BYTE MIXED & 2ND LOWER CASE
         BC   8,NEXTD                NO HIT, GO & GET NEXT FIELD
         LR   R9,1                   SAVE HIT ADDRESS WITHIN FIELD
         STC  R2,CLILD+1             FUNCTION BYTE TO CLI MASK
CLILD    CLI  1(R9),X¢00¢            X¢00¢ = EXPECTED 2ND BYTE
         BE   FINDD                  CONTINUE WITH THE NEXT BYTE
*
         TRT  0(0,R9),TBLUD          1ST BYTE UPPER & 2ND BUTE UPPER
         BC   8,LOOPD                NO HIT, GO FOR LOWER CASE
         STC  R2,CLIUD+1             FUNCTION BYTE TO CLI MASK
CLIUD    CLI  1(R9),X¢00¢            X¢00¢ = EXPECTED 2ND BYTE
         BNE  LOOPD                  GO FOR 2N BYTE IN LOWER CASE
*
FINDD    TR   0(0,R9),DBWTD          GET WEIGHT FOR 1ST BYTE
         LA   R9,1(R9)               NO. TRANSLATED
*
LOOPD    SR   R9,R5                  NO. OF BYTES TRANSLATED-1
         LA   R9,1(R9)               NO. TRANSLATED
         AR   R5,R9                  POSITION TO THE NEXT BYTE
         SR   R7,R9                  REMAIN TO TRANSLATE
         BP   EXTRD                  CONTINUE TO TRANSLATE
*
*-----------------------------------------------------------*
NEXTD    L    13,SAVE                RESTORE 13
         RETURN (14,12),RC=0
*-----------------------------------------------------------*
*
MOVED    MVC  FPVDVALE-FPVD(0,R4),FPVDVALE-FPVD(R5)
TRAND    TR   FPVDVALE-FPVD(0,R4),CPLAD
TRTLD    TRT  FPVDVALE-FPVD(0,R4),TBLLD
*
*               0 1 2 3 4 5 6 7 8 9 A B C D E F
CPLAE    DC     X¢000102030405060708090A0B0C0D0E0F¢       00 - 0F
         DC     X¢101112131415161718191A1B1C1D1E1F¢       10 - 1F
         DC     X¢202122232425262728292A2B2C2D2E2F¢       20 - 2F
         DC     X¢303132333435363738393A3B3C3D3E3F¢       30 - 3F
         DC     X¢40697A7CE17678868884554A65536247¢       40 - 4F
         DC     X¢5F949A98E7A4A6B2B0DC565C5D5445AF¢       50 - 5F
         DC     X¢43497B7D4C77798789856844614167480¢      60 - 6F
         DC     X¢BA959B99E8A5A7B3B18D46605A516652¢       70 - 7F
         DC     X¢AE7480828A929C9EA0A2D6BE90F5D2DA¢       80 - 8F
         DC     X¢BBA8AAACB6B8C0CACCCEB4BCD84F505B¢       90 - 9F
         DC     X¢7E4DD4DDE3EDEFF1F3F7D7BF91F6D3DB¢       A0 - AF
         DC     X¢4E7FFDE2FE59FBF9FCFAB5BDD94B8C64¢       B0 - BF
         DC     X¢577581838B939D9FA1A342C4C6D0C2C8¢       C0 - CF
         DC     X¢58A9ABADB7B9C1CBCDCF97EBE9DFE596¢       D0 - DF
         DC     X¢5E63D5DEE4EEF0F2F4F88EC5C7D1C3C9¢       E0 - EF
         DC     X¢6A6B6C6D6E6F707172738FECEAE0E6FF¢       F0 - FF
*               0 1 2 3 4 5 6 7 8 9 A B C D E F
*
```

Figure 220 (Part 4 of 7). Field Procedure Sample

```
*
TBLUE    DC    16X¢00¢                                     00 - 0F
         DC    16X¢00¢                                     10 - 1F
         DC    16X¢00¢                                     20 - 2F
         DC    16X¢00¢                                     30 - 3F
         DC    16X¢00¢                                     40 - 4F
         DC    16X¢00¢                                     50 - 5F
         DC    16X¢00¢                                     60 - 6F
         DC    16X¢00¢                                     70 - 7F
         DC    16X¢00¢                                     70 - 7F
         DC    16X¢00¢                                     90 - 9F
         DC    16X¢00¢                                     90 - 9F
         DC    16X¢00¢                                     90 - 9F
         DC    16X¢00¢                                     C0 - CF
         DC    16X¢00¢                                     D0 - DF
         DC    16X¢00¢                                     E0 - EF
         DC    16X¢00¢                                     F0 - FF
*
TBLLE    DC    16X¢00¢                                     00 - 0F
         DC    16X¢00¢                                     10 - 1F
         DC    16X¢00¢                                     20 - 2F
         DC    16X¢00¢                                     30 - 3F
         DC    16X¢00¢                                     40 - 4F
         DC    16X¢00¢                                     50 - 5F
         DC    16X¢00¢                                     60 - 6F
         DC    16X¢00¢                                     70 - 7F
         DC    16X¢00¢                                     70 - 7F
         DC    16X¢00¢                                     90 - 9F
         DC    16X¢00¢                                     90 - 9F
         DC    16X¢00¢                                     90 - 9F
         DC    16X¢00¢                                     C0 - CF
         DC    16X¢00¢                                     D0 - DF
         DC    16X¢00¢                                     E0 - EF
         DC    16X¢00¢                                     F0 - FF
*
DBWTE    DC    16X¢00¢                                     00 - 0F
         DC    16X¢00¢                                     10 - 1F
         DC    16X¢00¢                                     20 - 2F
         DC    16X¢00¢                                     30 - 3F
         DC    16X¢00¢                                     40 - 4F
         DC    16X¢00¢                                     50 - 5F
         DC    16X¢00¢                                     60 - 6F
         DC    16X¢00¢                                     70 - 7F
         DC      X¢00000000000000000000008C8D00000000¢      80 - 8F
         DC    16X¢00¢                                     90 - 9F
         DC      X¢0000000000000000000000000000000000¢      A0 - AF
         DC      X¢0000000000000000000000000000000000¢      B0 - BF
         DC    16X¢00¢                                     C0 - CF
         DC    16X¢00¢                                     D0 - DF
         DC    16X¢00¢                                     E0 - EF
         DC    16X¢00¢                                     F0 - FF
*
```

Figure 220 (Part 5 of 7). Field Procedure Sample

```
*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
CPLAD    DC      X¢000102030405060708090A0B0C0D0E0F¢ 0
         DC      X¢101112131415161718191A1B1C1D1E1F¢ 1
         DC      X¢202122232425262728292A2B2C2D2E2F¢ 2
         DC      X¢303132333435363738393A3B3C3D3E3F¢ 3
         DC      X¢406DCA606B5E7A4F6F614BBD64A1B09D¢ 4
         DC      X¢9E7D7F4D5D4A5AC0D0B57C9F5B5CE050¢ 5
         DC      X¢7B6C4EE1BF4C7E6E6A41F0F1F2F3F4F5¢ 6
         DC      X¢F6F7F8F981C14565466642624363A0B1¢ 7
         DC      X¢82C283C349694767486884C4BE79EAFA¢ 8
         DC      X¢8CAC85C55171DFDA5373527286C687C7¢ 9
         DC      X¢88C889C95575567691D192D293D3805F¢ A
         DC      X¢587857779ABA94D495D570909BBB8BAB¢ B
         DC      X¢96D6CEEECBEBCCECCFEF97D798D899D9¢ C
         DC      X¢CDED8EAEA2E28AAA9CBC8FAF59A3E3DD¢ D
         DC      X¢FD44B3A4E4DEFE5474DCFCDBFBA5E5A6¢ E
         DC      X¢E6A7E7A8E88DADA9E9B7B9B6B8B2B4FF¢ F
*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
*
*                  CHECK FIRST MIXED SECOND LOWER
*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
TBLLD    DC    16X¢00¢                               0
         DC    16X¢00¢                               1
         DC    16X¢00¢                               2
         DC    16X¢00¢                               3
         DC    16X¢00¢                               4
         DC    16X¢00¢                               4
         DC    16X¢00¢                               6
         DC    16X¢00¢                               6
         DC    16X¢00¢                               6
         DC    16X¢00¢                               6
         DC    16X¢00¢                               A
         DC    16X¢00¢                               A
         DC    16X¢00¢                               C
         DC    16X¢00¢                               D
         DC    16X¢00¢                               E
         DC    16X¢00¢                               F
*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
*
*                  CHECK FIRST UPPER SECOND UPPER
*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
TBLUD    DC    16X¢00¢                               0
         DC    16X¢00¢                               1
         DC    16X¢00¢                               2
         DC    16X¢00¢                               3
         DC    16X¢00¢                               4
         DC    16X¢00¢                               4
         DC    16X¢00¢                               6
         DC    16X¢00¢                               6
         DC    16X¢00¢                               8
         DC    16X¢00¢                               8
         DC    16X¢00¢                               A
         DC    16X¢00¢                               B
         DC    16X¢00¢                               C
```

Figure 220 (Part 6 of 7). Field Procedure Sample

```
         DC    16X¢00¢                              D
         DC    16X¢00¢                              E
         DC    16X¢00¢                              F
*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
*
*                  DECODE DOUBLE BYTE VOWEL
*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
DBWTD    DC    16X¢00¢                              0
         DC    16X¢00¢                              1
         DC    16X¢00¢                              2
         DC    16X¢00¢                              3
         DC    16X¢00¢                              4
         DC      X¢000000000000000000000000000000D3¢ 5
         DC    16X¢00¢                              6
         DC      X¢9500000000000000000C4000000000000¢ 7
         DC      X¢930000000000000000000000000000000¢ 8
         DC      X¢D50000000000000000000000000000000¢ 9
         DC    16X¢00¢                              A
         DC      X¢00000000000000000000000000008400¢ B
         DC    16X¢00¢                              C
         DC    16X¢00¢                              D
         DC    16X¢00¢                              E
         DC    16X¢00¢                              F
*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
*
SAVE     DS    F                       SAVE AREA
*
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
*
         ARIBFPPB                      SQL/DS COPY MACRO
         END
```

Figure 220 (Part 7 of 7). Field Procedure Sample

A.4 CDRA Conversion Tables Installation

This appendix provides an example of how the conversion tables were installed in the VSE and VM systems. The conversion tables, the load programs and the jobs to load the tables are given.

The DRDA connections via the 3174 and the 3745 are supplied.

A.4.1 CDRA Conversion Table from 500 to 870

```
000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
202122232425262728292A2B2C2D2E2F
303132333435363738393A3B3C3D3E3F
404142434445464748494A4B4C4D4E4F
505152535455565758595A5B5C5D5E5F
606162636465666768697O6B6C6D6E6F
72717473777576788O797A7B7C7D7E7F
8A8182838485868788898B8C8E8D8F9A
909192939495969798999B9C9E9DA09F
AAA1A2A3A4A5A6A7A8A9ABAEACADAFB0
B1B2B3B4B6B5B7B8B9BABB6ABCBDBEBF
C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF
D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF
E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF
F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF
```

Figure 221. CDRA Conversion Table from 500 to 870

A.4.2 CDRA Conversion Table from 870 to 500

```
000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
202122232425262728292A2B2C2D2E2F
303132333435363738393A3B3C3D3E3F
404142434445464748494A4B4C4D4E4F
505152535455565758595A5B5C5D5E5F
60616263646566676869BB6B6C6D6E6F
6A7170737275767477797A7B7C7D7E7F
788182838485868788898O8A8B8D8C8E
909192939495969798998F9A9B9D9C9F
9EA1A2A3A4A5A6A7A8A9A0AAACADABAE
AFB0B1B2B3B5B4B6B7B8B9BABCBDBEBF
C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF
D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF
E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF
F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF
```

Figure 222. CDRA Conversion Table from 870 to 500

A.4.3 Sample Program to LOAD Conversion Tables under VM

```
*
*****************************************************************
*    COBCCSID BEGIN PROGRAM                                     *
*****************************************************************
*
 IDENTIFICATION     DIVISION.
 PROGRAM-ID.        COBCCSID.
 ENVIRONMENT        DIVISION.
 CONFIGURATION      SECTION.
 INPUT-OUTPUT       SECTION.
 FILE-CONTROL.
     SELECT INPDATA ASSIGN TO  INPDATA.
     SELECT INPCONV ASSIGN TO  INPCONV.
 DATA               DIVISION.
 FILE               SECTION.
*
*****************************************************************
*    COBCCSID FILES DEFINITIONS                                 *
*****************************************************************
*
 FD  INPDATA
     LABEL RECORDS  OMITTED
     RECORDING MODE F.
 01  DAT-INPDATA.
      05   DAT-INCCSID    PIC  9(008).
      05   DAT-OUTCCSID   PIC  9(008).
      05   DAT-TRANSTYPE  PIC  X(002).
      05   DAT-ERRORBYTE  PIC  X(001).
      05   DAT-SUBTYPE    PIC  X(001).
      05   DAT-TRANSPROC  PIC  X(008).
      05   DAT-FILLER     PIC  X(052).
*
 FD  INPCONV
     LABEL RECORDS  STANDARD
     RECORDING MODE F.
 01  CON-INPCONV.
      05   CON-TRANSTAB1  PIC  X(064).
      05   CON-TRANSTAB2  PIC  X(192).
*
 WORKING-STORAGE   SECTION.
*
*****************************************************************
*    COBCCSID WORKING STORAGE AREAS                             *
*****************************************************************
*
 01  DECODED-SQLCODE      PIC   ---------9.
 01  DECODED-SQLERRD1     PIC   ---------9.
 01  DECODED-SQLERRD2     PIC   ---------9.
*
```

Figure 223 (Part 1 of 5). Conversion Table from 500 to 870

```
 01  WRK-INPDATA.
      05   WRK-INCCSID    PIC  9(008).
      05   WRK-OUTCCSID   PIC  9(008).
      05   WRK-TRANSTYPE  PIC  X(002).
      05   WRK-ERRORBYTE  PIC  X(001).
      05   WRK-SUBTYPE    PIC  X(001).
      05   WRK-TRANSPROC  PIC  X(008).
      05   WRK-FILLER     PIC  X(052).


*
 01  WRK-INPCONV.
      05   WRK-TRANSTAB1  PIC  X(064).
      05   WRK-TRANSTAB2  PIC  X(192).
*
*****************************************************************
*    COBCCSID HOST VARIABLES DECLARATIONS                        *
*****************************************************************
*
     EXEC SQL      BEGIN  DECLARE  SECTION     END-EXEC.
*
 01  SYS-INCCSID          PIC  S9(08) COMP     VALUE +0.
 01  SYS-OUTCCSID         PIC  S9(08) COMP     VALUE +0.
 01  SYS-TRANSTYPE        PIC  X(002)          VALUE SPACES.
 01  SYS-ERRORBYTE        PIC  X(001)          VALUE SPACES.
 01  SYS-SUBTYPE          PIC  X(001)          VALUE SPACES.
 01  SYS-TRANSPROC        PIC  X(008)          VALUE SPACES.
 01  SYS-TRANSTAB1        PIC  X(064)          VALUE SPACES.
 01  SYS-TRANSTAB2        PIC  X(192)          VALUE SPACES.
*
*
 01  WS-USERID            PIC   X(08)          VALUE ¢SQLDBA  ¢.
 01  WS-PASSWORD          PIC   X(08)          VALUE ¢SQLDBAPW¢.
*
     EXEC SQL      END    DECLARE  SECTION     END-EXEC.
     EXEC SQL      INCLUDE     SQLCA           END-EXEC.
*
*****************************************************************
*    COBCCSID EXECUTION BEGIN                                    *
*****************************************************************
*
 PROCEDURE         DIVISION.
*
     DISPLAY      ¢COBCCSID EXECUTION BEGIN¢
        UPON       CONSOLE
*
     EXEC SQL      WHENEVER NOT FOUND          CONTINUE
     END-EXEC.
     EXEC SQL      WHENEVER SQLWARNING         CONTINUE
     END-EXEC.
     EXEC SQL      WHENEVER SQLERROR           CONTINUE
     END-EXEC.
*
```

Figure 223 (Part 2 of 5). Conversion Table from 500 to 870

```
*****************************************************************
*    CONNECT THE SQLDBA USERID                                  *
*****************************************************************
*
     EXEC SQL      CONNECT                    :WS-USERID
                   IDENTIFIED BY              :WS-PASSWORD
     END-EXEC.
*
     IF SQLCODE  < 0
        GO TO      COBCCSID-SQLCHK.
*
*****************************************************************
*    OPEN INPUT DATA FILE                                       *
*****************************************************************
*
 COBCCSID-OPEN-INPDATA.
     OPEN INPUT    INPDATA.
*
 SQLPEDIT-READ-INPDATA.
     READ          INPDATA                     INTO WRK-INPDATA
        AT END
           GO TO                               COBCCSID-DATCHK.
*
 SQLPEDIT-MOVE-INPDATA.
     MOVE          WRK-INCCSID                 TO SYS-INCCSID.
     MOVE          WRK-OUTCCSID                TO SYS-OUTCCSID.
     MOVE          WRK-TRANSTYPE               TO SYS-TRANSTYPE.
     MOVE          WRK-ERRORBYTE               TO SYS-ERRORBYTE.
     MOVE          WRK-SUBTYPE                 TO SYS-SUBTYPE.
     MOVE          WRK-TRANSPROC               TO SYS-TRANSPROC.
*
 COBCCSID-CLOSE-INPDATA.
     CLOSE                        INPDATA.
*
*****************************************************************
*    OPEN INPUT CONV FILE                                       *
*****************************************************************
*
 COBCCSID-OPEN-INPCONV.
     OPEN INPUT    INPCONV.
*
 SQLPEDIT-READ-INPCONV.
     READ INPCONV                              INTO WRK-INPCONV
        AT END
           GO TO                               COBCCSID-CONCHK.
*
 SQLPEDIT-MOVE-INPCONV.
     MOVE          WRK-TRANSTAB1               TO SYS-TRANSTAB1.
     MOVE          WRK-TRANSTAB2               TO SYS-TRANSTAB2.
*
 COBCCSID-CLOSE-INPCONV.
     CLOSE                        INPCONV.
*
```

Figure 223 (Part 3 of 5). Conversion Table from 500 to 870

```
*****************************************************************
*    DELETE EXISTING CONVERSION CCSID TABLE RECORD              *
*****************************************************************
*
 COBCCSID-DELETE-CCSID-RECORD.
*
     EXEC SQL      DELETE
          FROM     SYSTEM.SYSSTRINGS
          WHERE    INCCSID  = :SYS-INCCSID
            AND    OUTCCSID = :SYS-OUTCCSID
     END-EXEC.
*
     IF SQLCODE  < 0
        GO TO      COBCCSID-SQLCHK.
*
*****************************************************************
*    INSERT NEW CONVERSION CCSID TABLE RECORD                   *
*****************************************************************
*
 COBCCSID-INSERT-CCSID-RECORD.
*
     EXEC SQL      INSERT
          INTO     SYSTEM.SYSSTRINGS
          VALUES
                  (
                   :SYS-INCCSID,
                   :SYS-OUTCCSID,
                   :SYS-TRANSTYPE,
                    NULL,
                    NULL,
                   :SYS-TRANSPROC,
                   :SYS-TRANSTAB1,
                   :SYS-TRANSTAB2
                  )
     END-EXEC.
*
     IF SQLCODE  < 0
        GO TO      COBCCSID-SQLCHK.
*
     GO TO         COBCCSID-ENDPROG.
*
 COBCCSID-SQLCHK.
*
     MOVE          SQLCODE                    TO DECODED-SQLCODE
     DISPLAY      ¢SQLCODE  = ¢               DECODED-SQLCODE
        UPON       CONSOLE.
     MOVE          SQLERRD(1)                 TO DECODED-SQLERRD1
     DISPLAY      ¢SQLERRD1 = ¢               DECODED-SQLERRD1
        UPON       CONSOLE.
     MOVE          SQLERRD(2)                 TO DECODED-SQLERRD2
     DISPLAY      ¢SQLERRD2 = ¢               DECODED-SQLERRD2
        UPON       CONSOLE.
*
```

Figure 223 (Part 4 of 5). Conversion Table from 500 to 870

```
           EXEC SQL      WHENEVER SQLERROR          CONTINUE
           END-EXEC.
           EXEC SQL      ROLLBACK WORK RELEASE
           END-EXEC.
      *
           DISPLAY      ¢COBCCSID EXECUTION CANCELED¢
              UPON       CONSOLE
      *
           GOBACK.
      *
       COBCCSID-DATCHK.
      *
           DISPLAY      ¢READ ERROR FOR INPDATA FILE¢
              UPON       CONSOLE.
      *
           EXEC SQL      WHENEVER SQLERROR          CONTINUE
           END-EXEC.
           EXEC SQL      ROLLBACK WORK RELEASE
           END-EXEC.
      *
           DISPLAY      ¢COBCCSID EXECUTION CANCELED¢
              UPON       CONSOLE
      *
           GOBACK.
      *
       COBCCSID-CONCHK.
      *
           DISPLAY      ¢READ ERROR FOR INPCONV FILE¢
              UPON       CONSOLE.
      *
           EXEC SQL      WHENEVER SQLERROR          CONTINUE
           END-EXEC.
           EXEC SQL      ROLLBACK WORK RELEASE
           END-EXEC.
      *
           DISPLAY      ¢COBCCSID EXECUTION CANCELED¢
              UPON       CONSOLE
      *
           GOBACK.
      *
       COBCCSID-ENDPROG.
      *
           EXEC SQL      COMMIT WORK RELEASE
           END-EXEC.
      *
           DISPLAY      ¢COBCCSID EXECUTION END¢
              UPON       CONSOLE.
      *
           MOVE          0                          TO RETURN-CODE.
      *
           GOBACK.
      *
      *****************************************************************
      *    COBCCSID END PROGRAM                                       *
      *****************************************************************
      *
```

Figure 223 (Part 5 of 5). Conversion Table from 500 to 870

A.4.4 Loading Conversion Tables under VM

```
/*******************************************************************/
/*    COBCCSID EXEC TO EXECUTE THE COBCCSID PROGRAM                */
/*******************************************************************/
/*    DEPENDENCIES:                                                */
/*                 1)  SQL/DS SQLDBA USERID WITH DBA AUTHORITY     */
/*******************************************************************/
 TRACE OFF
 ARG      ccsidn;
 GLOBAL   TXTLIB    VSC2CTXT VSC2LTXT VSC29TXT SQLXLIB
¢FILEDEF  INPDATA   DISK    ¢ccsidn¢  INPDATA  A (RECFM F LRECL 080)¢
¢FILEDEF  INPCONV   DISK    ¢ccsidn¢  INPCONV  A (RECFM V LRECL 256)¢
¢LOAD     COBCCSID (START    NOMAP)¢
 saverc = rc
 EXIT     saverc
```

Figure 224. Loading Conversion Tables under VM

A.4.5 Loading Conversion Tables under VSE

```
* $$ JOB JNM=COBCCSID,CLASS=0,DISP=D,PRI=3
* $$ LST CLASS=B,DISP=D,PRI=3,DEST=(*,LUCIANO)
* $$ PUN CLASS=B,DISP=D,PRI=3,DEST=(*,LUCIANO)
// JOB   COBCCSID  PROCESS 00370870 CONV
// LIBDEF *,SEARCH=(PRD2.PROD,PRD2.SQL350,PRD3.APPL)
// ASSGN    SYS004,SYSRDR
/* ASSGN    SYS005,DISK,VOL=SYSWK1,SHR
// ASSGN    SYS005,E01
// DLBL     INPCONV,¢CCSID.0037.0870.CONV¢,,SD
// EXTENT   SYS005,SYSWK1
// EXEC     COBCCSID,SIZE=AUTO
0000003700000870SS
/*
/&
* $$ EOJ
```

Figure 225. Loading Conversion Tables under VSE

A.4.6 Sample Program to LOAD Conversion Tables under VSE

```
* $$ JOB JNM=COBCCSID,CLASS=0,DISP=D,PRI=3
* $$ LST CLASS=B,DISP=D,PRI=3,DEST=(*,LUCIANO)
* $$ PUN DISP=I,PRI=9,CLASS=A
// JOB   COBCCSJ1   SQL/DS PRE-PROCESS EDIT PACKAGE PROGRAM
// LIBDEF *,SEARCH=(PRD2.PROD,PRD2.SQL350),CATALOG=PRD3.APPL
// ASSGN    SYS005,SYSRDR
// EXEC     IESINSRT
$ $$ LST CLASS=B,DISP=D,PRI=3,DEST=(*,LUCIANO)
// JOB   COBCCSJ2   COBOL COMPILE/LINK EDIT PACKAGE PROGRAM
// LIBDEF *,SEARCH=(PRD2.PROD,PRD2.SQL350),CATALOG=PRD3.APPL
// OPTION   ERRS,SXREF,SYM,CATAL,NODECK
   PHASE    COBCCSID,*
   MODE     AMODE(24)
   INCLUDE  ARIPRDID
   INCLUDE  ARIPADR4
/* EXEC     FCOBOL,SIZE=400K
// EXEC     IGYCRCTL,SIZE=IGYCRCTL
 CBL LIB,APOST,NOADV
* $$ END
// ON $CANCEL OR $ABEND GOTO ENDJ2
// OPTION   NOLIST,NODUMP,DECK
// EXEC     ARIPRPC,SIZE=AUTO,PARM=¢USER=SQLDBA/SQLDBAPW,              C
               PREP=COBCCSID,BLOCK,ISOLATION(CS),KEEP,DBNAME=SQLVSDB1¢
      *
      *****************************************************************
      *    COBCCSID BEGIN PROGRAM                                     *
      *****************************************************************
      *
       IDENTIFICATION     DIVISION.
       PROGRAM-ID.        COBCCSID.
       ENVIRONMENT        DIVISION.
       CONFIGURATION      SECTION.
       INPUT-OUTPUT       SECTION.
       FILE-CONTROL.
           SELECT INPDATA ASSIGN TO  INPDATA.
           SELECT INPCONV ASSIGN TO  INPCONV.
       DATA               DIVISION.
       FILE               SECTION.

      *
      *****************************************************************
      *    COBCCSID FILES DEFINITIONS                                 *
      *****************************************************************
      *
       FD  INPDATA
           LABEL RECORDS  OMITTED
           RECORDING MODE F.
```

Figure 226 (Part 1 of 6). Sample COBCCSID Installation Program

```
       01  DAT-INPDATA.
           05   DAT-INCCSID    PIC  9(008).
           05   DAT-OUTCCSID   PIC  9(008).
           05   DAT-TRANSTYPE  PIC  X(002).
           05   DAT-ERRORBYTE  PIC  X(001).
           05   DAT-SUBTYPE    PIC  X(001).
           05   DAT-TRANSPROC  PIC  X(008).
           05   DAT-FILLER     PIC  X(052).
      *
       FD  INPCONV
           LABEL RECORDS  STANDARD
           RECORDING MODE F.
       01  CON-INPCONV.
           05   CON-TRANSTAB1  PIC  X(064).
           05   CON-TRANSTAB2  PIC  X(192).
      *
       WORKING-STORAGE   SECTION.
      *
      *****************************************************************
      *    COBCCSID WORKING STORAGE AREAS                             *
      *****************************************************************
      *
       01  DECODED-SQLCODE      PIC   ---------9.
       01  DECODED-SQLERRD1     PIC   ---------9.
       01  DECODED-SQLERRD2     PIC   ---------9.
      *
       01  WRK-INPDATA.
           05   WRK-INCCSID    PIC  9(008).
           05   WRK-OUTCCSID   PIC  9(008).
           05   WRK-TRANSTYPE  PIC  X(002).
           05   WRK-ERRORBYTE  PIC  X(001).
           05   WRK-SUBTYPE    PIC  X(001).
           05   WRK-TRANSPROC  PIC  X(008).
           05   WRK-FILLER     PIC  X(052).
      *
       01  WRK-INPCONV.
           05   WRK-TRANSTAB1  PIC  X(064).
           05   WRK-TRANSTAB2  PIC  X(192).
      *
      *****************************************************************
      *    COBCCSID HOST VARIABLES DECLARATIONS                       *
      *****************************************************************
      *
           EXEC SQL      BEGIN  DECLARE  SECTION     END-EXEC.
      *
       01  SYS-INCCSID          PIC  S9(08) COMP     VALUE +0.
       01  SYS-OUTCCSID         PIC  S9(08) COMP     VALUE +0.
       01  SYS-TRANSTYPE        PIC  X(002)          VALUE SPACES.
       01  SYS-ERRORBYTE        PIC  X(001)          VALUE SPACES.
       01  SYS-SUBTYPE          PIC  X(001)          VALUE SPACES.
       01  SYS-TRANSPROC        PIC  X(008)          VALUE SPACES.
       01  SYS-TRANSTAB1        PIC  X(064)          VALUE SPACES.
       01  SYS-TRANSTAB2        PIC  X(192)          VALUE SPACES.
      *
```

Figure 226 (Part 2 of 6). Sample COBCCSID Installation Program

```
      *
       01  WS-USERID            PIC   X(08)         VALUE ¢SQLDBA  ¢.
       01  WS-PASSWORD          PIC   X(08)         VALUE ¢SQLDBAPW¢.
      *
           EXEC SQL      END    DECLARE  SECTION    END-EXEC.
           EXEC SQL      INCLUDE     SQLCA          END-EXEC.
      *
      *****************************************************************
      *    COBCCSID EXECUTION BEGIN                                   *
      *****************************************************************
      *
       PROCEDURE         DIVISION.
      *
           DISPLAY      ¢COBCCSID EXECUTION BEGIN¢
              UPON       CONSOLE
      *
           EXEC SQL      WHENEVER NOT FOUND          CONTINUE
           END-EXEC.
           EXEC SQL      WHENEVER SQLWARNING         CONTINUE
           END-EXEC.
           EXEC SQL      WHENEVER SQLERROR           CONTINUE
           END-EXEC.
      *
      *****************************************************************
      *    CONNECT THE SQLDBA USERID                                  *
      *****************************************************************
      *
           EXEC SQL      CONNECT                    :WS-USERID
                         IDENTIFIED BY              :WS-PASSWORD
           END-EXEC.
      *
           IF SQLCODE  < 0
              GO TO      COBCCSID-SQLCHK.
      *
      *****************************************************************
      *    OPEN INPUT DATA FILE                                       *
      *****************************************************************
      *
       COBCCSID-OPEN-INPDATA.
           OPEN INPUT    INPDATA.
      *
       SQLPEDIT-READ-INPDATA.
           READ          INPDATA                     INTO WRK-INPDATA
              AT END
                 GO TO                               COBCCSID-DATCHK.
      *
       SQLPEDIT-MOVE-INPDATA.
           MOVE         WRK-INCCSID                  TO SYS-INCCSID.
           MOVE         WRK-OUTCCSID                 TO SYS-OUTCCSID.
           MOVE         WRK-TRANSTYPE                TO SYS-TRANSTYPE.
           MOVE         WRK-ERRORBYTE                TO SYS-ERRORBYTE.
           MOVE         WRK-SUBTYPE                  TO SYS-SUBTYPE.
           MOVE         WRK-TRANSPROC                TO SYS-TRANSPROC.
      *
       COBCCSID-CLOSE-INPDATA.
           CLOSE                       INPDATA.
```

Figure 226 (Part 3 of 6). Sample COBCCSID Installation Program

```
 *
 ****************************************************************
 *    OPEN INPUT CONV FILE                                      *
 ****************************************************************
 *
  COBCCSID-OPEN-INPCONV.
      OPEN INPUT    INPCONV.
 *
  SQLPEDIT-READ-INPCONV.
      READ INPCONV                          INTO WRK-INPCONV
         AT END
            GO TO                           COBCCSID-CONCHK.
 *
  SQLPEDIT-MOVE-INPCONV.
      MOVE         WRK-TRANSTAB1            TO SYS-TRANSTAB1.
      MOVE         WRK-TRANSTAB2            TO SYS-TRANSTAB2.
 *
  COBCCSID-CLOSE-INPCONV.
      CLOSE                     INPCONV.
 *
 ****************************************************************
 *    DELETE EXISTING CONVERSION CCSID TABLE RECORD             *
 ****************************************************************
 *
  COBCCSID-DELETE-CCSID-RECORD.
 *
      EXEC SQL     DELETE
           FROM    SYSTEM.SYSSTRINGS
           WHERE   INCCSID  = :SYS-INCCSID
             AND   OUTCCSID = :SYS-OUTCCSID
      END-EXEC.
 *
      IF SQLCODE  < 0
         GO TO     COBCCSID-SQLCHK.
 *
 ****************************************************************
 *    INSERT NEW CONVERSION CCSID TABLE RECORD                  *
 ****************************************************************
 *
  COBCCSID-INSERT-CCSID-RECORD.
 *
      EXEC SQL     INSERT
           INTO    SYSTEM.SYSSTRINGS
           VALUES
                  (
                   :SYS-INCCSID,
                   :SYS-OUTCCSID,
                   :SYS-TRANSTYPE,
                    NULL,
                    NULL,
                   :SYS-TRANSPROC,
                   :SYS-TRANSTAB1,
                   :SYS-TRANSTAB2
                  )
      END-EXEC.
```

Figure 226 (Part 4 of 6). Sample COBCCSID Installation Program

```
      *
           IF SQLCODE  < 0
              GO TO      COBCCSID-SQLCHK.
      *
           GO TO         COBCCSID-ENDPROG.
      *
       COBCCSID-SQLCHK.
      *
           MOVE          SQLCODE                    TO DECODED-SQLCODE
           DISPLAY      ¢SQLCODE  = ¢               DECODED-SQLCODE
              UPON       CONSOLE.
           MOVE          SQLERRD(1)                 TO DECODED-SQLERRD1
           DISPLAY      ¢SQLERRD1 = ¢               DECODED-SQLERRD1
              UPON       CONSOLE.
           MOVE          SQLERRD(2)                 TO DECODED-SQLERRD2
           DISPLAY      ¢SQLERRD2 = ¢               DECODED-SQLERRD2
              UPON       CONSOLE.
      *
           EXEC SQL      WHENEVER SQLERROR          CONTINUE
           END-EXEC.
           EXEC SQL      ROLLBACK WORK RELEASE
           END-EXEC.
      *
           DISPLAY      ¢COBCCSID EXECUTION CANCELED¢
              UPON       CONSOLE
      *
           GOBACK.
      *
       COBCCSID-DATCHK.
      *
           DISPLAY      ¢READ ERROR FOR INPDATA FILE¢
              UPON       CONSOLE.
      *
           EXEC SQL      WHENEVER SQLERROR          CONTINUE
           END-EXEC.
           EXEC SQL      ROLLBACK WORK RELEASE
           END-EXEC.
      *
           DISPLAY      ¢COBCCSID EXECUTION CANCELED¢
              UPON       CONSOLE
      *
           GOBACK.
      *
       COBCCSID-CONCHK.
      *
           DISPLAY      ¢READ ERROR FOR INPCONV FILE¢
              UPON       CONSOLE.
      *
           EXEC SQL      WHENEVER SQLERROR          CONTINUE
           END-EXEC.
           EXEC SQL      ROLLBACK WORK RELEASE
           END-EXEC.
      *
```

Figure 226 (Part 5 of 6). Sample COBCCSID Installation Program

```
           DISPLAY     ¢COBCCSID EXECUTION CANCELED¢
             UPON       CONSOLE
      *
           GOBACK.
      *
       COBCCSID-ENDPROG.
      *
           EXEC SQL     COMMIT WORK RELEASE
           END-EXEC.
      *
           DISPLAY     ¢COBCCSID EXECUTION END¢
             UPON       CONSOLE.
      *
           MOVE         0                          TO RETURN-CODE.
      *
           GOBACK.
      *
      *****************************************************************
      *    COBCCSID END PROGRAM                                       *
      *****************************************************************
      *
/*
// EXEC     IESINSRT
/*
// EXEC     LNKEDT,SIZE=256K
/*
/. ENDJ2
#&
$ $$ EOJ
* $$ END
/&
* $$ EOJ
```

Figure 226 (Part 6 of 6). Sample COBCCSID Installation Program

A.4.7 Loading Conversion Tables under VSE

```
* $$ JOB JNM=COBCCSID,CLASS=0,DISP=D,PRI=3
* $$ LST CLASS=B,DISP=D,PRI=3,DEST=(*,LUCIANO)
* $$ PUN CLASS=B,DISP=D,PRI=3,DEST=(*,LUCIANO)
// JOB   COBCCSID  PROCESS 05000870 CONV
// LIBDEF *,SEARCH=(PRD2.PROD,PRD2.SQL350,PRD3.APPL)
// ASSGN    SYS004,SYSRDR
// ASSGN    SYS005,181
// TLBL     INPCONV
// EXEC     COBCCSID,SIZE=AUTO
0000050000000870SS
/*
/&
* $$ EOJ
```

Figure 227. Loading Conversion Tables under VSE

A.4.8 Preprocessing Sample Conversion Program under VM

```
/*******************************************************************/
/*    COBCPREP EXEC TO PREPROCESS COBCCSID AND SQLPEXPL PROGRAMS    */
/*******************************************************************/
 TRACE OFF
/*******************************************************************/
/*    PREPROCESS THE COBCCSID PROGRAM                              */
/*******************************************************************/
¢EXEC SQLPREP COB PP(PREP=COBCCSID,USER=SQLDBA/SQLDBAPW,CCSIDS(500))
              SYSIN       (COBCCSID COBSQL   A1)
              SYSPRINT    (COBCCSID PRPLIST  A1)
              SYSPUNCH    (COBCCSID COBOL    A1)¢
¢COBOL2 COBCCSID (APOST¢
¢ERASE  COBCCSID  PRPLIST¢
¢ERASE  COBCCSID  LISTING¢
¢ERASE  COBCCSID  COBOL¢
 EXIT
```

Figure 228. Preprocessing Sample Conversion Program under VM

A.4.9 Connection via 3174 for ITSO and ZPIZ

```
DEFINE_LOCAL_CP  FQ_CP_NAME(DEIBMIPF.IPFCP20E)
                 DESCRIPTION(LOCAL NODE FOR IPFL3174 VIA 3174)
                 CP_ALIAS(IPFCP20E)
                 NAU_ADDRESS(INDEPENDENT_LU)
                 NODE_TYPE(EN)
                 NODE_ID(X¢05DE000E¢)
                 NW_FP_SUPPORT(NONE)
                 HOST_FP_SUPPORT(YES)
                 MAX_COMP_LEVEL(NONE)
                 MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK  LINK_NAME(LINK0001)
                     DESCRIPTION(VIA 3174)
                     ADJACENT_NODE_TYPE(NN)
                     PREFERRED_NN_SERVER(NO)
                     DLC_NAME(IBMTRNET)
                     ADAPTER_NUMBER(0)
                     DESTINATION_ADDRESS(X¢400020201001O4¢)
                     ETHERNET_FORMAT(NO)
                     CP_CP_SESSION_SUPPORT(YES)
                     ACTIVATE_AT_STARTUP(YES)
                     LIMITED_RESOURCE(NO)
                     LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                     SOLICIT_SSCP_SESSION(YES)
                     MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                     USE_PUNAME_AS_CPNAME(NO)
                     EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                     COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                     COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                     SECURITY(USE_ADAPTER_DEFINITION)
                     PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_3(USE_ADAPTER_DEFINITION);
```

Figure 229 (Part 1 of 4). DRDA Connection via 3174 for ITSO and ZPIZ

```
DEFINE_LOGICAL_LINK  LINK_NAME(HOST0001)
                     FQ_ADJACENT_CP_NAME(DEIBMIPF.IPF      )
                     ADJACENT_NODE_TYPE(LEN)
                     DLC_NAME(IBMTRNET)
                     ADAPTER_NUMBER(0)
                     DESTINATION_ADDRESS(X¢400010301001O4¢)
                     ETHERNET_FORMAT(NO)
                     CP_CP_SESSION_SUPPORT(NO)
                     ACTIVATE_AT_STARTUP(YES)
                     LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                     LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                     SOLICIT_SSCP_SESSION(YES)
                     PU_NAME(IPFP100B)
                     NODE_ID(X¢05DE000B¢)
                     MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                     USE_PUNAME_AS_CPNAME(NO)
                     EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                     COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                     COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                     SECURITY(USE_ADAPTER_DEFINITION)
                     PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOCAL_LU  LU_NAME(IPFCL0E0)
                 DESCRIPTION(AVSVM GATEWAY)
                 LU_ALIAS(IPF3174 )
                 NAU_ADDRESS(INDEPENDENT_LU);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(DEIBMIPF.IPFA2GL4)
                   DESCRIPTION(ITSO 3174 VM SQL/DS)
                   PARTNER_LU_ALIAS(BOEVMIS2)
                   PARTNER_LU_UNINTERPRETED_NAME(IPFA2GL4)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(DEIBMIPF.IPFA21CD)
                   DESCRIPTION(ITSO 3174 VSE SQL/DS)
                   PARTNER_LU_ALIAS(VSEANL13)
                   PARTNER_LU_UNINTERPRETED_NAME(IPFA21CD)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(YES)
                   PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(SIZPIZ00.Z00AVSL1)
                   DESCRIPTION(ZPIZ 3174 VM SQL/DS)
                   PARTNER_LU_ALIAS(Z00VM317)
                   PARTNER_LU_UNINTERPRETED_NAME(Z00AVSL1)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);
```

Figure 229 (Part 2 of 4). DRDA Connection via 3174 for ITSO and ZPIZ

```
DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(SIZPIZ00.Z00CICS4)
                   DESCRIPTION(ZPIZ 3174 VSE SQL/DS)
                   PARTNER_LU_ALIAS(Z00VS317)
                   PARTNER_LU_UNINTERPRETED_NAME(Z00CICS4)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(YES)
                   PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(DEIBMIPF.IPF     )
                   PARTNER_LU_ALIAS(IPF)
                   PARTNER_LU_UNINTERPRETED_NAME(IPF     )
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(DEIBMIPF.IPF     )
                            WILDCARD_ENTRY(NO)
                            FQ_OWNING_CP_NAME(DEIBMIPF.IPF     )
                            LOCAL_NODE_NN_SERVER(NO);

DEFINE_MODE  MODE_NAME(IBMRDBM )
             DESCRIPTION(ITSO IBMRDBM)
             COS_NAME(#CONNECT)
             DEFAULT_RU_SIZE(YES)
             RECEIVE_PACING_WINDOW(4)
             MAX_NEGOTIABLE_SESSION_LIMIT(32767)
             PLU_MODE_SESSION_LIMIT(20)
             MIN_CONWINNERS_SOURCE(10)
             COMPRESSION_NEED(PROHIBITED)
             PLU_SLU_COMPRESSION(NONE)
             SLU_PLU_COMPRESSION(NONE);

DEFINE_MODE  MODE_NAME(IBMRDB  )
             DESCRIPTION(ZPIZ IBMRDB)
             COS_NAME(#CONNECT)
             DEFAULT_RU_SIZE(YES)
             RECEIVE_PACING_WINDOW(4)
             MAX_NEGOTIABLE_SESSION_LIMIT(32767)
             PLU_MODE_SESSION_LIMIT(20)
             MIN_CONWINNERS_SOURCE(10)
             COMPRESSION_NEED(PROHIBITED)
             PLU_SLU_COMPRESSION(NONE)
             SLU_PLU_COMPRESSION(NONE);

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                 DEFAULT_MODE_NAME(BLANK)
                 DEFAULT_LOCAL_LU_ALIAS(IPF3174 )
                 MAX_MC_LL_SEND_SIZE(32767)
                 DIRECTORY_FOR_INBOUND_ATTACHES(*)
                 DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                 DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                 DEFAULT_TP_CONV_SECURITY_RQD(NO)
                 MAX_HELD_ALERTS(10);
```

Figure 229 (Part 3 of 4). DRDA Connection via 3174 for ITSO and ZPIZ

```
DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(IPFVM1S4)
                        DESCRIPTION(ITSO 3174 VM SQL - SQLVMDB0)
                        PARTNER_LU_ALIAS(BOEVMIS2         )
                        MODE_NAME(IBMRDBM )
                        TP_NAME(SQLVMDB0);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(IPFVS1S4)
                        DESCRIPTION(ITSO 3174 VSE SQL  - SQLVSDB1)
                        PARTNER_LU_ALIAS(VSEANL13         )
                        MODE_NAME(IBMRDBM )
                        TP_NAME(TPN1);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(IPFVM2S4)
                        DESCRIPTION(ITSO 3174 VM SQL - SQLVMDB6)
                        PARTNER_LU_ALIAS(BOEVMIS2         )
                        MODE_NAME(IBMRDBM )
                        TP_NAME(SQLVMDB6);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(Z00VM1S4)
                        DESCRIPTION(ZPIZ 3174 VM SQL - SQLDBA)
                        PARTNER_LU_ALIAS(Z00VM317         )
                        MODE_NAME(IBMRDB  )
                        TP_NAME(SQLDBA);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(Z00VS1S4)
                        DESCRIPTION(ZPIZ 3174 VSE SQL  - SQLDS)
                        PARTNER_LU_ALIAS(Z00VS317         )
                        MODE_NAME(IBMRDB  )
                        TP_NAME(TPN1);

START_ATTACH_MANAGER;
```

Figure 229 (Part 4 of 4). DRDA Connection via 3174 for ITSO and ZPIZ

A.4.10 Connection via 3745 for ITSO and ZPIZ

```
DEFINE_LOCAL_CP  FQ_CP_NAME(DEIBMIPF.IPFCPS0E)
                 DESCRIPTION(LOCAL NODE FOR IPFL3745 VIA NCP)
                 CP_ALIAS(IPFCPS0E)
                 NAU_ADDRESS(INDEPENDENT_LU)
                 NODE_TYPE(EN)
                 NODE_ID(X¢05DE000E¢)
                 NW_FP_SUPPORT(NONE)
                 HOST_FP_SUPPORT(YES)
                 HOST_FP_LINK_NAME(HOST0001)
                 MAX_COMP_LEVEL(NONE)
                 MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK  LINK_NAME(HOST0001)
                     DESCRIPTION(ITSO & ZPIZ VIA 3745 CONNECTION)
                     FQ_ADJACENT_CP_NAME(DEIBMIPF.IPFV2    )
                     ADJACENT_NODE_TYPE(LEN)
                     DLC_NAME(IBMTRNET)
                     ADAPTER_NUMBER(0)
                     DESTINATION_ADDRESS(X¢400010301002O4¢)
                     ETHERNET_FORMAT(NO)
                     CP_CP_SESSION_SUPPORT(NO)
                     ACTIVATE_AT_STARTUP(YES)
                     LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                     LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                     SOLICIT_SSCP_SESSION(YES)
                     NODE_ID(X¢05DE000E¢)
                     MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                     USE_PUNAME_AS_CPNAME(NO)
                     EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                     COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                     COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                     SECURITY(USE_ADAPTER_DEFINITION)
                     PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_3(USE_ADAPTER_DEFINITION);
```

Figure 230 (Part 1 of 4). DRDA Connection via 3745 for ITSO and ZPIZ

```
DEFINE_LOGICAL_LINK  LINK_NAME(HOST0002)
                     DESCRIPTION(FOR 3270 SESSIONS)
                     FQ_ADJACENT_CP_NAME(DEIBMIPF.IPF      )
                     ADJACENT_NODE_TYPE(LEN)
                     DLC_NAME(IBMTRNET)
                     ADAPTER_NUMBER(0)
                     DESTINATION_ADDRESS(X¢400010301001 04¢)
                     ETHERNET_FORMAT(NO)
                     CP_CP_SESSION_SUPPORT(NO)
                     ACTIVATE_AT_STARTUP(YES)
                     LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                     LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                     SOLICIT_SSCP_SESSION(YES)
                     PU_NAME(IPFP100B)
                     NODE_ID(X¢05DE000B¢)
                     MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                     USE_PUNAME_AS_CPNAME(NO)
                     EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                     COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                     COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                     SECURITY(USE_ADAPTER_DEFINITION)
                     PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOCAL_LU  LU_NAME(IPFSL0E0)
                 DESCRIPTION(ITSO -LOCAL QUEUE MANAGER VIA 3745)
                 LU_ALIAS(IPF3745 )
                 NAU_ADDRESS(INDEPENDENT_LU);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(DEIBMIPF.IPFA2GL4)
                   DESCRIPTION(ITSO 3745 VM SQL/DS)
                   PARTNER_LU_ALIAS(BOEVMIS2)
                   PARTNER_LU_UNINTERPRETED_NAME(IPFA2GL4)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(DEIBMIPF.IPFA21CD)
                   DESCRIPTION(ITSO 3745 VSE SQL/DS)
                   PARTNER_LU_ALIAS(VSEANL13)
                   PARTNER_LU_UNINTERPRETED_NAME(IPFA21CD)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(SIZPIZ00.Z00AVSL1)
                   DESCRIPTION(ZPIZ 3745 VM SQL/DS)
                   PARTNER_LU_ALIAS(Z00VM374)
                   PARTNER_LU_UNINTERPRETED_NAME(Z00AVSL1)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);
```

Figure 230 (Part 2 of 4). DRDA Connection via 3745 for ITSO and ZPIZ

```
DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(SIZPIZ00.Z00CICS4)
                   DESCRIPTION(ZPIZ 3745 VSE SQL/DS)
                   PARTNER_LU_ALIAS(Z00VS374)
                   PARTNER_LU_UNINTERPRETED_NAME(Z00CICS4)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(DEIBMIPF.IPF     )
                   DESCRIPTION(3270 SESSIONS)
                   PARTNER_LU_ALIAS(IPF)
                   PARTNER_LU_UNINTERPRETED_NAME(IPF     )
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(DEIBMIPF.IPFA2GL4)
                            WILDCARD_ENTRY(NO)
                            FQ_OWNING_CP_NAME(DEIBMIPF.IPFV2   )
                            LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(DEIBMIPF.IPFA21CD)
                            WILDCARD_ENTRY(NO)
                            FQ_OWNING_CP_NAME(DEIBMIPF.IPFV2   )
                            LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(SIZPIZ00.Z00AVSL1)
                            WILDCARD_ENTRY(NO)
                            FQ_OWNING_CP_NAME(DEIBMIPF.IPFV2   )
                            LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(SIZPIZ00.Z00CICS4)
                            WILDCARD_ENTRY(NO)
                            FQ_OWNING_CP_NAME(DEIBMIPF.IPFV2   )
                            LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(DEIBMIPF.IPF     )
                            WILDCARD_ENTRY(NO)
                            FQ_OWNING_CP_NAME(DEIBMIPF.IPF     )
                            LOCAL_NODE_NN_SERVER(NO);

DEFINE_MODE  MODE_NAME(IBMRDBM )
             DESCRIPTION(ITSO IBMRDBM)
             COS_NAME(#CONNECT)
             DEFAULT_RU_SIZE(YES)
             RECEIVE_PACING_WINDOW(4)
             MAX_NEGOTIABLE_SESSION_LIMIT(32767)
             PLU_MODE_SESSION_LIMIT(20)
             MIN_CONWINNERS_SOURCE(10)
             COMPRESSION_NEED(PROHIBITED)
             PLU_SLU_COMPRESSION(NONE)
             SLU_PLU_COMPRESSION(NONE);
```

Figure 230 (Part 3 of 4). DRDA Connection via 3745 for ITSO and ZPIZ

```
DEFINE_MODE  MODE_NAME(IBMRDB  )
             DESCRIPTION(ZPIZ IBMRDB)
             COS_NAME(#CONNECT)
             DEFAULT_RU_SIZE(YES)
             RECEIVE_PACING_WINDOW(4)
             MAX_NEGOTIABLE_SESSION_LIMIT(32767)
             PLU_MODE_SESSION_LIMIT(20)
             MIN_CONWINNERS_SOURCE(10)
             COMPRESSION_NEED(PROHIBITED)
             PLU_SLU_COMPRESSION(NONE)
             SLU_PLU_COMPRESSION(NONE);

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                 DEFAULT_MODE_NAME(BLANK)
                 DEFAULT_LOCAL_LU_ALIAS(IPF3745 )
                 MAX_MC_LL_SEND_SIZE(32767)
                 DIRECTORY_FOR_INBOUND_ATTACHES(*)
                 DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                 DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                 DEFAULT_TP_CONV_SECURITY_RQD(NO)
                 MAX_HELD_ALERTS(10);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(IPFVS1S5)
                       DESCRIPTION(ITSO 3745 VSE SQL - SQLVSDB1)
                       PARTNER_LU_ALIAS(VSEANL13         )
                       MODE_NAME(IBMRDBM )
                       TP_NAME(TPN1);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(IPFVM1S5)
                       DESCRIPTION(ITSO 3745 VM SQL - SQLVSDB1)
                       PARTNER_LU_ALIAS(BOEVMIS2         )
                       MODE_NAME(IBMRDBM )
                       TP_NAME(SQLVMDB0);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(IPFVM2S5)
                       DESCRIPTION(ITSO 3745 VM SQL - SQLVSDB6)
                       PARTNER_LU_ALIAS(BOEVMIS2         )
                       MODE_NAME(IBMRDBM )
                       TP_NAME(SQLVMDB6);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(Z00VS1S5)
                       DESCRIPTION(ZPIZ 3745 VSE SQL - SQLDS)
                       PARTNER_LU_ALIAS(Z00VS374         )
                       MODE_NAME(IBMRDB  )
                       TP_NAME(TPN1);

DEFINE_CPIC_SIDE_INFO  SYMBOLIC_DESTINATION_NAME(Z00VM1S5)
                       DESCRIPTION(ZPIZ 3745 VM SQL - SQLDBA)
                       PARTNER_LU_ALIAS(Z00VM374         )
                       MODE_NAME(IBMRDB  )
                       TP_NAME(SQLDBA);

START_ATTACH_MANAGER;
```

Figure 230 (Part 4 of 4). DRDA Connection via 3745 for ITSO and ZPIZ

Appendix B. Customization and Displays

B.1 SNA Server/6000 Customization on ITSO1

B.1.1 SNA Global Information and Local LU Profile

Table 21. SNA Global Information and Local LU Profile on ITSO1

SNA Global Information	
Control point (CP) name	DEIBMIPF.IPFP221B
CP alias	IPFP221B
Node ID (for XID)	X′05DE001B′
Node type	End node (EN)
6.2 Local LU Profile	
Current profile name	IPF3174
Local LU name	IPFBOEDJ
Local LU alias	IPF3174

B.1.2 Token-Ring Adapter

```
                                                †Entry Fields†
Token-Ring Adapter                                 tok0
Description                                        Token-Ring High-Performance
Status                                             Available
Location                                           00-05
Receive data transfer OFFSET                       92
TRANSMIT queue size                                30
RECEIVE queue size                                 30
STATUS BLOCK queue size                            10
RING speed                                         16
Receive ATTENTION MAC frame                        no
Receive BEACON MAC frame                           no
Enable ALTERNATE TOKEN RING address                yes
ALTERNATE TOKEN RING address                       0x40001010101b
Apply change to DATABASE only                      no
```

Figure 231. Show Characteristics of a Token-Ring Adapter

B.1.3 Token-Ring Link Station

```
                                                   †Entry Fields†
Current profile name                                IPFL3174
New profile name
Use Control Point¢s XID node ID?                    yes
   If no, XID node ID                               *
SNA DLC Profile name                                tok0.00001
Stop link station on inactivity?                    no
   If yes, Inactivity time-out (0-10 minutes)       0
LU address registration?                            no
   If yes,
      LU Address Registration Profile name
Trace link?                                         no
   If yes, Trace size                               long

Adjacent Node Address Parameters
   Access routing                                   link_address
   If link_name, Remote link name
   If link_address,
      Remote link address                           400020201001
      Remote SAP address (02-fa)                    04

Adjacent Node Identification Parameters
   Verify adjacent node?                            no
   Network ID of adjacent node                      DEIBMIPF
   CP name of adjacent node
   XID node ID of adjacent node (LEN node only)     *
   Node type of adjacent node                       learn

Link Activation Parameters
   Solicit SSCP sessions?                           yes
   Initiate call when link station is activated?    yes
   Activate link station at SNA start up?           yes
   Activate on demand?                              no
   CP-CP sessions supported?                        yes
   If yes,
      Adjacent network node preferred server?       no
   Partner required to support CP-CP sessions?      no
   Initial TG number (0-20)                         0

Restart Parameters
   Restart on activation?                           no
   Restart on normal deactivation?                  no
   Restart on abnormal deactivation?                no

Transmission Group COS Characteristics
   Effective capacity                               15974400
   Cost per connect time                            0
   Cost per byte                                    0
   Security                                         nonsecure
   Propagation delay                                lan
   User-defined 1                                   128
   User-defined 2                                   128
   User-defined 3                                   128
```

Figure 232. Change/Show Token-Ring Link Station Profile IPFL3174

```
                                                     †Entry Fields†
Current profile name                                   Z00L3174
New profile name
Use Control Point¢s XID node ID?                       yes
   If no, XID node ID                                  *
SNA DLC Profile name                                   tok0.00001
Stop link station on inactivity?                       no
   If yes, Inactivity time-out (0-10 minutes)          0
LU address registration?                               no
   If yes,
      LU Address Registration Profile name
Trace link?                                            no
   If yes, Trace size                                  long

Adjacent Node Address Parameters
   Access routing                                      link_address
   If link_name, Remote link name
   If link_address,
      Remote link address                              400020201001
      Remote SAP address (02-fa)                       04

Adjacent Node Identification Parameters
   Verify adjacent node?                               no
   Network ID of adjacent node                         SIZPIZ00
   CP name of adjacent node
   XID node ID of adjacent node (LEN node only)        *
   Node type of adjacent node                          learn

Link Activation Parameters
   Solicit SSCP sessions?                              yes
   Initiate call when link station is activated?       yes
   Activate link station at SNA start up?              yes
   Activate on demand?                                 no
   CP-CP sessions supported?                           yes
   If yes,
      Adjacent network node preferred server?          no
   Partner required to support CP-CP sessions?         no
   Initial TG number (0-20)                            0

Restart Parameters
   Restart on activation?                              no
   Restart on normal deactivation?                     no
   Restart on abnormal deactivation?                   no

Transmission Group COS Characteristics
   Effective capacity                                  15974400
   Cost per connect time                               0
   Cost per byte                                       0
   Security                                            nonsecure
   Propagation delay                                   lan
   User-defined 1                                      128
   User-defined 2                                      128
   User-defined 3                                      128
```

Figure 233. Change/Show Token-Ring Link Station Profile Z00L3174

B.1.4 Control Point

```
                                                      †Entry Fields†
Profile name                                            node_cp
XID node ID                                             05de001b
Network name                                            DEIBMIPF
Control Point (CP) name                                 IPFP221B
Control Point alias                                     IPFP221B
Control Point type                                      appn_end_node
Maximum number of cached routing trees                  500
Maximum number of nodes in the TRS database             500
Route addition resistance                               128
Comments
```

Figure 234. Change/Show Control Point Profile

B.1.5 LU 6.2 Local LU

```
                                                     †Entry Fields†
Current profile name                                    IPF3174
New profile name
Local LU name                                           IPFBOEDJ
Local LU alias                                          IPF3174
Local LU is dependent?                                  no
   If yes,
      Local LU address (1-255)
      System services control point
         (SSCP) ID (*, 0-65535)                         *
      Link Station Profile name
Conversation Security Access List Profile name
Comments                                                LU used for 3174
```

Figure 235. Change/Show LU 6.2 Local LU Profile

B.1.6 Partner LU 6.2 Location

```
                                                      †Entry Fields†

Current profile name                                    IPFVS4
New profile name
Fully qualified partner LU name                         DEIBMIPF.IPFA21CD
Partner LU location method                              link_station
If owning_cp,
   Fully qualified owning Control Point (CP) name
   Local node is network server for LEN node?           no
   Fully qualified network node server name
If link_station,
   Local LU name                                        IPFBOEDJ
   Link Station Profile name                            IPFL3174
Comments
```

Figure 236. Change/Show Partner LU 6.2 Location Profile IPFVS4

```
                                                   †Entry Fields†

Current profile name                                  IPFVM4
New profile name
Fully qualified partner LU name                       DEIBMIPF.IPFA2GL4
Partner LU location method                            link_station
If owning_cp,
   Fully qualified owning Control Point (CP) name
   Local node is network server for LEN node?         no
   Fully qualified network node server name
If link_station,
   Local LU name                                      IPFBOEDJ
   Link Station Profile name                          IPFL3174
Comments
```

Figure 237. Change/Show Partner LU 6.2 Location Profile IPFVM4

```
                                                   †Entry Fields†

Current profile name                                  Z00VS4
New profile name
Fully qualified partner LU name                       SIZPIZ00.Z00CICS4
Partner LU location method                            link_station
If owning_cp,
   Fully qualified owning Control Point (CP) name
   Local node is network server for LEN node?         no
   Fully qualified network node server name
If link_station,
   Local LU name                                      IPFBOEDJ
   Link Station Profile name                          Z00L3174
Comments
```

Figure 238. Change/Show Partner LU 6.2 Location Profile Z00VS4

```
                                                   †Entry Fields†

Current profile name                                  Z00VM4
New profile name
Fully qualified partner LU name                       SIZPIZ00.Z00AVSL1
Partner LU location method                            link_station
If owning_cp,
   Fully qualified owning Control Point (CP) name
   Local node is network server for LEN node?         no
   Fully qualified network node server name
If link_station,
   Local LU name                                      IPFBOEDJ
   Link Station Profile name                          Z00L3174
Comments
```

Figure 239. Change/Show Partner LU 6.2 Location Profile Z00VM4

B.1.7 LU 6.2 Mode

```
                                               †Entry Fields†
Current profile name                              IBMRDBM
New profile name
Mode name                                         IBMRDBM
Maximum number of sessions (1-5000)               60
Minimum contention winners (0-5000)               30
Minimum contention losers (0-5000)                30
Auto activate limit (0-500)                       0
Upper bound for adaptive receive pacing window    16
Receive pacing window (0-63)                      3
Maximum RU size (128,...,32768: multiples of 32)  4096
Minimum RU size (128,...,32768: multiples of 32)  1024
Class of Service (COS) name                       #CONNECT
Comments
```

Figure 240. Change/Show LU 6.2 Mode Profile IBMRDBM

```
                                               †Entry Fields†
Current profile name                              IBMRDB
New profile name
Mode name                                         IBMRDB
Maximum number of sessions (1-5000)               20
Minimum contention winners (0-5000)               10
Minimum contention losers (0-5000)                10
Auto activate limit (0-500)                       0
Upper bound for adaptive receive pacing window    16
Receive pacing window (0-63)                      7
Maximum RU size (128,...,32768: multiples of 32)  1920
Minimum RU size (128,...,32768: multiples of 32)  256
Class of Service (COS) name                       #CONNECT
Comments
```

Figure 241. Change/Show LU 6.2 Mode Profile IBMRDB

B.1.8 LU 6.2 Side Information Profile

```
Current profile name                              IPFVS1S4
New profile name
Local LU or Control Point alias                   IPF3174
Provide only one of the following:
   Partner LU alias
   Fully qualified partner LU name                DEIBMIPF.IPFA21CD
Mode name                                         IBMRDBM
Remote transaction program name (RTPN)            TPN1
RTPN in hexadecimal?                              no
Comments
```

Figure 242. Change/Show LU 6.2 Side Information Profile IPFVS1S4

```
Current profile name                              IPFVM1S4
New profile name
Local LU or Control Point alias                   IPF3174
Provide only one of the following:
   Partner LU alias
   Fully qualified partner LU name                DEIBMIPF.IPFA2GL4
Mode name                                         IBMRDBM
Remote transaction program name (RTPN)            SQLVMDB0
RTPN in hexadecimal?                              no
```

Figure 243. Change/Show LU 6.2 Side Information Profile IPFVM1S4

```
Current profile name                              Z00VS1S4
New profile name
Local LU or Control Point alias                   IPF3174
Provide only one of the following:
   Partner LU alias
   Fully qualified partner LU name                SIZPIZ00.Z00CICS4
Mode name                                         IBMRDB
Remote transaction program name (RTPN)            TPN1
RTPN in hexadecimal?                              no
```

Figure 244. Change/Show LU 6.2 Side Information Profile Z00VS1S4

```
Current profile name                              Z00VM1S4
New profile name
Local LU or Control Point alias                   IPF3174
Provide only one of the following:
   Partner LU alias
   Fully qualified partner LU name                SIZPIZ00.Z00AVSL1
Mode name                                         IBMRDB
Remote transaction program name (RTPN)            SQLDBA
RTPN in hexadecimal?                              no
```

Figure 245. Change/Show LU 6.2 Side Information Profile Z00VM1S4

B.2 DB2 for AIX Customization on ITSO1

B.2.1 DB2 for AIX Database Directory

```
 System Database Directory

Number of entries in the directory = 4


Database 1 entry:

Database alias                  = IPFVS1A4
Database name                   = IPFVS1D4
Node name                       = IPFVS1N4
Database release level          = 6.00
Comment                         =
Directory entry type            = Remote
Authentication                  = DCS

Database 2 entry:

Database alias                  = IPFVM1A4
Database name                   = IPFVM1D4
Node name                       = IPFVM1N4
Database release level          = 6.00
Comment                         =
Directory entry type            = Remote
Authentication                  = DCS

Database 3 entry:

Database alias                  = Z00VS1A4
Database name                   = Z00VS1D4
Node name                       = Z00VS1N4
Database release level          = 6.00
Comment                         =
Directory entry type            = Remote
Authentication                  = DCS

Database 4 entry:

Database alias                  = Z00VM1A4
Database name                   = Z00VM1D4
Node name                       = Z00VM1N4
Database release level          = 6.00
Comment                         =
Directory entry type            = Remote
Authentication                  = DCS
```

Figure 246. Database Directory on ITSO1

B.2.2 DB2 for AIX DCS Directory

```
    Database Connection Services (DCS) Directory

Number of entries in the directory = 4


DCS 1 entry:

Local database name                = IPFVS1D4
Target database name               = SQLVSDB1
Application requestor name         =
DCS parameters                     =
Comment                            =
DCS directory release level        = 0x0100

DCS 2 entry:

Local database name                = IPFVM1D4
Target database name               = SQLVMDB0
Application requestor name         =
DCS parameters                     =
Comment                            =
DCS directory release level        = 0x0100

DCS 3 entry:

Local database name                = Z00VS1D4
Target database name               = SQLDS
Application requestor name         =
DCS parameters                     =
Comment                            =
DCS directory release level        = 0x0100

DCS 4 entry:

Local database name                = Z00VM1D4
Target database name               = SQLDBA
Application requestor name         =
DCS parameters                     =
Comment                            =
DCS directory release level        = 0x0100
```

Figure 247. DCS Directory on ITSO1

B.2.3 DB2 for AIX Node Directory

```
 Node Directory

Number of entries in the directory = 4


Node 1 entry:

Node name                       = IPFVS1N4
Comment                         =
Protocol                        = APPC
Symbolic destination name       = IPFVS1S4
Security type                   = PROGRAM

Node 2 entry:

Node name                       = IPFVM1N4
Comment                         =
Protocol                        = APPC
Symbolic destination name       = IPFVM1S4
Security type                   = PROGRAM

Node 3 entry:

Node name                       = Z00VS1N4
Comment                         =
Protocol                        = APPC
Symbolic destination name       = Z00VS1S4
Security type                   = PROGRAM

Node 4 entry:

Node name                       = Z00VM1N4
Comment                         =
Protocol                        = APPC
Symbolic destination name       = Z00VM1S4
Security type                   = PROGRAM
```

Figure 248. Node Directory on ITSO1

B.3 Database Connection from ITSO1 to SQL/DS on VM/VSE

B.3.1 Status of SNA

```
                         COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

Subsystem         Group            PID     Status
 sna              sna              6411    active


F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit
```

Figure 249. Display the Status of SNA

B.3.2 SNA Global Information

```
                         COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

•TOP“
**************************************
       SNA Global Information
**************************************
Status                                  Active
Control point (CP) name                 DEIBMIPF.IPFP221B
CP alias                                IPFP221B
Node ID (for XID)                       X¢05DE001B¢
Node type                               End node (EN)
Max. number of cached routing trees     500
Max. number of nodes in the TDB         500
Route additional resistance             128
Maximum number of sessions              200
Maximum number of conversations         200
Implicit partner LU support?            Yes
NMVT action when no NMVT process        Reject
Control Point (CP) profile comment
Product version                         1.3.95.293
Local hostname (TCP/IP)                 itso1
Time of last verified configuration     Thu Oct 17 12:10:43 1996

•BOTTOM“

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit
```

Figure 250. Display SNA Global Information

B.3.3 Active Link Information

```
                                  COMMAND STATUS

Command: OK             stdout: yes             stderr: no

Before command completion, additional instructions may appear below.

    Link          Adjacent      Node    Device            # of local  In
   station        CP name       type     name      State   sessions   use
-------------- ----------------- ----- --------- ---------- ---------- -----
@tok0.4                                tok0      Starting            0 No
IPFL3174       DEIBMIPF.IPFCP200 NN    tok0      Active              2 Yes


F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit
```

Figure 251. Display Active Link Information

B.3.4 LU 6.2 Session Information

```
                                  COMMAND STATUS

Command: OK             stdout: yes             stderr: no

Before command completion, additional instructions may appear below.

             Local             Partner          Mode      Link
 CGID       LU name            LU name          name     station     State
------ ----------------- ----------------- -------- ----------- ----------
2      DEIBMIPF.IPFP221B DEIBMIPF.IPFCP200 CPSVCMG  IPFL3174    Available
1      DEIBMIPF.IPFP221B DEIBMIPF.IPFCP200 CPSVCMG  IPFL3174    Available


F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit
```

Figure 252. Display LU 6.2 Session Information before Database Connection

B.3.5 Database Connection with Corresponding CCSID

The connection is made from ITSO1 with code page IBM-850 to SQL/DS on VSE (Böblingen) with CCSID INTERNATIONAL (500).

```
DIR:/home/db2inst1
itso1(db2inst1) $ db2 † connect to ipfvs1a4 user demo using sql34n†

   Database Connection Information

 Database product       = SQL/DS VSE 3.5.0
 SQL authorization ID   = DEMO
 Local database alias   = IPFVS1A4

DIR:/home/db2inst1
itso1(db2inst1) $
```

Figure 253. Connect to SQL/DS on VSE

B.3.6 Display Sessions

```
                                  COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

              Local             Partner        Mode      Link
 CGID        LU name            LU name        name     station     State
------ ----------------- ----------------- -------- ----------- ----------
6      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA21CD IBMRDBM  IPFL3174    Available
5      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA21CD IBMRDBM  IPFL3174    Allocated
4      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA21CD SNASVCMG IPFL3174    Available
3      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA21CD SNASVCMG IPFL3174    Available
2      DEIBMIPF.IPFP221B DEIBMIPF.IPFCP200 CPSVCMG  IPFL3174    Available
1      DEIBMIPF.IPFP221B DEIBMIPF.IPFCP200 CPSVCMG  IPFL3174    Available


F1=Help              F2=Refresh           F3=Cancel            F6=Command
F8=Image             F9=Shell             F10=Exit
```

Figure 254. Display Sessions after First Connect

B.3.7 Database Connection with Corresponding CCSID

The connection is made from ITSO1 with code page IBM-850 to SQL/DS on VM (Böblingen) with CCSID INTERNATIONAL (500).

```
DIR:/home/db2inst1
itso1(db2inst1) $ db2 †connect to ipfvm1a4 user drdax1 using itso10bb†

   Database Connection Information

 Database product       = SQL/DS VM 3.5.0
 SQL authorization ID   = DRDAX1
 Local database alias   = IPFVM1A4

DIR:/home/db2inst1
itso1(db2inst1) $
```

Figure 255. Connect from ITSO1 to SQL/DS on VM

B.3.8 Display Sessions

```
                                  COMMAND STATUS

Command: OK             stdout: yes            stderr: no

Before command completion, additional instructions may appear below.

•TOP“
             Local              Partner         Mode      Link
 CGID       LU name             LU name         name     station     State
------ ----------------- ----------------- -------- ----------- ----------
27     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Allocated
26     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
25     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
24     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
23     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
22     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
21     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
20     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
19     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
18     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
17     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
16     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
15     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
14     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
13     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
12     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
11     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
10     DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
9      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
8      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 IBMRDBM  IPFL3174    Available
```

Figure 256 (Part 1 of 2). Display Sessions after Second Connect

```
7      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA2GL4 SNASVCMG IPFL3174    Available
6      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA21CD IBMRDBM  IPFL3174    Available
5      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA21CD IBMRDBM  IPFL3174    Allocated
4      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA21CD SNASVCMG IPFL3174    Available
3      DEIBMIPF.IPFBOEDJ DEIBMIPF.IPFA21CD SNASVCMG IPFL3174    Available
2      DEIBMIPF.IPFP221B DEIBMIPF.IPFCP200 CPSVCMG  IPFL3174    Available
1      DEIBMIPF.IPFP221B DEIBMIPF.IPFCP200 CPSVCMG  IPFL3174    Available

•BOTTOM“

F1=Help            F2=Refresh          F3=Cancel           F6=Command
F8=Image           F9=Shell            F10=Exit
```

Figure 256 (Part 2 of 2). Display Sessions after Second Connect

B.3.9 Database Connection with not Corresponding CCSID

The CCSID for the SQL/DS database on VSE is now changed from 500 to 870 (Slovenian). Then the database connection from ITSO1 to SQL/DS on VSE is tried again.

```
DIR:/home/db2inst1
itso1(db2inst1) $ db2 † connect to ipfvs1a4 user demo using sql34n†
SQL0332N  There is no available conversion for the source code page †870† to
the target code page †850†.  Reason Code †1†.  SQLSTATE=57017
DIR:/home/db2inst1
itso1(db2inst1) $
```

Figure 257. SQL Message when Using Not Corresponding CCSIDs

Appendix C. Sample Code Pages

C.1 Sample Latin-1 Code Pages

C.1.1 Sample English Code Page

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	µ SM170000	^ SD150000	{ SM110000	} SM140000	\ SM070000	0 ND100000
-1	(RSP) SP300000	é LE110000	/ SP120000	É LE120000	a LA010000	j LJ010000	~ SD190000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	· SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	§ SM240000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ì LI130000	Ç LC420000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	` SD130000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	¢ SC040000	! SP020000	¦ SM650000	: SP130000	« SP170000	ª SM210000	¡ SP030000	[SM060000	(SHY) SP320000	^{1} ND011000	^{2} ND021000	^{3} ND031000
-B	. SP110000	$ SC030000	, SP080000	# SM010000	» SP180000	º SM200000	¿ SP160000	] SM080000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	@ SM050000	ð LD630000	æ LA510000	Đ LD620000	¯ SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	| SM130000	¬ SM660000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00037

C.1.2 Sample Austrian and German Code Page

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	µ SM170000	¢ SC040000	ä LA170000	ü LU170000	Ö LO180000	0 ND100000
-1	(RSP) SP300000	é LE110000	/ SP120000	É LE120000	a LA010000	j LJ010000	ß LS610000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	{ SM110000	ë LE170000	[SM060000	Ë LE180000	c LC010000	l LL010000	t LT010000	· SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	@ SM050000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ì LI130000	Ç LC420000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	~ SD190000	Ñ LN200000	` SD130000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	Ä LA180000	Ü LU180000	ö LO170000	: SP130000	« SP170000	ª SM210000	¡ SP030000	¬ SM660000	(SHY) SP320000	¹ ND011000	² ND021000	³ ND031000
-B	. SP110000	$ SC030000	, SP080000	# SM010000	» SP180000	º SM200000	¿ SP160000	| SM130000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	§ SM240000	ð LD630000	æ LA510000	Ð LD620000	¯ SM150000	¦ SM650000	} SM140000	\ SM070000	] SM080000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	! SP020000	^ SD150000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00273

C.1.3 Sample Brazilian Code Page

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	µ SM170000	¢ SC040000	õ LO190000	é LE110000	\ SM070000	0 ND100000
-1	(RSP) SP300000	} SM140000	/ SP120000	[SM060000	a LA010000	j LJ010000	~ SD190000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	· SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	§ SM240000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	` SD130000	î LI150000	@ SM050000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	¦ SM650000	ì LI130000	] SM080000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	ã LA190000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	É LE120000	$ SC030000	ç LC410000	: SP130000	« SP170000	ª SM210000	¡ SP030000	¬ SM660000	(SHY) SP320000	¹ ND011000	² ND021000	³ ND031000
-B	. SP110000	Ç LC420000	, SP080000	Õ LO200000	» SP180000	º SM200000	¿ SP160000	| SM130000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	Ã LA200000	ð LD630000	æ LA510000	Đ LD620000	¯ SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	! SP020000	^ SD150000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	{ SM110000	ÿ LY170000	# SM010000	(EO)

Code Page 00275

C.1.4 Sample Italian Code Page

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	[SM060000	µ SM170000	¢ SC040000	à LA130000	è LE130000	ç LC410000	0 ND100000
-1	(RSP) SP300000	] SM080000	/ SP120000	É LE120000	a LA010000	j LJ010000	ì LI130000	# SM010000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	· SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	{ SM110000	} SM140000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	@ SM050000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	\ SM070000	~ SD190000	Ç LC420000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	ù LU130000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	° SM190000	é LE110000	ò LO130000	: SP130000	« SP170000	ª SM210000	¡ SP030000	¬ SM660000	(SHY) SP320000	¹ ND011000	² ND021000	³ ND031000
-B	. SP110000	$ SC030000	, SP080000	£ SC020000	» SP180000	º SM200000	¿ SP160000	| SM130000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	§ SM240000	ð LD630000	æ LA510000	Ð LD620000	¯ SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	¦ SM650000	` SD130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	! SP020000	^ SD150000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00280

C.1.5 Sample Spanish Code Page

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	µ SM170000	¢ SC040000	{ SM110000	} SM140000	\ SM070000	0 ND100000
-1	(RSP) SP300000	é LE110000	/ SP120000	É LE120000	a LA010000	j LJ010000	¨ SD170000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	· SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	§ SM240000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ì LI130000	Ç LC420000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	¦ SM650000	ß LS610000	# SM010000	` SD130000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	[SM060000	] SM080000	ñ LN190000	: SP130000	« SP170000	ª SM210000	¡ SP030000	^ SD150000	(SHY) SP320000	¹ ND011000	² ND021000	³ ND031000
-B	. SP110000	$ SC030000	, SP080000	Ñ LN200000	» SP180000	º SM200000	¿ SP160000	! SP020000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	@ SM050000	ð LD630000	æ LA510000	Đ LD620000	¯ SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	~ SD190000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	| SM130000	¬ SM660000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00284

C.1.6 Sample French Code Page

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	[SM060000	` SD130000	¢ SC040000	é LE110000	è LE130000	ç LC410000	0 ND100000
-1	(RSP) SP300000	{ SM110000	/ SP120000	É LE120000	a LA010000	j LJ010000	¨ SD170000	# SM010000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	· SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	@ SM050000	} SM140000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	] SM080000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	\ SM070000	ì LI130000	Ç LC420000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	µ SM170000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	° SM190000	§ SM240000	ù LU130000	: SP130000	« SP170000	ª SM210000	¡ SP030000	¬ SM660000	(SHY) SP320000	¹ ND011000	² ND021000	³ ND031000
-B	. SP110000	$ SC030000	, SP080000	£ SC020000	» SP180000	º SM200000	¿ SP160000	| SM130000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	à LA130000	ð LD630000	æ LA510000	Ð LD620000	¯ SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	~ SD190000	ò LO130000	¦ SM650000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	! SP020000	^ SD150000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00297

C.1.7 Sample English PC Code Page

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	╨ SF460000	α GA010000	≡ SA480000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	æ LA510000	í LI110000	▒ SF150000	┴ SF070000	╤ SF470000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Æ LA520000	ó LO110000	▓ SF160000	┬ SF060000	╥ SF480000	Γ GG020000	≥ SA530000
-3	♥ SS020000	‼ SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	╙ SF490000	π GP010000	≤ SA520000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ä LA170000	ö LO170000	ñ LN190000	┤ SF090000	─ SF100000	╘ SF500000	Σ GS020000	⌠ SS260000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	ò LO130000	Ñ LN200000	╡ SF190000	┼ SF050000	╒ SF510000	σ GS010000	⌡ SS270000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	å LA270000	û LU150000	ª SM210000	╢ SF200000	╞ SF360000	╓ SF520000	µ GM010000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	ù LU130000	º SM200000	╖ SF210000	╟ SF370000	╫ SF530000	τ GT010000	≈ SA700000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	ÿ LY170000	¿ SP160000	╕ SF220000	╚ SF380000	╪ SF540000	Φ GF020000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ö LO180000	⌐ SM680000	╣ SF230000	╔ SF390000	┘ SF040000	Θ GT620000	∙ SA790000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	║ SF240000	╩ SF400000	┌ SF010000	Ω GO320000	· SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	ï LI170000	¢ SC040000	½ NF010000	╗ SF250000	╦ SF410000	█ SF610000	δ GD010000	√ SA800000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	î LI150000	£ SC020000	¼ NF040000	╝ SF260000	╠ SF420000	▄ SF570000	∞ SA450000	^{n} LN011000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	ì LI130000	¥ SC050000	¡ SP030000	╜ SF270000	═ SF430000	▌ SF580000	ϕ GF010001	^{2} ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ä LA180000	₧ SC060000	« SP170000	╛ SF280000	╬ SF440000	▐ SF590000	ε GE010000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Å LA280000	ƒ SC070000	» SP180000	┐ SF030000	╧ SF450000	▀ SF600000	∩ SA380000	(RSP) SP300000

Code Page 00437

C.1.8 Sample International Code Page

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	µ SM170000	¢ SC040000	{ SM110000	} SM140000	\ SM070000	0 ND100000
-1	(RSP) SP300000	é LE110000	/ SP120000	É LE120000	a LA010000	j LJ010000	~ SD190000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	· SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	§ SM240000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ì LI130000	Ç LC420000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	` SD130000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	[SM060000	] SM080000	¦ SM650000	: SP130000	« SP170000	ª SM210000	¡ SP030000	¬ SM660000	(SHY) SP320000	¹ ND011000	² ND021000	³ ND031000
-B	. SP110000	$ SC030000	, SP080000	# SM010000	» SP180000	º SM200000	¿ SP160000	| SM130000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	@ SM050000	ð LD630000	æ LA510000	Đ LD620000	¯ SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	! SP020000	^ SD150000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00500

C.2 Sample Latin-2 Code Page

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ˇ SD210000	˘ SD230000	° SM190000	ą LA430000	˙ SD290000	{ SM110000	} SM140000	\ SM070000	0 ND100000
-1	(RSP) SP300000	é LE110000	/ SP120000	É LE120000	a LA010000	j LJ010000	~ SD190000	Ą LA440000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ę LE430000	Â LA160000	Ę LE440000	b LB010000	k LK010000	s LS010000	ż LZ290000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	Ţ LT420000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	ţ LT410000	ů LU270000	˝ SD250000	Ů LU280000	d LD010000	m LM010000	u LU010000	Ż LZ300000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	§ SM240000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ă LA230000	î LI150000	Ă LA240000	Î LI160000	f LF010000	o LO010000	w LW010000	ž LZ210000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	č LC210000	ľ LL210000	Č LC220000	Ľ LL220000	g LG010000	p LP010000	x LX010000	ź LZ110000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ĺ LL110000	Ç LC420000	Ĺ LL120000	h LH010000	q LQ010000	y LY010000	Ž LZ220000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ć LC110000	ß LS610000	Ć LC120000	` SD130000	i LI010000	r LR010000	z LZ010000	Ź LZ120000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	[SM060000	] SM080000	| SM130000	: SP130000	ś LS110000	ł LL610000	Ś LS120000	Ł LL620000	(SHY) SP320000	Ě LE220000	ď LD210000	Ď LD220000
-B	. SP110000	$ SC030000	, SP080000	# SM010000	ň LN210000	ń LN110000	Ň LN220000	Ń LN120000	ô LO150000	ű LU250000	Ô LO160000	Ű LU260000
-C	< SA030000	* SM040000	% SM020000	@ SM050000	đ LD610000	š LS210000	Đ LD620000	Š LS220000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ŕ LR110000	ť LT210000	Ŕ LR120000	Ť LT220000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	ř LR210000	˛ SD430000	Ř LR220000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	! SP020000	^ SD150000	? SP150000	" SP040000	ş LS410000	¤ SC010000	Ş LS420000	× SA070000	ő LO250000	ě LE210000	Ő LO260000	(EO)

Code Page 00870

Appendix D. Special Notices

This publication is intended to provide customers and IBM systems engineers with guidance in the setup and usage of multi-language in C/S environments. The information in this publication is not intended as the specification of any programming interfaces that are provided by SQL/DS 3.5, DB2/6000 and DB2/2.

See the PUBLICATIONS section of the IBM Programming Announcements for SQL/DS 3.5, DB2/6000 and DB2/2 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM	AIX
APPN	CICS
CICS/ESA	CICS/VSE
CUA	Current
DB2	DB2/2
DB2/6000	DRDA
IBM	ILE
IMS	ISSC
OPC	OS/2
OS/390	OS/400
Presentation Manager	PS/2
RACF	RISC System/6000
S/390	SAA
SQL/DS	System/390
VALE	VM/ESA
VSE/ESA	VTAM
3090	

The following terms are trademarks of other companies:

C-bus	Corollary, Inc.
DOS	Microsoft Corporation
HP	Hewlett-Packard Company
Intel	Intel Corporation
PC Direct	Ziff Communications Company (used by IBM Corporation under license)
SX	Intel Corporation
UNIX	X/Open Company Ltd. (registered trademark in the United States and other countries)
Windows, Windows 95 logo	Microsoft Corporation
386	Intel Corporation
486	Intel Corporation

Appendix E. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How To Get ITSO Redbooks" on page 259.

- *Distributed Relational Database Cross Platform Connectivity and Application*, SG24-4311
- *Setup and Usage of SQL/DS in a DRDA Environment*, GG24-3733

E.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

E.3 Other Publications

For a more detailed discussion of the topics covered in this document, please refer to the platform related publications, such as:

- DB2 for MVS
- DB2 for VSE & VM
- DB2 for Common Server (OS/2, AIX, Windows, HP-UX, and Solaris)
- DB2 for OS/400

The following publications are also relevant as further information sources.

- *Distributed Relational Database Architecture Reference*, SC26-4651
- *Distributed Relational Database Architecture Connectivity Guide*, SC26-4783
- *Character Data Representation Architecture Reference and Registry*, SC09-2190
- *Character Data Representation Architecture Level 2*, SC09-1390

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	Internet
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

  Redbooks Home Page http://www.redbooks.ibm.com
  IBM Direct Publications Catalog http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

AIX	Advanced Interactive eXecutive
AMODE	Addressing mode
APAR	Authorized program analysis report
API	Application Program Interface
APPC	Advanced Program-to-Program Communication
APPLID	Application identifier
APPN	Advanced Peer-to-Peer Networking
ASCII	American National Standard Code for Information Interchange
AVS	APPC/VM VTAM support
BIND	Request to activate a session between two LUs
BIT	Binary digit
BLOB	Binary large object
BP	Business partner
CAE	Client Application Enabler
CCSID	Coded character set identifier
CDRA	Character data representation architecture
CDRM	Cross-domain resource manager
CDRSC	Cross-domain resources
CEDA	Resource Definition Online Transaction
CEMT	Master Terminal Transaction
CET	Central European Time
CF	Communications facility
CICS	Customer Information Control System
CLP	Control language procedure
CMS	Conversational Monitor System
CNOS	Change number of sessions
COBOL	COmmon Business Oriented Language
COS	Customer order services
CP	Control point
CPI	Common Programming Interface
CPIC	Common Programming Interface for Communications
CPU	Central Processing Unit
CU	Control Unit
DB	Data base
DBA	Data base administrator
DBCS	Data base control system
DBN	Database name
DBS	Data base support/system
DB2	DATABASE 2
DCB	Data control buffer
DCS	Data communications system Database connection services
DDCS	Distributed database connection services
DE	Data entry
DF	Data facility
DLC	Data link control
DML	Data manipulation language
DML	Dialogue manager link
DOS	Disk Operating System
DP	Data processing
DRDA	Distributed Relational Database Architecture
EBCDIC	Extended binary coded decimal interchange code
EN	End node
ESA	Enterprise Systems Architecture
FD: OCA	Formatted Data Object Content Architecture
GCS	Group Control System
HP-UX	Hewlett-Packard UNIX
HW	Hardware
I/O	Input/output
I/S	Information systems
I/T	Information technology
IBM	International Business Machines
IEEE	Institute of Electrical and Electronics Engineers
IPF	Interactive productivity facility

ISO	International Organization for Standardization
ISQL	Interactive structured query language
ITSO	International Technical Support Organization
IUCV	Inter-User Communication Vehicle
JCL	Job control language
LAN	Local Area Network
LC	Line control
LEN	Low End Network
LINE	An SNA resource related to a link
LRECL	Logical record length
LU	Logical Unit
LUW	Logical units of work
L1	Level one cache
MAC	Medium access control
MACLIB	Macro library
MPTS	Multiple Protocol Transport Services
MVS	Multiple Virtual Storage
NCP	Network control Program
NDIS	Network driver interface specification
NETBIOS	NETwork Basic Input Output System
NETID	Network identification
NETVIEW	Network observation tool
NLS	National Language Support
NMVT	Network management vector transport
NN	Network node
NPM	Network performance monitor
OS/400	Operating System for AS/400
OS2	Operating System 2
PC	Personal Computer
PCCU	Primary communication control unit
PCT	Program control table
PID	Partition identifier
PIU	Path information unit
PM	Performance monitoring
PNET	Power NETworking interface
PS	Personal System
PS/2	IBM Personal System/2
PU	Physical Unit
PVM	Pass through Virtual Machine
R/O	Read-only
RACF	Resource Access Control Facility
RC	Return code
RDBMS	Relational database management system
RDO	Resource definition on-line
REXX	REstructured eXtended eXecutor Language
RISC	Reduced Instruction-Set Computer
RLD	Relocation dictionary
RMODE	Residency mode
RSCS	Remote Spooling Communications Subsystem
RSL	Remote service link
RU	Request/response unit
SAA	Systems Application Architecture
SAP	OSI service access point
SBCS	Single-byte character set
SDK	Software developers kit
SDLC	Synchronous Data Link Control
SID	Session identifier
SIT	System Initialization Table
SLO	Slovenian
SMIT	System Management Interface Tool
SMTP	Simple mail transfer protocol
SNA	Systems Network Architecture
SNI	Switch network interface
SNT	Sign-on table
SQL	Structured Query Language
SQL/DS	Structured Query Language/Data System
SQLCA	SQL communication area
SQLDA	SQL data area
SS	Session services
SSCP	System services control point
SVC	Supervisor call instruction
SW	Software

SYSADM	System administrator
SYSIN	System input stream
TCP/IP	Transmission Control Protocol/Internet Protocol
TDB	Test data base
TG	Transmission group
TIC	Token-ring interface connector/coupler
TP	Transaction program
TPN	Transaction program name
TPNS	Teleprocessing network simulator
UNIX	Operating system developed by Bell Laboratories (trademark)
UPM	User Profile Manager
USERID	USER IDentification
USS	Unformatted system services
VCTCA	Virtual channel-to-channel adapter
VM	Virtual Machine
VM/ESA	Virtual Machine/Enterprise Systems Architecture
VM/VTAM	VM/Virtual Teleprocessing Access Method
VR	Virtual route
VS	Virtual storage
VSE	Virtual Storage Extended
VSE/ESA	Virtual Storage Extended/Enterprise Systems Architecture
VTAM	Virtual Telecommunications Access Method
WAN	Wide Area Network
XID	Exchange identifier

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

T

U

V

W

X

Z

Printed in U.S.A.

SG24-4846-00

Artwork Definitions			
id	**File**	**Page**	**References**
ITLOGO	4846SU	i	i
ITLOGOS	4846SU	i	

Table Definitions			
id	**File**	**Page**	**References**
AX2TTT5	4846VARS	i	229, 229, 229, 229, 229, 229, 229
AX2TTT6	4846VARS	i	229, 229
CH2TD05	4846VARS	i	
CH2TD10	4846VARS	i	
CH2TD20	4846VARS	i	
CH2TD30	4846VARS	i	
CH2TD40	4846VARS	i	
CH3TD10	4846VARS	i	
CH3TD20	4846VARS	i	
CH5TD11	4846VARS	i	106, 106, 106, 128, 137, 137, 137, 137, 146
CH5TD12	4846VARS	i	106, 106, 106, 106, 106, 106, 106, 106, 106, 106, 106, 106, 106, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 137, 137, 137, 137, 137, 137, 137, 137, 137, 137, 137, 137, 137, 146, 146, 146, 146, 146, 146, 146, 146, 146, 146, 146
CH5TD13	4846VARS	i	128, 128, 128, 146, 146, 146
CH6TT10	4846VARS	i	149
CH6TTT1	4846VARS	i	149, 149, 149, 149, 149, 149, 149, 149, 149, 149, 149, 150, 150, 150, 150
CH6TTT2	4846VARS	i	149, 149, 149
CH6TT20	4846VARS	i	150
CH6TTT3	4846VARS	i	150, 150, 150, 150, 150, 150, 150
CH6TTT4	4846VARS	i	150, 150
CH6TTT5	4846VARS	i	
CH6TTT6	4846VARS	i	
CH6TTA5	4846VARS	i	151, 151, 151
CH6TTA6	4846VARS	i	151
SEC1	4846VARS	i	170
SEC2	4846VARS	i	171
CH7TD10	4846VARS	i	
CH7TD20	4846VARS	i	181
CH7TD21	4846VARS	i	181
CH7TD30	4846VARS	i	
CH7TD40	4846VARS	i	
VSCT1	4846VARS	i	
VSCT1H	4846VARS	i	

Figures

id	File	Page	References	
CH1F010	4846CH1	3	1	2, 3
CH1F020	4846CH1	4	2	4
CH1F028	4846CH1	5	3	
CH1F030	4846CH1	6	4	6
CH1F040	4846CH1	7	5	7
CH1F050	4846CH1	8	6	
CH1F055	4846CH1	11	7	10
CH1F060	4846CH1	12	8	12
CH2F010	4846CH2	19	9	18, 19, 19
CH2F020	4846CH2	20	10	19
CH2F025	4846CH2	21	11	
CH2F026	4846CH2	21	12	
CH2F028	4846CH2	22	13	21
CH2F032	4846CH2	23	14	
CH2F030	4846CH2	26	15	
CH2F040	4846CH2	27	16	
CH2F050	4846CH2	27	17	
CH2F060	4846CH2	28	18	
CH2F065	4846CH2	31	19	31
CH2F070	4846CH2	32	20	
CH2F080	4846CH2	32	21	
CH2F090	4846CH2	32	22	
CH2F100	4846CH2	32	23	
CH2F110	4846CH2	32	24	
CH2F120	4846CH2	33	25	
CH2F130	4846CH2	33	26	
CH2F140	4846CH2	33	27	
CH2F150	4846CH2	33	28	
CH2F160	4846CH2	34	29	
CH2F170	4846CH2	36	30	
CH2F180	4846CH2	36	31	
CH2F190	4846CH2	36	32	
CH2F200	4846CH2	44	33	
CH2F210	4846CH2	44	34	
CH2F220	4846CH2	44	35	
CH2F230	4846CH2	44	36	

CH2F240	4846CH2	45	37	
CH2F250	4846CH2	45	38	
CH2F260	4846CH2	46	39	
CH2F270	4846CH2	46	40	
CH2F280	4846CH2	46	41	
CH2F290	4846CH2	46	42	
CH2F300	4846CH2	47	43	
CH2F310	4846CH2	47	44	
CH2F320	4846CH2	47	45	
CH2F330	4846CH2	47	46	
CH2F340	4846CH2	48	47	
CH2F350	4846CH2	48	48	
CH2F360	4846CH2	50	49	
CH3F010	4846CH3	56	50	
CH3F020	4846CH3	56	51	
CH3F030	4846CH3	57	52	
CH3F040	4846CH3	58	53	
CH3F050	4846CH3	58	54	
CH3F060	4846CH3	59	55	
CH3F070	4846CH3	59	56	
CH3F080	4846CH3	59	57	
CH3F090	4846CH3	63	58	95
CH3F110	4846CH3	64	59	
CH3F120	4846CH3	65	60	
CH3F130	4846CH3	66	61	
CH3F140	4846CH3	66	62	
CH3F150	4846CH3	67	63	
CH3F160	4846CH3	67	64	
CH3F180	4846CH3	68	65	
CH3F190	4846CH3	68	66	
CH3F200	4846CH3	69	67	
CH3F210	4846CH3	69	68	
CH3F220	4846CH3	70	69	
CH3F230	4846CH3	70	70	
CH3F240	4846CH3	72	71	72, 87
CH3F250	4846CH3	73	72	87
CH3F270	4846CH3	74	73	87
CH3F290	4846CH3	75	74	87
CH3F320	4846CH3	77	75	
CH3F360	4846CH3	77	76	
CH3F370	4846CH3	78	77	

CH3F380	4846CH3	79	78
CH3F382	4846CH3	80	79
CH3F384	4846CH3	81	80
CH3F390	4846CH3	82	81
CH3F400	4846CH3	82	82
CH4F010	4846CH4	84	83
CH4F020	4846CH4	84	84
CH4F030	4846CH4	84	85
CH4F040	4846CH4	85	86
CH4F050	4846CH4	85	87
CH4F060	4846CH4	86	88
CH4F070	4846CH4	86	89
CH4F120	4846CH4	88	90
CH4F140	4846CH4	88	91
CH4F130	4846CH4	89	92
CH4F150	4846CH4	89	93
CH4F160	4846CH4	90	94
CH4F210	4846CH4	91	95
CH4F250	4846CH4	91	96
CH4F270	4846CH4	92	97
CH4F280	4846CH4	92	98
CH4F290	4846CH4	93	99
CH4F300	4846CH4	94	100
CH4F320	4846CH4	95	101
CH4F330	4846CH4	95	102
CH4F340	4846CH4	96	103
CH4F350	4846CH4	96	104
CH4F360	4846CH4	97	105

82

CH4F370	4846CH4	98	106
CH5F001	4846CH5	99	107
CH5F010	4846CH5	101	108
CH5F020	4846CH5	102	109
CH5F030	4846CH5	103	110
CH5F040	4846CH5	104	111
CH3F530	4846CH5	105	112
CH5F050	4846CH5	107	113
CH5F060	4846CH5	108	114
CH5F080	4846CH5	109	115
CH5F090	4846CH5	110	116
CH5F100	4846CH5	110	117
CH5F110	4846CH5	111	118
CH5F120	4846CH5	111	119
CH5F130	4846CH5	112	120

CH5F140	4846CH5	113	121
CH5F150	4846CH5	114	122
CH5F160	4846CH5	115	123
CH5F170	4846CH5	116	124
CH5F180	4846CH5	117	125
CH5F190	4846CH5	118	126
CH5F200	4846CH5	118	127
CH5F210	4846CH5	119	128
CH5F220	4846CH5	120	129
CH5F230	4846CH5	120	130
CH5F240	4846CH5	121	131
CH5F250	4846CH5	121	132
CH5F260	4846CH5	122	133
CH5F270	4846CH5	123	134
CH5F280	4846CH5	124	135
CH5F290	4846CH5	124	136
CH5F300	4846CH5	125	137
CH5F310	4846CH5	126	138
CH5F320	4846CH5	126	139
CH5F330	4846CH5	127	140
CH5F340	4846CH5	127	141
CH5F350	4846CH5	129	142
CH5F360	4846CH5	129	143
CH5F370	4846CH5	130	144
CH5F380	4846CH5	130	145
CH5F390	4846CH5	131	146
CH5F400	4846CH5	131	147
CH5F410	4846CH5	132	148
CH5F420	4846CH5	132	149
CH5F430	4846CH5	133	150
CH5F450	4846CH5	133	151
CH5F460	4846CH5	134	152
CH5F470	4846CH5	134	153
CH5F480	4846CH5	135	154
CH5F490	4846CH5	135	155
CH5F500	4846CH5	136	156
CH5F510	4846CH5	138	157
CH5F520	4846CH5	138	158
CH5F530	4846CH5	139	159
CH5F540	4846CH5	139	160
CH5F550	4846CH5	140	161
CH5F560	4846CH5	140	162
CH5F570	4846CH5	141	163
CH5F580	4846CH5		

CH5F590	4846CH5	141	164	
CH5F600	4846CH5	142	165	
CH5F610	4846CH5	142	166	
CH5F620	4846CH5	143	167	
CH5F630	4846CH5	143	168	
CH5F640	4846CH5	144	169	
CH5F650	4846CH5	144	170	
CH5F660	4846CH5	145	171	
CH5F670	4846CH5	145	172	
CH6FBIG	4846CH6	146	173	
CH6F241	4846CH6	152	174	
CH6F242	4846CH6	153	175	
CH6F24A	4846CH6	153	176	
CH6F24B	4846CH6	154	177	155, 158
CH6F24C	4846CH6	156	178	156
CH6F24D	4846CH6	157	179	157
CH6F24E	4846CH6	159	180	159
CH6F24F	4846CH6	160	181	160
CH6F24G	4846CH6	161	182	161, 162
CH6F24H	4846CH6	163	183	162
CH6F32A	4846CH6	164	184	164
CH6F32B	4846CH6	166	185	
CH6F32C	4846CH6	167	186	
CH6F51A	4846CH6	168	187	
CH6F51B	4846CH6	173	188	
CH6F51C	4846CH6	173	189	
CH6F52A	4846CH6	174	190	
CH6F52B	4846CH6	174	191	
CH6F600	4846CH6	175	192	
CH7F000	4846CH7	175	193	
CH7F010	4846CH7	179	194	
CH7F020	4846CH7	180	195	177
APP00	4846CH8	181	196	181
APP01	4846CH8	184	197	184
APP04	4846CH8	186	198	
AX1F010	4846AX1	189	199	189
AX1F020	4846AX1	191	200	
		192	201	

AX1F030	4846AX1	192	202	
AX1F040	4846AX1	193	203	136
AX1F050	4846AX1	193	204	136
AX1F060	4846AX1	194	205	
AX1F070	4846AX1	194	206	
AX1F080	4846AX1	195	207	100, 110, 119, 122
AX1F090	4846AX1	195	208	100
AX1F100	4846AX1	196	209	
AX1F110	4846AX1	196	210	
AX1F120	4846AX1	197	211	
AX1F130	4846AX1	197	212	
AX1F140	4846AX1	198	213	
AX1F150	4846AX1	198	214	
AX1F160	4846AX1	198	215	
AX1F170	4846AX1	198	216	
AX1F173	4846AX1	199	217	
AX1F176	4846AX1	199	218	
AX1F190	4846AX1	199	219	
AX1F180	4846AX1	200	220	
AX1F200	4846AX1	207	221	
AX1F210	4846AX1	207	222	
AX1F220	4846AX1	208	223	
AX1F230	4846AX1	213	224	
AX1F240	4846AX1	213	225	
AX1F250	4846AX1	214	226	
AX1F260	4846AX1	220	227	
AX1F270	4846AX1	220	228	
AX1F280	4846AX1	221	229	
AX1F290	4846AX1	225	230	
AX2F201	4846AX2	229	231	
AX2F202	4846AX2	230	232	
AX2F203	4846AX2	231	233	
AX2F204	4846AX2	232	234	
AX2F205	4846AX2	232	235	
AX2F206	4846AX2	232	236	
AX2F207	4846AX2	233	237	
AX2F208	4846AX2	233	238	
AX2F209	4846AX2	233	239	
AX2F210	4846AX2	234	240	
AX2F211	4846AX2	234	241	
AX2F212	4846AX2	235	242	
AX2F213	4846AX2			

		235	243
AX2F214	4846AX2	235	244
AX2F215	4846AX2	235	245
AX2F216	4846AX2	236	246
AX2F217	4846AX2	237	247
AX2F218	4846AX2	238	248
AX2F311	4846AX2	239	249
AX2F312	4846AX2	239	250
AX2F313	4846AX2	240	251
AX2F314	4846AX2	240	252
AX2F315	4846AX2	241	253
AX2F316	4846AX2	241	254
AX2F317	4846AX2	242	255
AX2F318	4846AX2	242	256
AX2F319	4846AX2	243	257

Headings

id	File	Page	References
CH1H000	4846CH1	1	Chapter 1, Introduction xvii
CH1H010	4846CH1	1	1.1, Situation
CH1H020	4846CH1	1	1.2, Objective
CH1H030	4846CH1	2	1.3, The Current Environment
CH1H040	4846CH1	4	1.4, Code Page Considerations
CH1H050	4846CH1	6	1.5, More Configuration Details
CH1H060	4846CH1	6	1.5.1, The ITSO Environment
CH1H070	4846CH1	7	1.5.2, The ZPIZ Customer Environment
CH1H080	4846CH1	7	1.5.3, A Short Description of ZPIZ
CH1H090	4846CH1	8	1.5.4, ZPIZ Project Requirement
CH1H100	4846CH1	9	1.5.5, Project Conclusion
CH1H110	4846CH1	9	1.5.6, Benefits
CH1H120	4846CH1	10	1.6, DRDA Summary
CH1H130	4846CH1	10	1.6.1, What Architectures are Used by DRDA
CH1H140	4846CH1	10	1.6.2, DRDA Terminology
CH1H150	4846CH1	11	1.6.3, Products Required to Support DRDA
CH1H160	4846CH1	11	1.7, Multi-Language Considerations
CH1H170	4846CH1	12	1.8, CDRA Overview
CH1H180	4846CH1	12	1.8.1, Background on Data Interchange
CH1H190	4846CH1	13	1.8.2, Requirements Summary
CH1H200	4846CH1	13	1.8.3, The Solution is Character Data Representation Architecture
CH1H210	4846CH1	15	1.8.4, Architecture Benefits
CH2H000	4846CH2	17	Chapter 2, Multi-Platform Considerations xvii
CH2H100	4846CH2		

		48	Connect to the Databases Again	
CH2H530	4846CH2	48	2.5.3, DB2 for AIX and DRDA Environment	41
CH2H531	4846CH2	48	2.5.3.1, Conversion of CCSIDs with DRDA	
CH2H532	4846CH2	49	2.5.3.2, Supported Conversions with DRDA	
CH2H540	4846CH2	51	2.5.4, Support for Alternate Collating Sequences	38
CH2H542	4846CH2	51	2.5.4.1, Defining Collating Sequence with API sqlecrea	
CH2H543	4846CH2	52	2.5.4.2, Defining Collating Sequence with CREATE DATABASE Command	
CH2H544	4846CH2	52	2.5.4.3, Samples of User-Defined Collating Sequences	34
CH3H000	4846CH3	55	Chapter 3, VM/ESA Implementation	xvii
CH3H010	4846CH3	55	3.1, General Consideration	
CH3H020	4846CH3	57	3.2, ITSO and ZPIZ System Setup	
CH3H030	4846CH3	57	3.2.1, Database Machine SQLUSER (DRDA AR)	
CH3H040	4846CH3	57	3.2.2, UCOMDIR NAMES File	
CH3H050	4846CH3	59	3.2.3, Database Machine SQLVMDB0 and SQLMACH (DRDA AS)	
CH3H060	4846CH3	60	3.3, ITSO and ZPIZ Database Setup	
CH3H070	4846CH3	60	3.3.1, SQL/DS for VM Startup	
CH3H080	4846CH3	61	3.3.2, SQLINIT Usage	
CH3H090	4846CH3	62	3.3.3, Defining Slovenian Character Set	30, 95, 97
CH3H110	4846CH3	67	3.3.4, SQLINIT QUERY Output	
CH3H120	4846CH3	69	3.3.5, SQLGLOB Usage	
CH3H130	4846CH3	69	3.4, AVS/VM Setup	
CH3H140	4846CH3	69	3.4.1, Setting Up AGWPROF Profile	
CH3H150	4846CH3	69	3.4.2, Defining AVS to VTAM	
CH3H160	4846CH3	70	3.5, SQL/DS for VM DRDA Flow	
CH3H170	4846CH3	72	3.5.1, SQL Connect Flow from VM AR to Local VSE AS at ITSO	
CH3H180	4846CH3	73	3.5.2, SQL Connect Flow from VM AR to Local VSE AS at ZPIZ	
CH3H200	4846CH3	74	3.5.3, SQL Connect Flow from ZPIZ VM AR to Remote VSE AS at ITSO	
CH3H220	4846CH3	75	3.5.4, SQL Connect Flow from ZPIZ VM AR to Remote VM AS at ITSO	
CH3H230	4846CH3	76	3.5.5, Various Console Captures	
CH3H240	4846CH3	81	3.5.6, Multi-Language Character Set Conversions	
CH4H000	4846CH4	83	Chapter 4, VSE/ESA Implementation	xvii
CH4H010	4846CH4	83	4.1, ITSO and ZPIZ VSE/ESA System Setup	
CH4H020	4846CH4	84	4.1.1, CICS/VSE Setup	151
CH4H030	4846CH4	86	4.1.2, DBNAME Directory	
CH4H040	4846CH4	87	4.1.3, SQL Connection Flow between ITSO and ZPIZ and Local	
CH4H050	4846CH4	87	4.1.4, CICS Screen Captures	
CH4H110	4846CH4	94	4.1.5, VTAM/VSE Setup	

AX1H020	4846AX1
AX1H030	4846AX1
AX1H040	4846AX1
AX1H050	4846AX1
AX1H060	4846AX1
AX1H070	4846AX1
AX1H080	4846AX1
AX1H090	4846AX1
AX1H100	4846AX1
AX1H110	4846AX1
AX1H120	4846AX1
AX1H130	4846AX1
AX1H140	4846AX1
AX2H000	4846AX2
AX2H100	4846AX2
AX2H110	4846AX2
AX2H120	4846AX2
AX2H130	4846AX2
AX2H140	4846AX2
AX2H150	4846AX2
AX2H160	4846AX2
AX2H170	4846AX2
AX2H180	4846AX2
AX2H200	4846AX2
AX2H210	4846AX2
AX2H220	4846AX2
AX2H230	4846AX2
AX2H300	4846AX2
AX2H310	4846AX2
AX2H320	4846AX2
AX2H330	4846AX2
AX2H340	4846AX2
AX2H350	4846AX2
AX2H360	4846AX2
AX2H370	4846AX2
AX2H380	4846AX2
AX2H390	4846AX2

191	A.1, VM/VTAM Definitions - BOEVMIS2 100
196	A.2, VSE/VTAM Definitions - BOEVMIS2 100
199	A.3, Field Procedure Sample 31, 63, 95
207	A.4, CDRA Conversion Tables Installation
207	A.4.1, CDRA Conversion Table from 500 to 870
207	A.4.2, CDRA Conversion Table from 870 to 500
208	A.4.3, Sample Program to LOAD Conversion Tables under VM
213	A.4.4, Loading Conversion Tables under VM
213	A.4.5, Loading Conversion Tables under VSE
214	A.4.6, Sample Program to LOAD Conversion Tables under VSE
220	A.4.7, Loading Conversion Tables under VSE
220	A.4.8, Preprocessing Sample Conversion Program under VM
221	A.4.9, Connection via 3174 for ITSO and ZPIZ
225	A.4.10, Connection via 3745 for ITSO and ZPIZ
229	Appendix B, Customization and Displays xviii
229	B.1, SNA Server/6000 Customization on ITSO1
229	B.1.1, SNA Global Information and Local LU Profile
229	B.1.2, Token-Ring Adapter
230	B.1.3, Token-Ring Link Station
232	B.1.4, Control Point
232	B.1.5, LU 6.2 Local LU
232	B.1.6, Partner LU 6.2 Location 160
234	B.1.7, LU 6.2 Mode 161
235	B.1.8, LU 6.2 Side Information Profile 162
236	B.2, DB2 for AIX Customization on ITSO1
236	B.2.1, DB2 for AIX Database Directory 166
237	B.2.2, DB2 for AIX DCS Directory 167
238	B.2.3, DB2 for AIX Node Directory 168
239	B.3, Database Connection from ITSO1 to SQL/DS on VM/VSE 174
239	B.3.1, Status of SNA
239	B.3.2, SNA Global Information
240	B.3.3, Active Link Information
240	B.3.4, LU 6.2 Session Information
241	B.3.5, Database Connection with Corresponding CCSID
241	B.3.6, Display Sessions
242	B.3.7, Database Connection with Corresponding CCSID
242	B.3.8, Display Sessions
243	B.3.9, Database Connection with not Corresponding CCSID

AX3H000	4846AX3	245	Appendix C, Sample Code Pages xviii, 4
AX3H100	4846AX3	245	C.1.1, Sample English Code Page
AX3H200	4846AX3	246	C.1.2, Sample Austrian and German Code Page
AX3H300	4846AX3	247	C.1.3, Sample Brazilian Code Page
AX3H400	4846AX3	248	C.1.4, Sample Italian Code Page
AX3H500	4846AX3	249	C.1.5, Sample Spanish Code Page
AX3H700	4846AX3	250	C.1.6, Sample French Code Page
AX3H800	4846AX3	251	C.1.7, Sample English PC Code Page
AX3H900	4846AX3	252	C.1.8, Sample International Code Page
AX3HA00	4846AX3	253	C.2, Sample Latin-2 Code Page
NOTICES	SG244846 SCRIPT	255	Appendix D, Special Notices ii
BIBL	4846BIBL	257	Appendix E, Related Publications
ORDER	REDB$ORD	259	How To Get ITSO Redbooks 257

Index Entries

id	File	Page	References
NETIND	4846VARS	i	(1) network 2, 21, 101, 102, 110, 118, 156, 158
CODPIND	4846VARS	i	(1) code page 4, 5, 24, 25, 26, 28, 32, 39, 39, 40, 42, 49, 50, 81, 99, 147, 178, 245, 253
DRDAIND	4846VARS	i	(1) DRDA 10, 10, 10, 18, 25, 49, 55, 70, 85, 85, 106, 128, 177, 177, 182
CDRAIND	4846VARS	i	(1) CDRA 13, 14, 14, 14, 14, 15, 207
3745IND	4846VARS	i	(1) 3745 100, 225
3174IND	4846VARS	i	(1) 3174 136, 144, 221
SNAIND	4846VARS	i	(1) SNA 10, 105, 105, 111, 118, 119, 122, 124, 124, 131, 163, 182
PROTIND	4846VARS	i	(1) protocol 17, 18, 18, 20, 22, 22, 60, 61, 100, 101, 102, 128, 147, 167, 167, 168
VMIND	4846VARS	i	(1) VM 19, 25, 55, 59, 59, 60, 61, 69, 72, 73, 74, 76, 76, 79, 100, 191
APPCIND	4846VARS	i	(1) APPC 22, 23, 105, 106, 167, 168, 168
SQLIND	4846VARS	i	(1) SQL 25, 26, 27, 28, 31, 60, 60, 61, 61, 61, 62, 67, 69, 71, 72, 81, 95, 96, 125, 167, 183, 184, 184, 184, 185, 189
OS2IND	4846VARS	i	(1) OS/2 32, 32, 32, 34, 99, 101, 101, 105, 106, 136, 137, 183, 183, 183
AIXIND	4846VARS	i	(1) AIX 35, 35, 36, 36, 37, 37, 38, 39, 40, 40, 42, 52, 149, 236, 237, 238
VTAMIND	4846VARS	i	(1) VTAM 69, 69, 73, 94, 100, 100, 102, 105, 105, 110, 112, 112, 122, 136, 156, 157, 191, 196

VSEIND	4846VARS	i	(1) VSE 83, 84, 94, 196
CICSIND	4846VARS	i	(1) CICS 84, 84, 84, 84, 85, 85, 85, 85, 86, 87, 87, 87, 87
LU62IND	4846VARS	i	(1) LU 6.2 84, 105, 105, 125, 131, 157, 158, 162, 168, 232, 232, 234, 235
LUIND	4846VARS	i	(1) LU 100, 118, 119, 119, 136, 136, 155, 155, 156, 158, 158, 232
CM2IND	4846VARS	i	(1) communications manager/2 105, 105, 130, 130, 137
APPNIND	4846VARS	i	(1) APPN 113, 119, 157, 159
RUIND	4846VARS	i	(1) RU 123, 160
DCSIND	4846VARS	i	(1) DCS directory 128, 167, 167, 237
SYSDIND	4846VARS	i	(1) system database directory 130, 165
APPLIND	4846VARS	i	(1) application 11, 11, 20, 39

Tables

id	File	Page	References
CH2T005	4846CH2	24	2 24
CH2T010	4846CH2	38	3 37
CH2T020	4846CH2	41	4 40
CH2T030	4846CH2	43	5 42, 42
CH2T040	4846CH2	49	6 49
CH3T010	4846CH3	60	7
VSCT01	4846CH4	90	8
CH5T010	4846CH5	106	9
CH5T020	4846CH5	128	10
CH5T030	4846CH5	137	11
CH5T040	4846CH5	146	12
CH6TDB1	4846CH6	149	13
CH6TDB2	4846CH6	150	14
CH6TDB3	4846CH6	150	15 150
CH6TDA3	4846CH6	151	16 150
DWORC	4846CH6	170	17 170, 171
DWRC	4846CH6	171	18 171
CH7T010	4846CH7	181	19
CH7T020	4846CH7	182	20
AX2T010	4846AX2		

Processing Options

Runtime values:

Document fileid	SG244846 SCRIPT
Document type	USERDOC
Document style	REDBOOK
Profile	EDFPRF30
Service Level	0029
SCRIPT/VS Release	4.0.0
Date	96.12.04
Time	09:34:58
Device	3820A
Number of Passes	4
Index	YES
SYSVAR D	YES
SYSVAR G	INLINE
SYSVAR S	OFFSET
SYSVAR X	YES

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	YES
DVCF conditions file	(none)
DVCF value 1	(none)
DVCF value 2	(none)
DVCF value 3	(none)
DVCF value 4	(none)
DVCF value 5	(none)
DVCF value 6	(none)
DVCF value 7	(none)
DVCF value 8	(none)
DVCF value 9	(none)
Explode	NO
Figure list on new page	YES
Figure/table number separation	YES
Folio-by-chapter	NO
Head 0 body text	Part
Head 1 body text	Chapter
Head 1 appendix text	Appendix
Hyphenation	NO
Justification	NO
Language	ENGL
Layout	OFF
Leader dots	YES
Master index	(none)
Partial TOC (maximum level)	4
Partial TOC (new page after)	INLINE
Print example id's	NO
Print cross reference page numbers	YES
Process value	(none)
Punctuation move characters	.,
Read cross-reference file	(none)
Running heading/footing rule	NONE
Show index entries	NO
Table of Contents (maximum level)	3
Table list on new page	YES
Title page (draft) alignment	RIGHT
Write cross-reference file	(none)

Imbed Trace

Page 0	4846SU
Page 0	4846VARS
Page 0	REDB$BOE
Page i	REDB$ED1
Page i	4846EDNO
Page i	REDB$ED2
Page xvii	4846ABST
Page xvii	4846ORG
Page xviii	4846ACKS
Page xviii	REDB$COM
Page xx	4846MAIN
Page xx	4846CH1
Page 15	4846CH2
Page 53	4846CH3
Page 82	4846CH4
Page 98	4846CH5
Page 148	4846CH6
Page 175	4846CH7
Page 182	4846CH8
Page 190	4846AX1
Page 228	4846AX2
Page 243	4846AX3
Page 255	4846SPEC
Page 255	REDB$SPE
Page 255	4846TMKS
Page 256	4846BIBL
Page 257	REDB$BIB
Page 258	REDB$ORD
Page 261	4846ABRV