SC28-1379-2
File No. S370-39

System Programming Library: TSO Extensions Installation and Planning Volume 1

Program Product

IBM

Third Edition (September, 1986)

This is a major revision of SC28-1379-1. See the Summary of Amendments following the Contents for a summary of the changes made to this book. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition with Technical Newsletter SN28-5066 applies to TSO Extensions (TSO/E) Release 3, Program Number 5665-285, and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. The previous edition still applies to TSO Extensions Releases 2 and 2.1 and can now be ordered using the temporary order number SQ28-1379. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

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 program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 921, PO Box 390, Poughkeepsie, New York 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This book is intended for system programmers responsible for:

- Planning the installation of TSO/E
- Installing TSO/E
- Doing post-installation updates.

To do these tasks, you must use this book (Volume 1) along with the following:

- The TSO/E Release 3 program directory
- *System Programming Library: TSO Extensions User Exits and Modifications, Volume 2*, SC28-1380
- *System Programming Library: TSO Extensions Command and Macro Reference, Volume 3*, SC28-1381.

In this book, Volume 2 refers to *User Exits and Modifications* and Volume 3 refers to *Command and Macro Reference.*

This book contains the following chapters:

1. Introduction
2. Planning the Installation of TSO/E
3. Modifying TSO/E after Installation
4. Additional Modifications: An Overview
5. Diagnosing Problems in the Information Center Facility.

Chapter 1: Introduction outlines the process of installing TSO/E, and describes the parts of the process covered in this book.

Chapter 2: Planning the Installation of TSO/E lists the steps for installing TSO/E.

Chapter 3: Modifying TSO/E After Installation describes some of the modifications to TSO/E that your installation might require.

Chapter 4: Additional Modifications: An Overview describes the modifications covered in Volume 2.

Chapter 5: Diagnosing Problems in the Information Center Facility describes the aids available for determining problems in the Information Center Facility, and refers to further sources of information.

Associated Publications

The books referred to in this book are:

MVS/370

Refer to *MVS/System Product Version 1 General Information Manual*, GC28-1025, for the order numbers of the following MVS/370 books on the level that you are using:

OS/VS2 System Programming Library: Initialization and Tuning Guide

OS/VS2 System Programming Library: Supervisor

OS/VS2 MVS JCL

OS/VS2 MVS System Commands

OS/VS2 MVS System Programming Library: Job Management

System Programming Library: JES3 Initialization and Tuning

OS/VS2 MVS Planning: Global Resource Serialization

MVS/XA

TSO Extensions Programmer's Guide to the Server-Requester Programming Interface for MVS/Extended Architecture, SC28-1309

Refer to *MVS/System Product Version 2 General Information Manual*, GC28-1118, for the order numbers of the following MVS/XA books on the level that you are using.

MVS/Extended Architecture System Programming Library: Initialization and Tuning

MVS/Extended Architecture System Programming Library: System Macros and Facilities, Volume 1

MVS/Extended Architecture JCL

MVS/Extended Architecture System Programming Library: JES2 Initialization and Tuning

MVS/Extended Architecture System Programming Library: JES3 Initialization and Tuning

MVS/Extended Architecture System Programming Library: JES2 User Modifications and Macros

MVS/Extended Architecture System Programming Library: JES3 User Modifications and Macros

MVS/Extended Architecture Operations: System Commands

MVS/Extended Architecture System Programming Library: System Modifications

MVS/Extended Architecture System Programming Library: System Management Facilities (SMF)

MVS/Extended Architecture Conversion Notebook

MVS/370 and MVS/XA

TSO Extensions CLISTs: Implementation and Reference, SC28-1304

TSO Extensions Command Language Reference, SC28-1307

TSO Extensions Guide to Writing a Terminal Monitor Program or a Command Processor, SC28-1136

| *TSO Extensions Information Center Facility Administrator's Guide*, GC28-1332

TSO Messages, GC28-1310

| *TSO Extensions User's Guide*, SC28-1333

| VM/PC

| *Virtual Machine/Personal Computer User's Guide for MVS/Extended Architecture Host Services*, SC28-1410

Other Products

Advanced Communications Function for TCAM Version 2 Base Installation Guide, SC30-3132

Advanced Communication Function for VTAM Version 2 Planning and Installation Reference, SC27-0610

A Departmental Reporting System Users Guide, SH20-2165

A Departmental Reporting System Systems Guide, LY20-2145

A Departmental Reporting System II Business Graphics, SH20-2658

The Financial Planning System - TSO Systems Guide, LB21-2337

Interactive System Productivity Facility (ISPF) Dialog Management Services, SC34-2088

Referenced Products

This publication refers to the following products:

1. All references to RACF in this publication indicate the program product **Resource Access Control Facility** (5740-XXH).

2. All references to MVS/System Product Version 2 (MVS/XA) in this publication indicate either **MVS/System Product - JES3** (5665-291) or **MVS/System Product - JES2** (5740-XC6).

3. All references to MVS/System Product Version 1 (MVS/370) in this publication indicate either **MVS/System Product - JES3** (5740-XYN) or **MVS/System Product - JES2** (5840-XYS).

4. All references to VTAM or TSO/VTAM in this publication indicate the program product **ACF/VTAM Version 2** (5665-280).

5. All references to TCAM in this publication indicate the program product **ACF/TCAM Version 2** (5735-RC3).

6. All references to PCF in this publication indicate the program product **Programming Control Facility II** (5798-CLW).

7. All references to APLDI or APLDI-II in this publication indicate the program product **APL Data Interface II** (5796-PNJ).

8. All references to ADRS or ADRS-II in this publication indicate the program product **A Departmental Reporting System II** (5796-PLN).

9. All references to FPS in this publication indicate the program product **Financial Planning System - TSO** (5798-CXP).

10. All references to GDDM/PGF in this publication indicate the program product **Graphical Data Display Manager** (5748-XXH) with **Presentation Graphics Feature** (5748-XXH-02),

11. All references to IIPS/IIAS in this publication indicate the program products **Interactive Instructional Presentation System** (5668-012) and **Interactive Instructional Authoring System** (5668-011).

12. All references to PS/TSO in this publication indicate the program product **Personal Services/TSO** (5665-346).

13. All references to AS in this publication indicate the program product **Application System** (5767-001).

14. All references to VS/APL in this publication indicate the program product **Virtual Storage/A Programming Language** (5748-AP1).

15. All references to APL2 in this publication indicate the program product **A Programming Language 2** (5668-899).

16. All references to Info Center/1 in this publication indicate the program product **Information Center/1** (5668-897).

17. All references to IBM BASIC/MVS in this publication indicate the program product **IBM BASIC/MVS** (5665-948).

18. All references to QMF in this publication indicate the program product **Query Management Facility** (5668-972).

19. All references to TIF in this publication indicate the program product **The Information Facility** (5665-339).

Contents

Figures

Summary of Amendments

Summary of Amendments
for SC28-1379-2
as Updated July 27, 1987
by Technical Newsletter SN28-5066

This Technical Newsletter contains changes to the Information Center Facility names directory. It applies only when the PTF for APAR OY05782 is installed on your system.

Summary of Amendments
for SC28-1379-2
TSO Extensions (TSO/E) Release 3

This edition supports TSO Extensions Release 3. Major technical changes include the following:

- General planning considerations, documented in *TSO Extensions General Information* but also relevant to installation, were added throughout the book, primarily in "Activating TSO/E for the First Time."
- Information about initializing MVSSERV was added (for MVS/XA only).
- The section "Required and Distributed Libraries" replaced the section "Required Libraries" and includes information about converting Information Center Facility libraries from Release 2 to Release 3.
- Sections about space management, printer support, and names service were added.
- Sections about the following user exits were added:
  - ADRS (VS/APL)
  - VM/PC Server Dynamic Allocation
  - Names
  - Termination CLIST.
- A section about defining output descriptors for the ALLOCATE command was added.
- The section "Additional Product Support" includes APL2, VS/APL, IBM BASIC/MVS, QMF, TIF, Info Center/1, PS/TSO, and AS.

- Information about changing the BRODCAST limit, LOGON limits, and EDIT defaults by submitting jobs to perform the updates, rather than through a system generation, was added (for MVS/XA only).
- Information about the JES3 // XMIT statement was added.

Minor technical and editorial changes were made throughout the book.

Summary of Amendments
for SC28-1379-1
TSO Extensions (TSO/E) Release 2.1

This edition supports TSO Extensions Release 2.1. The changes include the following:

- The book has been reorganized. Chapters 4-6 have been consolidated into a single chapter, and Chapter 7 has been incorporated into Chapter 3. A new Introduction has been added.
- The following changes have been made in the section "Information Center Facility" in Chapter 3:
  - The section "Required Libraries" has been updated to clarify the information on LOGON procedures.
  - The section "Additional Product Support" in Chapter 3 has been updated to include GDDM/PGF and IIPS/IIAS.
- The chapter "Information Center Facility — TRACE command" has been changed to include more information for diagnosing problems in the Information Center Facility.

Minor editorial changes have been made throughout the book.

Chapter 1. Introduction

Installing TSO/E requires three major tasks: planning, installing TSO/E on the system, and modifying TSO/E to meet the needs of your installation.

Planning

During the planning stage, you consider questions like the resources TSO/E requires, the additional products you can install with it, and when and how you modify TSO/E. The TSO/E Release 3 program directory, this book, and Volume 2 provide you the information for the installation process.

Installation

During installation, TSO/E is copied from the distribution medium to the host system. The steps for installing TSO/E on a system are listed in Chapter 2, "Planning to Install TSO/E." Detailed information about installing TSO/E is in the TSO/E Release 3 program directory.

Modification

You might need to modify TSO/E to activate functions in TSO/E. For example, if you install a product supported by the Information Center Facility, you must reset a variable to use that product through the Information Center Facility. Your installation might also require additional modifications to TSO/E, such as a LOGON cataloged procedure or an exit routine. You can make these modifications just after installing TSO/E, or at a later time.

Many of these modifications are described in Chapter 3, "Modifying TSO/E After Installation." See Chapter 4, "Additional Modifications: An Overview" for an outline of modifications covered in Volume 2, such as writing exit routines, customizing Information Center Facility functions, and adding subcommands to the EDIT command.

Chapter 2. Planning to Install TSO/E

When you are planning to install TSO/E, you consider whether TSO/E has been installed on the system previously. If a previous release of TSO/E has been installed, you might not need to perform the same steps required to install TSO/E for the first time.

Activating TSO/E for the First Time

If TSO/E has never been installed on your system, perform the following steps to activate the functions of TSO/E.

1. If you are doing a SYSGEN of MVS, and plan to use the Information Center Facility, the Session Manager, or both:

   - To include the Information Center Facility in the SYSGEN, use the SGIKJICQ macro. SGIKJICQ is in SYS1.AGENLIB. For more information, see "Doing an Information Center Facility SYSGEN."

   - To include the Session Manager CLISTs in the SYSGEN, use the SGIKJSM3 macro. SGIKJSM3 is in SYS1.AGENLIB. For more information, see "Including Session Manager CLISTs in a SYSGEN."

2. Reformat the SYS1.UADS, and assign new user attributes.

   To use logon improvements, reformat SYS1.UADS with the UADSREFM program. SYS1.UADS was expanded to 172 bytes per user ID, to hold more logon information; however, the attributes of the data set were not changed. If you do not reformat the UADS, the logon information is not saved across TSO sessions, and users receive default values.

   To assign the RECOVER user attribute, use the ADD or CHANGE subcommand of ACCOUNT. (Users must be authorized by the RECOVER attribute to use the TSO EDIT recovery command.) Also, assign user attributes to installation-supplied TSO default values as follows:

   HOLD - assigns a default value for output class
   JOBCLASS - assigns a default value for jobclass
   MSGCLASS - assigns a default value for message class
   SYSOUT - assigns a default value for SYSOUT class

   For information about the ACCOUNT command, see Volume 3. For more information about reformatting SYS1.UADS, see "Creating, Reformatting, and Updating SYS1.UADS and SYS1.BRODCAST."

3. Reformat the SYS1.BRODCAST data set.

   To reduce channel, control unit, and device busy time when you write a new record to the SYS1.BRODCAST data set, put a free search record in the SYS1.BRODCAST data set. Do this by using the SYNC subcommand of ACCOUNT to reformat the SYS1.BRODCAST data set to include a free search record.

   For information about the ACCOUNT command, see Volume 3. For more information about reformatting SYS1.BRODCAST, see "Creating, Reformatting, and Updating SYS1.UADS and SYS1.BRODCAST."

4. Update HELP data set members.

   The HELP facility of TSO provides help during prompts for command operands. To provide this support for positional parameters, a control character has been defined for the HELP data set. This control character allows requests for help on specific positional parameters as well as keyword operands. To use this enhancement, you must update the HELP data set members for commands that have positional parameters.

   MVS/XA TSO/E includes HELP data set members that support positional parameters for the following commands:

   ```
   ATTRIB  EDIT   OUTPUT
   CALL    EXEC   RUN
   CANCEL  HELP   SEND
   ```

   Note: The TEST command cannot be supported by this HELP enhancement.

   See *Guide to Writing a Terminal Monitor Program or a Command Processor* for more information about updating HELP data set members.

5. If your installation uses RACF, the system does not store user passwords in the terminal status block (TSB). Therefore, you must do one of the following:

   - Change any installation exit or routine that makes use of user passwords in the TSB. For more information, see "LOGON Pre-prompt" and "SUBMIT Command."

   - Force the system to store user passwords in the TSB by overriding the defaults set in the LOGON pre-prompt exit routine.

     Installations that change the defaults and store user passwords in the TSB, which is fetch-protected storage, risk exposing the passwords in dumps that display protected storage. The installation should take appropriate actions to protect these dumps. For more information about the LOGON pre-prompt exit routine, see "LOGON Pre-prompt."

6. Update the tables of APF-authorized commands and programs. For more information, see "Executing Authorized Programs and Commands."

7. Update the SMF CSECT IEEMB846.

For an MVS/XA environment, add the following to CSECT IEEMB846:

AND and OR, as subcommands of the TEST command
UNALLOC, as a new alias of the FREE command
TSOEXEC command
SMPUT command, and its alias SMP
SMCOPY command, and its alias SMC.

You must add the names listed above to IEEMB846 to use these commands, subcommands, and aliases recorded in type 32 SMF records. Any of the names not added to IEEMB846 are recorded as *OTHER in type 32 records. For more information, see *SPL: System Macros and Facilities, Volume 1.*

8. After installing TSO/E, reinstall the Programming Control Facility (PCF) with any applicable service. For more information, see the program directory.

9. Reinstall any installation-written versions of the following exit routines:

   - SUBMIT (IKJEFF10)

   - OUTPUT/STATUS/CANCEL (IKJEFF53)

   - Session Manager exit routines (ADFEXIT1, ADFEXIT2, ADFEXIT3)

   - TMP CSECTs (IKJEFTE2, IKJEFTE8, IKJEFTNS, IKJEFTAP), located in load module IKJTABLS (for MVS/XA only)

   - Interactive Data Transmission Facility installation options CSECT (INMXPARM).

     INMXPARM supplies defaults for the TRANSMIT and RECEIVE commands. IBM provides a dummy INMXPARM that an installation must replace. If you do not replace the IBM-supplied INMXPARM, the TRANSMIT and RECEIVE commands terminate with error messages.

     Member INMINOPT of SYS1.SAMPLIB contains a sample job stream to help you replace the IBM-supplied INMXPARM. Edit the sample job stream to create an SMP user-supplied system modification (USERMOD) that replaces the dummy INMXPARM with your installation's INMXPARM.

     If you are using SMP/E to install TSO/E Release 3 with the MVS/XA feature, update the RECEIVE, APPLY, and ACCEPT steps in the user modification with the SET BDY command.

10. IPL, specifying "clpa" to refresh LPA.

11. Tailor the Information Center Facility. For more information, see "Establishing an Information Center Facility Environment."

    Be aware of the following considerations:

    - If you plan to use printer support, identify available printers and create printer definitions for those printers.

If a printer is not accessed directly through TSO, the installation must also provide a print routine and the administrator must identify that routine when providing information about the printer. For more information, see "Printer Support Considerations."

- Provide a new user ID for any user of education services with an ID of EQ.

12. Tailor the Session Manager. For more information, see "Establishing a Session Manager Environment."

13. Modify any LOGON pre-prompt exit routine that is currently installed. If the exit routine that is currently installed sets the 'NO UADS' bit, change it to set the 'NO UADSE' bit also. The 'NO UADS' bit still prevents LOGON from opening the UADS to verify user attributes. The 'NO UADSE' bit allows LOGON to open the user attributes data set (UADS) only to take advantage of the reduction in I/O operations for the LISTBC command.

    For more information about the LOGON pre-prompt exit routine, see "LOGON Pre-prompt."

14. For an MVS/XA environment:

    - Review compatibility considerations for installation-written subcommands of TEST. (See *MVS/XA Conversion Notebook* for additional information.)
    - Be sure that routines executing in 31-bit addressing mode pass valid 31-bit addresses to the PUTLINE, PUTGET, GETLINE, and STACK service routines. If a routine executing in 31-bit addressing mode passes an address that is below 16 Mb in virtual storage, the high-order byte of that address must be zero.

Installing TSO/E if a Previous Level Was Installed

If you have a previous level of TSO/E installed on your system, review the steps in the preceding section to see if you need to perform any of them. For example, you might have already reformatted SYS1.UADS and SYS1.BRODCAST, and updated the HELP data set members. However, you may need to write and install the Interactive Data Transmission Facility installation options CSECT, INMXPARM, modify the currently-installed LOGON pre-prompt exit routine, and review the compatibility considerations for installation-written subcommands of TEST.

Before installing your new release of TSO/E, review the migration considerations outlined below:

- **Interactive Data Transmission Facility Considerations:** If you installed a previous release of TSO/E and modified member INMINOPT of SYS1.SAMPLIB, you must save a copy of the modified member before installing a new release. INMINOPT is a sample job stream that can be used to install the installation defaults for the TRANSMIT and RECEIVE commands. When you install TSO/E, an IBM-supplied version of

INMINOPT replaces an installation's existing INMINOPT. After installation, you can use the copy of INMINOPT to replace the IBM-supplied INMINOPT.

- **Information Center Facility Considerations:** If you installed the Information Center Facility for TSO/E Release 2, you must convert the tables used in the Information Center Facility names, news, education, and user type services before using those services in TSO/E Release 3. When you convert the tables, you create new tables with different names. Variables are added, updated, or deleted to create the new tables. If you installed and activated the Information Center Facility in TSO/E Release 2, some of the original tables may be deleted. The program directory for TSO/E Release 3 describes those tables. For more information about these tables and their conversion requirements, see "Converting from Release 2 to Release 3."

  The education conversion CLIST copies the course abstracts that an installation provided before Release 3. It also identifies all course names that are longer than 40 characters because course names are limited to 40 characters in TSO/E Release 3. Installations that added course abstracts or renamed courses that IBM shipped in TSO/E Release 2 need to rename the courses whose names are longer than 40 characters. A list of the names installations must change is put into the data set userprefix.V1R3M0.ICQABCRS. If a course that IBM shipped in Release 2 has its original name and that name is longer than 40 characters, the conversion CLIST shortens the name in the course abstract.

- **Broadcast Data Set Considerations:** When the broadcast data set is shared among different releases of TSO/E or TSO, use a TSO/E Release 2 or later system to add notices. Using Release 2 or a later system ensures that all TSO/E systems have a current in-storage list of notices.

Chapter 3. Modifying TSO/E After Installation

Before placing TSO/E into production, review the topics in this chapter to determine some of the changes you might need to make to TSO/E, and how to make those changes.

Initializing TSO/TCAM Time Sharing

If TCAM is the access method TSO is to use, perform the following preliminary steps:

- Tailor the message control program (MCP) to suit the installation's needs. (For more information, see *ACF/TCAM Version 2 Base Installation Guide*.) In addition, refer to "Message Handler (MH) and Message Control Program (MCP) Changes" on page 3-43 for information about changes to the MCP to support a session manager environment.
- Write the MCP cataloged procedure and include it in a procedure library.
- Construct the IKJPRM00 member (or an alternate member) of SYS1.PARMLIB to set terminal I/O coordinator (TIOC) parameters. (For more information, see *SPL: Initialization and Tuning*.) In addition, see "SYS1.PARMLIB Changes" on page 3-41 for information about changes to the SYS1.PARMLIB member to support a session manager environment.
- Write any LOGON cataloged procedures you require and include them in a procedure library. See "LOGON Procedure Changes" on page 3-43 for information about changes to a cataloged procedure to support a session manager environment.
- Include SYS1.CMDLIB in a LNKLSTxx member of SYS1.PARMLIB or in a LOGON cataloged procedure. (For information on LNKLSTxx, see *SPL: Initialization and Tuning*.)
- Create or reformat SYS1.UADS and SYS1.BRODCAST.

Before a terminal user can log on, TCAM must be active in the system and the terminal I/O controller (TIOC) must be initialized. The initialization of the TIOC completes the initialization for the time-sharing subsystem and allows TCAM to accept LOGON commands and pass them to the TIOC for processing.

To start TCAM, the system operator enters the START command as follows:

```
start tcam
```

where TCAM is the name of a procedure that executes the TCAM MCP.

After TCAM has been started, the system operator enters the MODIFY command to activate the TIOC as a subtask of TCAM:

```
modify tcam,ts=start
```

If a parmlib member other than IKJPRM00 is to be used for TIOC parameters, the member name must be included on the MODIFY command. For example:

```
modify tcam,ts=start,ikjprm01
```

For additional information about IKJPRM00, see *System Programming Library: Initialization and Tuning.*

To terminate all time-sharing users' connections with the system, the system operator must issue the MODIFY command:

```
modify tcam,ts=stop
```

For more information on START and MODIFY commands, see *System Commands.*

Writing a Message Control Program Cataloged Procedure

The cataloged procedure used to start an MCP specifies the MCP to be started through the PGM= operand of the EXEC statement. The MCP should be named IEDQTCAM; if a name other than IEDQTCAM is specified, the name must be added to the program properties table (PPT) and must be marked nonswappable. The PPT describes the environment that TCAM requires to operate properly. See "Assigning Special Program Properties" in *System Programming Library: Job Management*, or *System Programming Library: System Modifications.*

The cataloged procedure used to start the MCP also must define the line addresses dedicated to TCAM. This is done by issuing the LINEGRP macro instruction (used in generating the MCP) to specify the ddname of the DD statements that define the communication lines as data sets. For more information, see *ACF/TCAM Version 2 Base Installation Guide.*

Initializing TSO/VTAM Time Sharing

If VTAM is the access method TSO is to use, perform the following preliminary steps:

- Construct the TSOKEY00 member (or an alternate member) of SYS1.PARMLIB to set VTAM terminal I/O coordinator (VTIOC) parameters. (For more information, see *SPL: Initialization and Tuning.*) In addition, see "SYS1.PARMLIB Changes" on page 3-41 for more information about changes to the SYS1.PARMLIB member to support a session manager environment.

- Build translation tables to suit the installation's needs. (See Volume 2 for additional information.)

- Write the cataloged procedure that starts TSO/VTAM and include it in a procedure library.

- Write any editing, attention handling, and nonsupported terminal exit routines the installation needs. (For more information, see *ACF/VTAM Version 2 Planning and Installation Reference.*)

- Write any LOGON cataloged procedures you require and include them in a procedure library. See "LOGON Procedure Changes" on page 3-43 for information about changes to a cataloged procedure to support a session manager environment.

- Include SYS1.CMDLIB in a LNKLSTxx member of SYS1.PARMLIB or in a LOGON cataloged procedure. (For information on LNKLSTxx, see *SPL: Initialization and Tuning.*)

- Create or reformat SYS1.UADS and SYS1.BRODCAST.

Before a terminal user can log on to TSO/VTAM time sharing, both VTAM and the terminal control address space (TCAS) must be active in the system.

The system operator enters the START command to start VTAM. After VTAM has been started, the system operator enters the START command to activate TCAS. TCAS accepts logons from TSO/VTAM users and creates an address space for each user.

When a user logs on, the VTAM terminal I/O coordinator (VTIOC) is initialized. VTIOC controls the movement of data between TSO and VTAM. Parmlib member TSOKEY00 (or an alternate member) contains parameters that are used during VTIOC initialization. If a member other than TSOKEY00 is to be used, the member name may be included on the START command or in the cataloged procedure invoked by the START command. For a description of TSOKEY00 see *System Programming Library: Initialization and Tuning.*

The system operator uses the MODIFY command to modify TSO/VTAM time sharing. The STOP command is used to stop TSO/VTAM time sharing. For more information on the START, MODIFY, and STOP commands as they pertain to TSO/VTAM time sharing, see *System Commands.*

Writing the Procedure That Starts TSO/VTAM Time Sharing

The installation must write a cataloged procedure for starting TSO/VTAM time sharing, and include it either in SYS1.PROCLIB or in an installation-defined procedure library. The cataloged procedure must contain the following statements:

PROC
to name the cataloged procedure and, optionally, to assign default values to symbolic parameters defined in the procedure.

EXEC
to identify the program, IKTCAS00, to be executed.

PARMLIB DD

to identify the parmlib data set and member that contain TSO/VTAM time-sharing parameters. A symbolic parameter can be used for specifying the member name. If it is used, a default value must be specified in the PROC statement. When TSO/VTAM is started, the symbolic parameter receives either the value specified by the system operator on the MEMBER operand of the START command or, if MEMBER is not specified, the default value specified on the PROC statement.

PRINTOUT DD

to identify where the time-sharing parameters that are used should be listed.

A sample procedure for starting TSO/VTAM time sharing is:

```
//TSO        PROC    MBR=TSOKEY00
//STEP1      EXEC    PGM=IKTCAS00,TIME=1440
//PARMLIB    DD      DSN=SYS1.PARMLIB(&MBR),DISP=SHR,
//            FREE=CLOSE
//PRINTOUT   DD      SYSOUT=A,FREE=CLOSE
```

```
   //TSO        PROC    MBR=TSOKEY00
```

The PROC statement assigns the name TSO to the cataloged procedure, which means that the system operator enters START TSO to start TSO/VTAM time sharing. The PROC statement also designates a default parmlib member name, TSOKEY00.

```
   //STEP1      EXEC    PGM=IKTCAS00,TIME=1440
```

The EXEC statement specifies that program IKTCAS00 is to be executed. It also specifies TIME=1440 so that the execution does not have a time limit.

```
   //PARMLIB    DD      DSN=SYS1.PARMLIB(&MBR),DISP=SHR,
   //            FREE=CLOSE
```

The PARMLIB DD statement identifies the parmlib and member that contain time sharing parameters. Specifying &MBR allows the system operator to use the MEMBER operand of the START command to specify a member name; if the system operator does not specify MEMBER, TSOKEY00 is used. The PARMLIB DD statement also specifies DISP=SHR so that another job can use the parmlib data set simultaneously, and FREE=CLOSE so that it is deallocated when it is closed.

```
   //PRINTOUT   DD      SYSOUT=A,FREE=CLOSE
```

The PRINTOUT DD statement specifies that the time sharing parameters used by TSO/VTAM should be written to the device corresponding to class A, and that the output data set should be deallocated when it is closed.

Initializing MVSSERV (MVS/XA Only)

MVSSERV is a TSO command processor that allows IBM host computers (using MVS/XA) and properly-configured IBM personal computers to communicate. This communication enables personal computer (PC) users to access a host computer's resources, using IBM System/370 to IBM Personal Computer Enhanced Connectivity Facilities or VM/PC server support.

If you are to invoke MVSSERV, perform the following preliminary steps:

- Supply and initialize the input parameter data set.
- As an option, supply the diagnostic data sets for MVSSERV.
- Package the servers and initialization/termination programs.
- Install the servers (and initialization/termination programs).
- For VM/PC server support only, install the *TSO/E Release 3 CMS Commands for Host Services* diskette.

Supplying and Initializing the Input Parameter Data Set

To supply the input parameter data set, create the data set and make it available to an MVSSERV user.

> To create the input parameter data set, use the ALLOCATE command. The input parameter data set must have a ddname of CHSPARM, a logical record length of 80, and a fixed or fixed-blocked format.
>
> To make the input parameter data set available to an MVSSERV user, allocate the existing data set in the user's logon procedure, or in a CLIST or ISPF dialog that the user invokes to activate MVSSERV. Be sure the user is authorized to access the input parameter data set.

To make a server available to MVSSERV, initialize the input parameter data set by placing the name of the server's initialization/termination program in the input parameter data set.

> Each record of the input parameter data set must contain the name of one initialization/termination program, starting in column 1. The name must conform to MVS program naming conventions. (The name can have up to eight characters, including A-Z, 0-9, @, #, and $. The first character cannot be 0-9.)

Supplying the Optional Diagnostic Data Sets for MVSSERV

The diagnostic data sets for MVSSERV, and their functions, are as follows:

- **Trace data set** - receives trace data and messages
- **Dump data set** - receives system dump data
- **Dump suppression data set** - prevents dumps for ABEND codes you specify.

Each of these data sets is optional. Including one or more of them only increases your MVSSERV diagnostic information.

To supply each data set, create the data set and make it available to an MVSSERV user.

You can use the ALLOCATE command to create the data sets. To make each of the data sets available to an MVSSERV user, allocate the existing data set in the user's logon procedure, or in a CLIST or ISPF dialog that the user invokes to activate MVSSERV.

Trace Data Set

The trace data set must have a ddname of CHSTRACE, a logical record length of 80, and a fixed or fixed-blocked format. The amount of trace within the data set depends on the trace parameter with which you invoke MVSSERV. Each user should have an individual trace data set. An MVSSERV session causes the trace data set to be rewritten with new results. With an individual trace data set, a user can be sure that the results in the trace data set are from the user's last MVSSERV session.

Use of the trace parameters can affect MVSSERV performance. Therefore, your installation may decide not to use the MVSSERV trace parameters for production work. However, for testing or debugging servers, or requesting diagnostic help from IBM service personnel, you should use MVSSERV with the trace data set and a trace parameter, preferably IOTRACE.

Dump Data Set

The dump data set must have a ddname of SYSUDUMP, SYSMDUMP, or SYSABEND. (The *MVS/Extended Architecture JCL* book explains the difference between SYSABEND, SYSMDUMP, and SYSUDUMP.) The contents of the dump depends on the default options you specify in your SYS1.PARMLIB members SYSUDUMP, SYSMDUMP, and SYSABEND.

Dump Suppression Data Set

The dump suppression data set must have a ddname of CHSABEND, a logical record length of 80, and a fixed or fixed-blocked format.

To suppress a dump, initialize one of the dump suppression data set's 80-byte records. Each 80-byte record of the dump suppression data set must be in the following format:

OFFSET	LENGTH	DESCRIPTION
+0	3	EBCDIC ABEND code in hex. for system ABENDs in decimal for user ABENDs
+3	1	Reserved
+4	4	EBCDIC Reason code (hex)
+8	1	Reserved
+9	1	EBCDIC dump action field: 0 = Do not dump 1 = Dump
+10	71	Reserved

Use leading zeros for the ABEND and reason codes as needed. For example, to suppress dumps from ABENDs of the OPEN macro (ABEND code 913) caused by RACF authorization failure (reason code 38), type the following on a line of the dump suppression data set:

```
|...+....1....+....2....+....3....+....4....+....5...
913 0038 0
```

You can use X's to signify "all values." For example, to suppress dumps from *all* ABENDs of the OPEN macro, type the following:

```
|...+....1....+....2....+....3....+....4....+....5...
913 XXXX 0
```

By placing a '1' in the EBCDIC dump action field of a dump suppression data set line, a user receives the dump specified on the same line. If a dump is used only in certain cases, this option is useful. A user can initialize the dump suppression data set to suppress a dump, and code a '1' in the appropriate EBCDIC dump action field whenever the dump is needed.

For a list of ABEND and reason codes, see *MVS/XA Message Library: System Codes* and *MVS/XA Message Library: System Messages.*

Packaging Servers and Initialization/Termination Programs

You can package a server and its installation/termination program as CSECTs of the same load module, or as CSECTs in different load modules. The main consideration is server loading:

- If you do not want the initialization/termination program to load the server, place the initialization/termination program and server in the **same** load module The program can use a constant server address to define the server to MVSSERV.

- If you want the initialization/termination program to load the server, place the initialization/termination program and server in **different** load modules. The program can get the server address from the LOAD macro to define the server to MVSSERV.

Installing Servers (and Initialization/Termination Programs)

You can install a server in one of two ways:

- In a STEPLIB, in a user's logon procedure. This method of installation lets you restrict the server to a specific user or users, and is especially useful for testing a new server.

  The server must be in a load module. If the server's initialization/termination program is being tested, and it is not in the same load module as the server, you must also allocate it to a STEPLIB.

- In a system library, in the linklist concatenation. This method of installation lets you make a server available to system users.

  If you have tested a server initially in a STEPLIB, you can copy it to a new or existing member of a system library. If the server's initialization/termination program is in a different load module, be sure that module is also in the system library.

Installing the "TSO/E Release 3 CMS Commands for Host Services" Diskette (For VM/PC Server Support Only)

Two CMS commands, DSNMAP and TSO, support the function provided by MVSSERV and are on the *TSO/E Release 3 CMS Commands for Host Services* diskette as a VM/PC minidisk file. To install the diskette, complete the following steps:

1. Copy the file TSO.200 from the diskette to the DOS directory that contains VM/PC.

2. Start the VM/PC configurator, VMPCCON, to create a TSO user ID of TSO and to define a new minidisk at address 200.

   a. Enter your password at the configurator logo.

   b. Hit PF3 to update the user configuration.

   c. Create a new user.

      - Environment . . . User ID = TSO
        Password = TSO
        Storage = 512
        Auto IPL = (blank)

      - Minidisks . . . . . Address = 200
        Drive = C
        Size = 40 blocks
        Access = WRITE

      - Links . . . . Not necessar for
        User ID = TSO

   d. Hit PF5 to get to the User Selection Menu.

   e. Select the user ID you use to log on to VM/PC.

   f. Hit PF4 to set up a user link.

      - User ID = TSO
      - Virtual address = 200
      - Minidisk address = 200
      - Access = READ only

   g. End the configurator, saving all changes.

3. At DOS, issue the VMPC command.

4. Log on to VM/PC.

5. Access the TSO minidisk with the ACC 200 Z command.

To use the CMS commands TSO and DSNMAP, you must access address 200 at the beginning of each terminal session, unless the text files in the TSO.200 minidisk are copied to your A disk. If these files reside on the A disk, you automatically have access to the DSNMAP and TSO commands at each terminal session. Therefore, you do not need to access the TSO.200 minidisk. However, if the files are left on the TSO.200 minidisk, maintenance changes can be made to one data set and remain available to all VM/PC users.

For more information about MVSSERV, see *TSO Extensions Programmer's Guide to the Server-Requester Programming Interface for MVS/Extended Architecture.* For more VM/PC information, see *Virtual Machine/Personal Computer User's Guide for MVS/Extended Architecture Host Services.*

Writing a LOGON Cataloged Procedure

A LOGON cataloged procedure does the following tasks:

- It defines the system resources available to a terminal user.

- It defines or allows the dynamic allocation of all data sets used by a terminal user.

- It specifies which program is to be invoked after LOGON. This can be the terminal monitor program (TMP) distributed with TSO/E or an installation-written program.

You can define a LOGON cataloged procedure in:

- The PROC operand of the LOGON command

- An installation exit from the LOGON processor

- The entry for the userid in the UADS.

If TSO/VTAM is being used, you can also specify a LOGON cataloged procedure in:

- The data field of a VTAM LOGON command
- A VTAM unformatted system services (USS) definition table
- A VTAM interpret table.

If more than one procedure is defined for a userid/password combination, the procedure must be specified on the PROC operand of the LOGON command.

For LOGON procedures to reside in a separate library:

1. Code a PROCxx DD statement for the library in the JES2 or JES3 procedure.

2. For JES2, specify xx in the PROCLIB= parameter in the &TSU initialization parameter.

   For JES3, specify xx in the TSOPROC= parameter on the STANDARDS initialization statement. (For additional information about initialization parameters, see to *System Programming Library: JES2 Initialization and Tuning* or *System Programming Library: JES3 Initialization and Tuning.*)

   LOGON cataloged procedures must reside in the data set defined in the procedure used to start the primary job entry subsystem. This data set may be either SYS1.PROCLIB or a partitioned data set dedicated to LOGON procedures.

Determining TSO User Region Size

TSO uses the following search order (listed 1 through 5) to obtain the region size to be allocated to a TSO user:

Search Order	Effective Region Size
1	LOGON pre-prompt exit can specify the size.
2	Size operand of the LOGON command.
3	UADS size (as specified by the ACCOUNT command).
4	The REGION parameter on the EXEC statement in the user's LOGON procedure.
5	&TSU parameter with subparameter CONVPARM specifying a default region size.

Notes:

1. *All these effective region sizes are used in conjunction with the IEALIMIT and IEFUSI controls. (See to System Modifications for details about IEALIMIT and IEFUSI controls in MVS/XA. See SPL: Supervisor for details about IEALIMIT control in MVS/370.)*

2. *The region size value specified for locations 1, 2, and 3 (in the search order) cannot exceed the UADS MAXSIZE, except when a logon pre-prompt exit indicates to LOGON to ignore the UADS MAXSIZE value.*

3. *Once TSO obtains a valid region size value, it stops searching.*

EXEC Parameters for the LOGON Procedure

The TMP distributed with TSO/E is named IKJEFT01. If you are using an installation-written TMP for a particular procedure, you must substitute its module name for IKJEFT01 in the PGM= operand in the EXEC statement. REGION= can be used to limit the amount of virtual storage obtained by variable-length GETMAINs.

DYNAMNBR= is used to calculate the allowed number of concurrent allocations using dynamic allocation. (A constant of 2 is always added to the DYNAMNBR specification.) If DYNAMNBR= is omitted, the number of allocations is determined by the number of DD DYNAM statements. If DD DYNAM statements and DYNAMNBR= are both present, the number of concurrent allocations equals the combined total.

The DYNAMNBR parameter value in the EXEC statement should be carefully chosen. The value should be large enough so that it is not readily exceeded by dynamic allocation requests.

Note that the maximum number of concurrently allocated resources for any TSO session is 1635.

If you need job step timing, include the TIME parameter on the EXEC statement in the LOGON procedure.

Any PARM operand on the EXEC statement is interpreted by the terminal monitor program as the first line of input from the terminal. This input could be a command or the execution of a CLIST.

The EXEC statement is the only statement required in a LOGON procedure.

Optional Data Sets

Data sets needed by a processing program such as a compiler or a system utility can be defined dynamically through the ALLOCATE command or through dynamic allocation.

You can place data sets the user wants for TSO sessions in a LOGON procedure. This technique has three advantages:

1. It allows volumes to be mounted.

2. It provides recovery from an offline device condition. Messages tell the operator to vary the device online using the VARY command.

3. It saves repeated allocation and freeing of the same data set by successive commands in the same TSO session.

Certain DD statements have special meaning and can be included in a LOGON procedure depending upon the installation's needs. They are:

SYSPROC – The SYSPROC DD statement defines the current CLIST library to the EXEC command when the implicit form of the command is used. The data set described by this DD statement must be partitioned with a record format of V, VB, F or FB. This statement can be defined in the LOGON procedure or can be dynamically allocated using the ALLOCATE command.

Sample listing (when used in LOGON procedure)

```
//SYSPROC    DD     DSN=CLIST.PROC.LIB,DISP=SHR
```

Sample terminal input (when used in ALLOCATE command)

```
allocate da('clist.proc.lib')  f(sysproc)  shr
```

STEPLIB – A LOGON procedure can have a private step library defined by a STEPLIB DD statement. This library can contain command processors or a user-written TMP that the installation wants to make available to selected TSO users. (Note: SYS1.CMDLIB can be specified as a step library. However, it is not recommended; it would nullify the improvements that can be obtained by putting command processors in the LPA.) Most TSO users should not have STEPLIB in their LOGON procedure because of the extra search time required for each command and CLIST.

Note: If users of the EDIT command require the ability to save their data sets with the SAVE subcommand, ensure that the applicable DASD volumes are mounted with the attribute of 'STORAGE'.

Sample Procedure

Figure 3-1 shows a sample listing of a LOGON cataloged procedure. The sample LOGON procedure can be useful to a programmer using Assembler H. The statements specify the TSO standard TMP for execution; define the library for the users EXEC commands, the work data sets for the assembler, the CLIST library, and the assembler macro library; and specify that SYSIN and SYSPRINT are to be directed to the user's terminal.

```
//AFPROC     EXEC   PGM=IKJEFT01,DYNAMNBR=7
//SYSUT1     DD     DSN=&SYSUT1,UNIT=SYSDA,SPACE=(1700,(400,50))
//SYSUT2     DD     DSN=&SYSUT2,UNIT=SYSDA,SPACE=(1700,(400,50))
//SYSUT3     DD     DSN=&SYSUT3,UNIT=SYSDA,SPACE=(1700,(400,50))
//SYSPROC    DD     DSN=CLIST.PROC.LIB,DISP=SHR
//SYSLIB     DD     DSN=SYS1.MACLIB,DISP=SHR
//SYSIN      DD     TERM=TS
//SYSPRINT   DD     TERM=TS
```

Figure 3-1. Sample Listing of LOGON Cataloged Procedure

Invoking the Terminal Monitor Program

The terminal monitor program (TMP) is attached as APF-authorized and executes supervisor state and problem program key. It provides an interface between the user, command processors, and the TSO control program. The TMP is invoked in one of two ways:

- Using the LOGON procedure for foreground execution
- Using the EXEC statement in the input stream for background execution

  Note: Any job that invokes the TMP in the background is given the authorization to execute the ACCOUNT command, as long as the 'USER=' field is not specified. For security purposes, RACF-protect or password-protect the UADS, or write a JCL exit to limit access to the background TMP.

Executing the TMP as a Batch Job

The terminal monitor program (TMP) creates a TSO environment so that the TSO service routines, supported command processors, and the access method services utilities can function in the background. The DD statements SYSIN and SYSPRINT are replaced by SYSTSIN (for input) and SYSTSPRT (for output) when executing the TMP as a batch job. Input for a GETLINE is obtained from the data set defined by the SYSTSIN DD statement. The TMP issues the STACK macro instruction to put the SYSTSIN data set on the input stack. The commands that are read from SYSTSIN are logged on SYSTSPRT; therefore, the commands and subcommands can be entered using SYSTSIN. Each command must begin on a separate card.

Messages issued using PUTLINE are written to the data set defined by the SYSTSPRT DD statement. Multilevel informational messages are automatically written out as if you entered a question mark (?). Prompting messages, those that require responses, are not considered informational messages and are not written out. In addition, the "HELP" messages for the prompting messages are not written out.

No prompting is done because the TMP sets options as if the following PROFILE command was issued:

```
profile noprompt
```

Since "no prefixing" of data set names is the default in the background, an unqualified data set name will not be prefixed with a userid. If you want a userid prefixed to the data set names, include the following command:

```
profile prefix(userid)
```

at the beginning of the SYSTSIN stream.

Note: If you have RACF installed and you are defined to it, you can specify USER=userid on the JOB card and eliminate the need for the preceding PROFILE command. (The specified userid is used as the prefix.)

The JCL used to run the TMP as a batch job includes:

1. An EXEC statement that specifies PGM=IKJEFT01. If any command processors will dynamically allocate data sets, specify the DYNAMNBR parameter. If you specify the PARM parameter, its value is interpreted as the first line of input from SYSTSIN.

2. A SYSTSPRT DD statement for commands and subcommands executed, plus messages.

3. A SYSTSIN DD statement for data sets

4. A SYSTSIN DD statement for data sets containing commands and subcommands.

Notes:

1. *The SYSTSIN and SYSTSPRT DD statements may refer to a sequential or partitioned data set. If the data set is partitioned, the member-name must be specified on the DD statement as DSN=pdsname(membername). The SYSTSIN data set cannot be a concatenated data set.*

2. *TSO was designed as a one-step job. Therefore, the TSO LOGON procedure should have only one execute statement per job.*

Creating, Reformatting, and Updating SYS1.UADS and SYS1.BRODCAST

After system generation, two data sets, SYS1.UADS (the UADS) and SYS1.BRODCAST (the broadcast data set), must be available to the system before users can log on to TSO.

- For installations that are installing TSO for the first time, the system programmer must create both data sets.

- For installations that are installing a new release of TSO/E, the system programmer may have to reformat both data sets.

- During subsequent operation, the system programmer may have to update both data sets by adding, changing, or deleting entries. In addition, the system programmer may reformat the data sets to eliminate wasted space caused by the periodic additions, changes, and deletions.

Allocation and Processing Considerations

Your installation should consider the following when allocating or processing the UADS and the broadcast data set:

- Do not allocate either SYS1.UADS or SYS1.BRODCAST on shared DASD that is accessed by more than one processor, unless you use global resource serialization. See "Global Resource Serialization" on page 3-22 for additional information.

- If SYS1.UADS is processed by programs other than the reformatting program (UADSREFM) or the ACCOUNT command, unpredictable results may occur later during reformatting, or during processing using the subcommands of ACCOUNT.

Content and Structure of the UADS

The user attribute data set (UADS) is basically a list of terminal users who are authorized to use TSO. The UADS contains information about each of the users, and is used to regulate access to the system. An entry exists in the UADS for each terminal user. Each entry consists of the following information:

- A user identification.

- One or more passwords, or a single null field, associated with the user identification.

- One or more account numbers, or a single null field, associated with each password.

- One or more procedure names associated with each account number. Each name identifies a procedure that may be invoked when the user enters the LOGON command.

- Additional attribute information as described in Volume 3 under the ACCOUNT command.

The organization of the information contained in the UADS is shown in Figure 3-2. Figure 3-3 shows the simplest structure that an entry in the UADS can have, and Figure 3-4 shows a more complex structure.

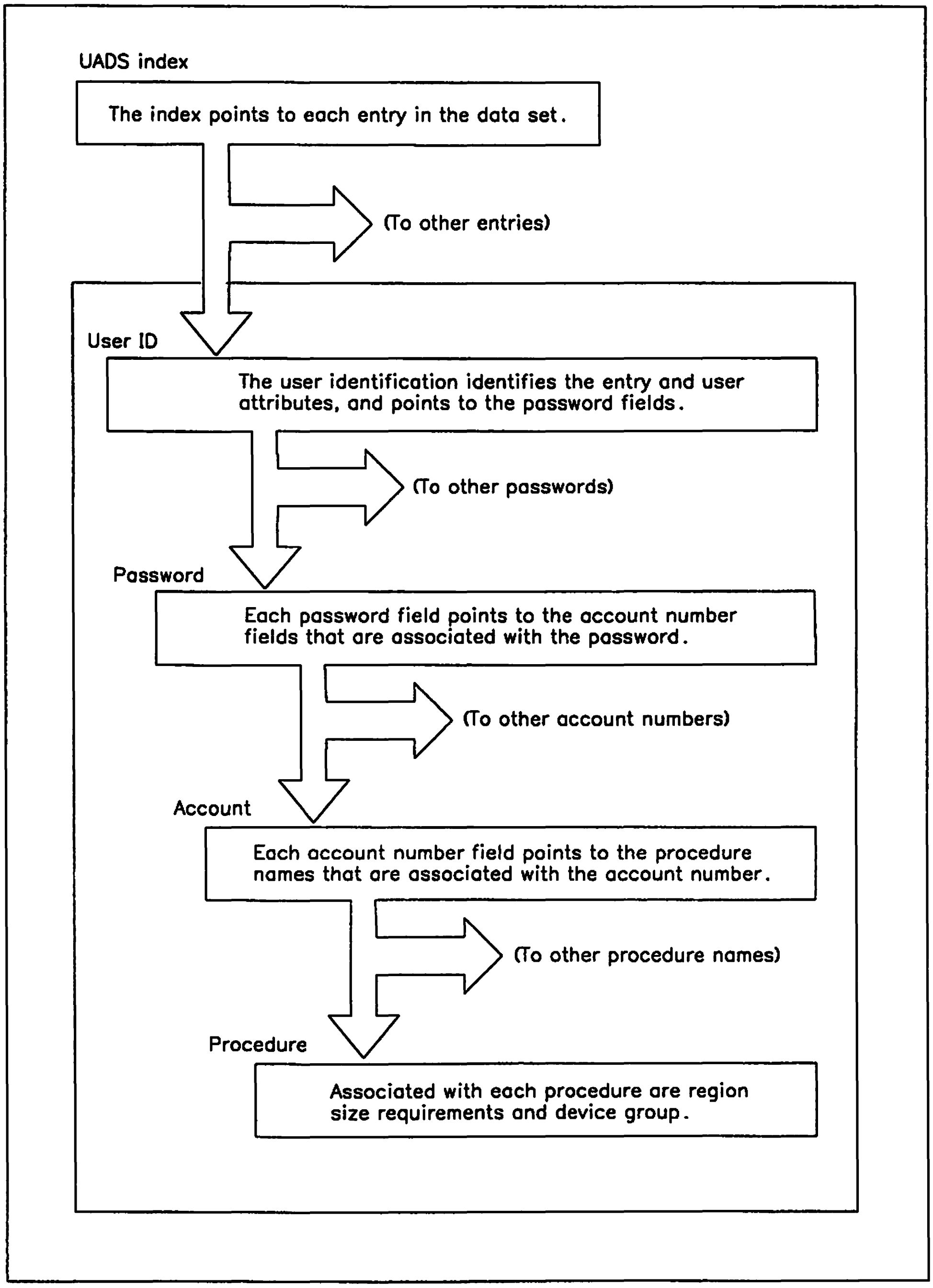
UADS index
The index points to each entry in the data set.
(To other entries)
User ID
The user identification identifies the entry and user attributes, and points to the password fields.
(To other passwords)
Password
Each password field points to the account number fields that are associated with the password.
(To other account numbers)
Account
Each account number field points to the procedure names that are associated with the account number.
(To other procedure names)
Procedure
Associated with each procedure are region size requirements and device group.

Figure 3-2. Organization of the UADS

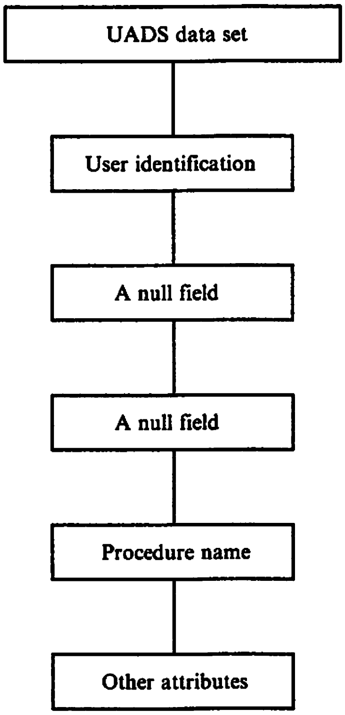

Figure 3-3. The Simplest Structure for a Typical UADS Entry

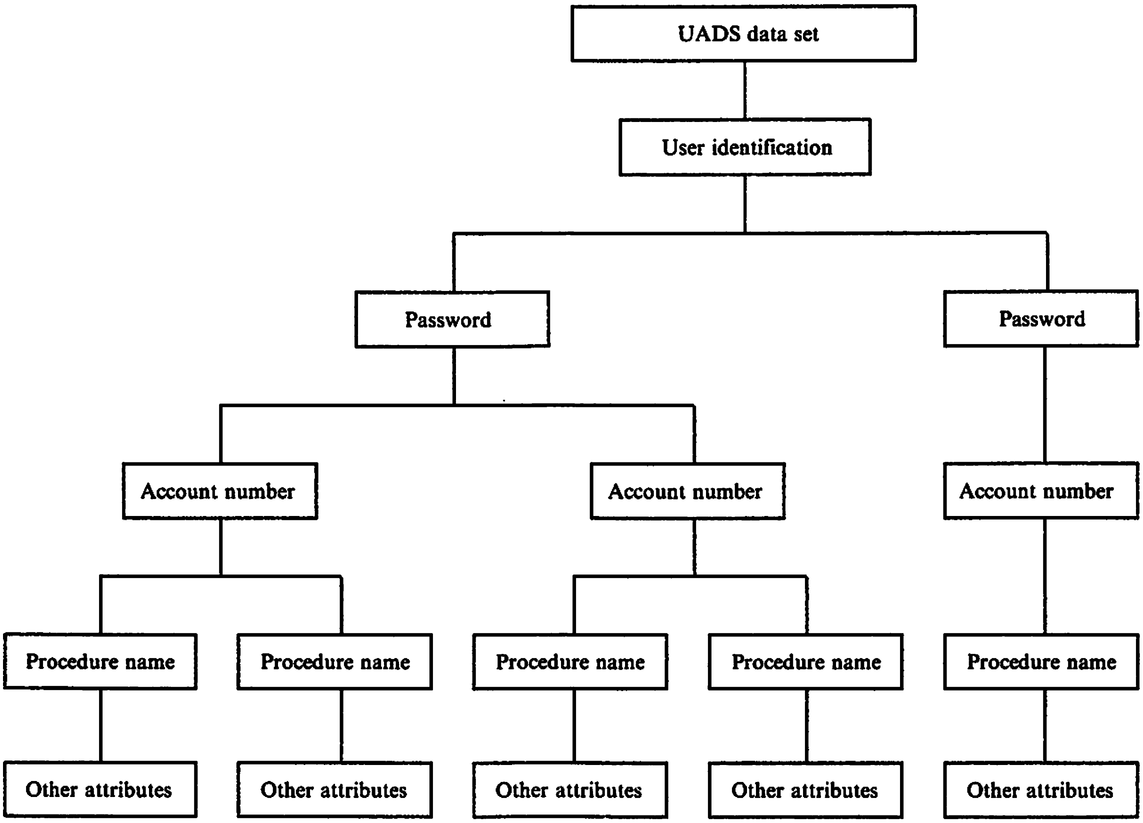

Figure 3-4. A Complex Structure for a Typical UADS Entry

Content of SYS1.BRODCAST

The broadcast data set contains messages intended for terminal users. The messages are sent to the broadcast data set using the SEND command or the SEND subcommand of OPERATOR. The broadcast data set contains two sections: the mail section and the notices section. The mail section contains messages intended for particular users; the notices section contains messages intended for all users.

ACCOUNT Command and Its Subcommands

The ACCOUNT command and its subcommands are used to create and update the entries in the UADS. Specifically, the ACCOUNT command can:

- Add new entries or more data to an existing entry (ADD subcommand)
- Change data in an entry (CHANGE subcommand)
- Delete entries or parts of entries (DELETE subcommand)
- Build a new broadcast data set and synchronize it with an existing UADS (SYNC subcommand).

Creating the UADS and the Broadcast Data Set from a Terminal

To create the UADS and the broadcast data set from a terminal, add to the procedure library a LOGON procedure named IKJACCNT. During system generation, one userid (IBMUSER) is copied into the newly-created UADS. IBMUSER is authorized to use one LOGON procedure, IKJACCNT. A sample IKJACCNT LOGON procedure follows:

```
//IKJACCNT  EXEC   PGM=IKJEFT01,DYNAMNBR=10
//SYSUADS   DD     DSN=new UADS created during sysgen
//SYSLBC    DD     DSN=SYS1.BRODCAST created during sysgen
```

Activate TCAM or VTAM.

Log on using the following command:

```
logon ibmuser nonotices nomail
```

The keywords NONOTICES and NOMAIL prevent the LOGON processor from accessing the broadcast data set before broadcast is formatted. Enter the ACCOUNT command and issue the SYNC subcommand to format a skeleton of the broadcast data set. Issue ADD subcommands to add the new userids to both UADS and broadcast.

Log on again with a new userid that has ACCOUNT authority. Enter the ACCOUNT command and issue the DELETE subcommand to delete the IBMUSER userid.

Creating the UADS and the Broadcast Data Set with a Batch Job

To create the UADS and the broadcast data set without having TSO active, execute the terminal monitor program (TMP) as a batch job. Include the ACCOUNT command and use the SYNC subcommand to format a skeleton of the broadcast data set. Then, as each userid is added to the UADS using the ADD subcommand, a corresponding entry is made in the broadcast data set. After creating the UADS, use the DELETE subcommand to delete IBMUSER (the userid with ACCOUNT authority provided during system generation). Figure 3-5 is a sample listing showing the creation of the UADS and broadcast data set. An explanation of the JCL can be found in "Executing the TMP as a Batch Job" on page 3-13.

```
//jobname      JOB      job card parameters
//             EXEC     PGM=IKJEFT01
//SYSTSPRT     DD       SYSOUT=A
//SYSUADS      DD       DSN=uads created during sysgen
//SYSLBC       DD       DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN      DD       *
ACCOUNT
SYNC
ADD new userid (see ADD subcommand of ACCOUNT for other operands)
 .
 .
 .
DELETE (IBMUSER)
END
/*
```

Figure 3-5. Creating the UADS and the Broadcast Data Set with a Batch Job

Reformatting the UADS and the Broadcast Data Set

There are four major steps you must follow to reformat the UADS and the broadcast data set:

1. Allocate a new UADS
2. Reformat the UADS and the broadcast data set using UADSREFM
3. Reset the UADS catalog entry
4. IPL your system again.

Allocating a New UADS

To allocate a new UADS using ISPF, enter ISPF option 3.2, and allocate a partitioned data set name that is different from the name of the existing UADS. To allocate a new UADS using a batch job, include the data set attributes on the DD statement for the new data set. Figure 3-6 on page 3-20 shows sample job for using a batch job to allocate the new UADS.

```
//jobname   JOB  job card parameters
//          EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=A
//SYSIN     DD   DUMMY
//SYSUT2    DD   DSN=new.UADS.name,UNIT=unit,VOL=SER=volser,
//              DISP=(NEW,CATLG),SPACE=(TRK,(?,?)),
//              DCB=(RECFM=??,LRECL=??,BLKSIZE=????)
//SYSUT1    DD   *
/*
```

Figure 3-6. Allocating the New UADS as a Batch Job

Reformatting the UADS and the Broadcast Data Set Using UADSREFM

To reformat the UADS and the broadcast data set, execute the TMP as a batch job. In that batch job, execute UADSREFM, the UADS reformatting program. UADSREFM reads an entry from the old UADS, builds a logical copy of that entry, eliminates any wasted space, and writes the newly-formatted entry into the new UADS. The process repeats automatically for each entry in the UADS. However, UADSREFM does not reformat an entry if the user is currently logged on. It writes messages indicating which entries were not reformatted.

Note: You can also use the UADSREFM program to change the block size of the UADS.

In the batch job with UADSREFM, include the ACCOUNT command. Use the SYNC subcommand to reformat the broadcast data set. After reformatting, use the DELETE subcommand to delete IBMUSER.

Figure 3-7 is a sample listing showing the reformatting of the UADS and the broadcast data set. For an explanation of this JCL, see "Executing the TMP as a Batch Job" on page 3-13.

```
//jobname      JOB      job card parameters
//             EXEC     PGM=IKJEFT01
//SYSTSPRT     DD       SYSOUT=A
//SYSUADN      DD       DSN=old format uads
//SYSUADS      DD       DSN=reformatted uads
//SYSLBC       DD       DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN      DD       *
UADSREFM
ACCOUNT
DELETE (IBMUSER)
SYNC
END
/*
```

Figure 3-7. Reformatting the UADS and the Broadcast Data Set

Updating the UADS and the Broadcast Data Set: To update the UADS and the broadcast data set from a terminal, ensure that the UADS to be updated is allocated by the SYSUADS DD statement (either in a LOGON procedure or by using an ALLOCATE command). Make the updates (additions, changes, deletions) using the ACCOUNT command and its subcommands.

To update the UADS and the broadcast data set with a batch job, you can use the sample JCL shown in Figure 3-8. If you do not include a DD statement specifying SYS1.BRODCAST, the data set must be cataloged.

```
//jobname     JOB     job card parameters
//            EXEC    PGM=IKJEFT01
//SYSTSPRT    DD      SYSOUT=A
//SYSUADS     DD      DSN=SYS1.UADS
//SYSLBC      DD      DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN     DD      *
ACCOUNT
   .
   .     ACCOUNT subcommands for updating
   .
END
/*
```

Figure 3-8. Sample Listing — Updating the UADS with a Batch Job

Resetting the UADS Catalog Entry

After the UADS has been reformatted, the UADS catalog entry must be reset. To reset the UADS catalog entry using ISPF, enter option 3.2 and delete the old UADS, then rename the new UADS to the name of the old UADS. To reset the UADS catalog entry using a batch job, execute IEHPROGM as shown in Figure 3-9.

```
//jobname  JOB  job card parameters
//         EXEC PGM=IEHPROGM
//SYSPRINT DD   SYSOUT=A
//DD1      DD   UNIT=device,VOL=SER=volser,DISP=OLD
//DD2      DD   UNIT=device,VOL=SER=volser,DISP=SHR
//SYSIN    DD   *
       SCRATCH  DSNAME=old.UADS.name,VOL=device=volser
       UNCATLG  DSNAME=old.UADS.name
       RENAME   DSNAME=new.UADS.name,old.UADS.name,
                  VOL=device=volser
       CATLG    DSNAME=old.UADS.name,VOL=device=volser
       UNCATLG  DSNAME=new.UADS.name
/*
```

Figure 3-9. Resetting the UADS Catalog Entry as a Batch Job

The batch job in Figure 3-9:

1. Deletes the old UADS (SCRATCH)
2. Uncatalogs the old UADS (UNCATLG)
3. Renames the new UADS to the old UADS name (RENAME)

4. Catalogs the new UADS under the old UADS name (CATLG)
5. Uncatalogs the old UADS name (UNCATLG).

Global Resource Serialization

An installation can place a single version of both SYS1.UADS and SYS1.BRODCAST on a shared DASD and access each one from any system in a multi-system complex by using global resource serialization (that is, the resources - SYS1.UADS and SYS1.BRODCAST - may be globally shared). However, to ensure that an installation can evaluate the applicability of global resource serialization in its TSO environment before using it, the minor names (SYS1.UADS and SYS1.BRODCAST) of both data sets are included in the default SYSTEMS exclusion resource name list distributed by IBM (that is, the resources are **excluded** from global sharing). In the process of evaluation, plan in advance to investigate and measure:

- Resource requirements (the effort required to merge multiple versions of the two data sets into a single version of each and test the new versions)

- Performance implications (one version of each data set accessed by all users versus *n* versions of the same data sets, each accessed by a subset of those users).

Advantages:

1. Only two data sets to maintain; rather than 2*n* where *n* is the number of systems in a complex.

2. A user can logon from any system in a complex to allow a better workload balance.

3. For foreground-initiated background jobs, a user who specifies NOTIFY always receives the job-ended message regardless of which system in a complex processed the job.

Requirements:

1. Merge all existing versions of SYS1.UADS and SYS1.BRODCAST into a single version of each data set.

2. Modify the resource name lists, as distributed by IBM, as follows:

   a. Delete the minor names SYS1.UADS and SYS1.BRODCAST from the distributed default SYSTEMS exclusion resource name list.

   b. Add the major name SYSIKJUA as a generic entry in the SYSTEM inclusion resource name list (for SYS1.UADS sharing).

   c. Add the major name SYSIKJBC as a generic entry in the SYSTEM inclusion resource name list (for SYS1.BRODCAST sharing).

See *OS/VS2 Planning: MVS Global Resource Serialization* for information regarding the contents of, and how to modify, the resource name lists.

Establishing an Information Center Facility Environment

If your installation is planning to use the Information Center Facility, review and implement (as necessary) the procedures and processes discussed in this section.

Required and Distributed Libraries

The following libraries are used with the Information Center Facility. All the libraries are PDSs except APL workspace. The libraries that IBM distributes with TSO/E have a (D) in their description; the others are created by the installation after TSO/E is installed.

IBM-Supplied Data Set Name	DDNAME	Description	RECFM	LRECL	BLKSIZE
ICQ.ICQPLIB	ISPPLIB	Panel library (D)	FB	80	6400
ICQ.ICQMLIB	ISPMLIB	Message library (D)	FB	80	6400
ICQ.ICQSLIB	ISPSLIB	Skeleton library (D)	FB	80	6400
ICQ.ICQTLIB	ISPTLIB ICQTABL	General table library	FB	80	6400
ICQ.ICQAATAB	ICQAATAB	Names table library	FB	80	6400
ICQ.ICQABTAB	ICQABTAB	Administrator courses table library	FB	80	6400
ICQ.ICQANTAB	ICQANTAB	Administrator news table and text library	FB	80	6400
ICQ.ICQAPTAB	ICQAPTAB	Printer support table library	FB	80	6400
ICQ.ICQGCTAB	ICQGCTAB	IIPS/IIAS Registration table library	FB	80	6400
ICQ.ICQCCLIB	SYSPROC	User CLIST library (D)			
ICQ.ICQACLIB	SYSPROC	Administrator CLIST library (D)			
ICQ.ICQTABLS		ISPF defaults and user types table library (D)	FB	80	6400
ICQ.ICQABTXT		Abstracts for courses (D)	FB	80	6400
ICQ.ICQAPL		APL workspace (D)	FBS	80	6400

All of the libraries shown are required for the operation of the Information Center Facility, except for ICQ.ICQTABLS, ICQ.ICQABTXT, and ICQ.ICQAPL. These three libraries contain members that you must copy to other libraries before using the Information Center Facility. You must copy:

- ICQ.ICQAPL into the library @PL.@W000051.ICQUPDTS.
- The following members of the library ICQ.ICQTABLS:
  - ICQABT10 into ICQ.ICQABTAB.
  - ICQADT00, ICQAIT00, ICQAIT01, and ICQCMDS into ICQ.ICQTLIB.
  - ICQAPT00 into ICQ.ICQAPTAB.

- All members of ICQ.ICQABTXT into ICQ.ICQABTAB. If you have Release 2 installed, do not copy ICQ.ICQABTXT into ICQABTAB. When you installed Release 2, the members in ICQ.ICQABTXT were copied into ICQ.ICQAB. When you convert the Information Center Facility libraries and tables, library ICQ.ICQAB is automatically copied into ICQ.ICQABTAB.

IBM distributes the CLIST libraries, ICQ.ICQACLIB and ICQ.ICQCCLIB, with a RECFM of FB and an LRECL of 80. Only CLIST libraries with the same characteristics should be concatenated. If your production CLIST data set(s) has a RECFM of VB, run the CLIST ICQSMC00, a member of ICQ.ICQSAMP, against the CLIST libraries during installation to convert the libraries to a RECFM of VB.

Before using the Information Center Facility, the required libraries must be allocated, either in a LOGON procedure, using DD statements, or by using ALLOCATE commands. Figure 3-10 shows a sample LOGON procedure for administrators.

```
//ICQAPROC EXEC PGM=IKJEFT01,REGION=4096K,DYNAMNBR=40,
//          PARM='%ICQICFA'
//*
//*         CLIST DATA SETS
//*
//SYSPROC   DD   DSN=ICQ.ICQACLIB.CLIST,DISP=SHR
//          DD   DSN=ICQ.ICQCCLIB.CLIST,DISP=SHR
//          DD   DSN= *** SYSTEM CLIST LIBRARY ***
//          DD   DSN= *** PDF CLIST LIBRARY ***
//*
//*         HELP DATA SETS
//*
//SYSHELP   DD   DSN= *** SYSTEM HELP LIBRARY ***
//*
//*         SYSPRINT
//*
//SYSPRINT  DD   TERM=TS
//*
//*         SYSIN
//*
//SYSIN     DD   TERM=TS
//*
//*         ISPF DATA BASE DATA SETS
//*
//ISPPLIB   DD   DSN=ICQ.ICQPLIB,DISP=SHR
//          DD   DSN= *** PDF PANEL LIBRARY ***
//          DD   DSN= *** ISPF PANEL LIBRARY ***
//*
//ISPMLIB   DD   DSN=ICQ.ICQMLIB,DISP=SHR
//          DD   DSN= *** PDF MESSAGE LIBRARY ***
//          DD   DSN= *** ISPF MESSAGE LIBRARY ***
//*
//ISPSLIB   DD   DSN=ICQ.ICQSLIB,DISP=SHR
//          DD   DSN= *** PDF SKELETON LIBRARY ***
//          DD   DSN= *** ISPF SKELETON LIBRARY ***
//*
//ISPTLIB   DD   DSN=ICQ.ICQTLIB,DISP=SHR
//          DD   DSN= *** PDF TABLE LIBRARY ***
//          DD   DSN= *** ISPF TABLE LIBRARY ***
//*
//*         INFORMATION CENTER FACILITY DATA SETS
//*
//ICQTABL   DD   DSN=ICQ.ICQTLIB,DISP=SHR
//ICQAATAB  DD   DSN=ICQ.ICQAATAB,DISP=SHR
//ICQABTAB  DD   DSN=ICQ.ICQABTAB,DISP=SHR
//ICQANTAB  DD   DSN=ICQ.ICQANTAB,DISP=SHR
//ICQAPTAB  DD   DSN=ICQ.ICQAPTAB,DISP=SHR
//ICQGCTAB  DD   DSN=ICQ.ICQGCTAB,DISP=SHR
//*
//*         ISPF TEMPORARY DATA SETS
//*
//ISPCTL1   DD DISP=NEW,UNIT=SYSVIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=800,RECFM=FB,BUFNO=5)
//ISPCTL2   DD DISP=NEW,UNIT=SYSVIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=800,RECFM=FB,BUFNO=5)
//ISPLST1   DD DISP=NEW,UNIT=SYSVIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA,BUFNO=5)
//ISPLST2   DD DISP=NEW,UNIT=SYSVIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA,BUFNO=5)
```

Figure 3-10. Sample Administrator LOGON procedure

To adapt the sample LOGON procedure shown in Figure 3-10 (or the equivalent CLIST) for TSO/E users, make the changes indicated by the arrows in Figure 3-11.

```
//ICQPROC  EXEC PGM=IKJEFT01,REGION=4096K,              <====
//          DYNAMNBR=40,PARM='%ICQICF'                  <====
//*
//*         CLIST DATA SETS
//*
                 (line deleted)                              <====
//SYSPROC   DD   DSN=ICQ.ICQCCLIB.CLIST,DISP=SHR
//          DD   DSN= *** SYSTEM CLIST LIBRARY ***
//          DD   DSN= *** PDF CLIST LIBRARY ***
//*
.
.
.
//ISPLST2   DD DISP=NEW,UNIT=SYSVIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA,BUFNO=5)
```

Figure 3-11. Modifications for Sample User LOGON Procedure

Delete the line from the SYSPROC statement only if the user should not have access to the administrator CLIST library (ICQ.ICQACLIB).

Converting from Release 2 to Release 3

If your installation had TSO/E Release 2 installed, you must convert Information Center Facility libraries and tables for the installation of TSO/E Release 3. Perform the conversion in two steps:

1. Logon with a LOGON procedure (or execute an equivalent CLIST) that allocates the Release 2 libraries you must convert. For an example of the changes you can make to the sample LOGON procedure to allocate the libraries required for conversion, see Figure 3-12; the arrows in Figure 3-12 indicate the sample changes. The LOGON procedure or CLIST must allocate the Release 2 libraries for the administrator computer-based training tables, the administrator news tables, the IIPS/IIAS registration tables, and the user types tables.

2. Enter ISPF dialog test option 7.1, display panel ICQGAMR2, and select each of the options. See the online help for assistance with the conversion routines panel.

```
//ICQAPROC  EXEC PGM=IKJEFT01,REGION=4096K,
//          DYNAMNBR=40,PARM='%ICQCNVT'       <====
//*
 .
 .
 .
//ISPTLIB   DD   DSN=ICQ.ICQTLIB,DISP=SHR
//          DD   DSN= *** PDF TABLE LIBRARY ***
//          DD   DSN= *** ISPF TABLE LIBRARY ***
//*                                             <====
//*         LIBRARIES REQUIRED FOR CONVERSION    <====
//*                                             <====
//ICABTABL DD   DSN=ICQ.ICABTLIB,DISP=SHR    <====
//ICANTABL DD   DSN=ICQ.ICANTLIB,DISP=SHR    <====
//ICGCTABL DD   DSN=ICQ.ICGCTLIB,DISP=SHR    <====
//*
//*         INFORMATION CENTER FACILITY DATA SETS
//*
//ICQTABL   DD   DSN=ICQ.ICQTLIB,DISP=SHR
 .
 .
 .
//ISPLST2   DD DISP=NEW,UNIT=SYSVIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA,BUFNO=5)
```

Figure 3-12. Modifications for Sample Conversion LOGON Procedure

Program Libraries

Dialog functions coded as programs must be link edited. The load modules ICQGCL00, ICQCALN1, and ICQCAL00 are link edited into SYS1.CMDLIB when TSO/E is installed. If you wish, you can move the load modules into a step library, a system link library (such as SYS1.LINKLIB), or the link pack area. Alternately, you can concatenate load libraries to the ISPF link library (ddname = ISPLLIB, RECFM = U).

Installation-Developed Applications

If your installation has applications that you want to add to the Information Center Facility - ISPF library concatenation, consider the advantages and disadvantages of the following different concatenation sequences:

Concatenation Sequence	Advantages	Disadvantages
1. IBM-supplied applications 2. Installation-developed applications (system-wide, group, and user)	● SMP keeps track of all modifications to IBM libraries ● Only one copy of each IBM panel exists ● Control over the primary option menu is maintained.	● Must be familiar with SMP ● Modifying IBM-supplied libraries makes the next installation a little more time-consuming.
1. Installation-developed applications (system-wide) 2. IBM-supplied applications 3. Installation-developed applications (group and user)	● Need not be familiar with SMP ● No modifications to IBM-supplied libraries ● Control over the primary option menu is maintained.	● Difficult to implement more than one set of user models.
1. Installation-developed applications (system-wide and group) 2. IBM-supplied applications 3. Installation-developed applications (user)	● Need not be familiar with SMP. ● No modifications to IBM-supplied libraries.	● Control over the primary option menu is not maintained. ● System-wide modules can be overridden inadvertently.
1. Installation-developed applications (system-wide, group, and user) 2. IBM-supplied applications	● Need not be familiar with SMP. ● Total user freedom.	● Lack of consistency throughout the installation.

Doing an Information Center Facility SYSGEN

If you are doing a SYSGEN of MVS, and plan to use the Information Center Facility, you must use the SYSGEN macro SGIKJICQ in SYS1.AGENLIB to include the Information Center Facility in the SYSGEN. (See also the section "Including Session Manager CLISTs in a SYSGEN.")

After you load the distribution libraries, do one of the following:

- Code SGIKJICQ to build a job outside of the SYSGEN process
- Code SGIKJICQ and add it to the stage 1 system generation deck.

Assembling SGIKJICQ with its Parameters

Before assembling SGIKJICQ, consider the following:

- You must catalog the distribution library data sets.
- If you specify a value for one of the space parameters on the SGIKJICQ macro, the macro allocates the system target library with the space you specify. To use the default allocation values for a device type 3350 (D/T3350), specify DEFAULTS for the space parameter. If you do not specify one of the space parameters, or specify a space parameter with a null value (xxxxSPE=), the system assumes that the library is allocated and cataloged. Any other parameters you supply for that library are ignored.

  Notes:
  1. *You can use any device type that is valid for JCL on the unit parameters.*
  2. *Enter values for the space parameters (xxxxSPE) in the same format as for JCL. For example:*

     ```
     SAMPSPE=(TRK,(5,,1))
     ```

  The following example shows the default values of the parameters for SGIKJICQ. All parameters are optional.

```
SGIKJICQ JCLASS=A,
     OCLASS=A,
     WRKUNT=SYSDA,
     LIBPRE=ICQ,
     CATLG=NO,
     CMDUNT=3350,
     CMDVOL=SYSRES,
     CMDPRE=SYS1,
     PLIBUNT=3350,
     PLIBVOL=SYSRES,
     PLIBSPE=,
     PLIBBLK=3120,
     PLIBRFM=FB,
     PLIBLRC=80,
     SLIBUNT=3350,
     SLIBVOL=SYSRES,
     SLIBSPE=,
     SLIBBLK=3120,
     SLIBRFM=FB,
     SLIBLRC=80,
     MLIBUNT=3350,
     MLIBVOL=SYSRES,
     MLIBSPE=,
     MLIBBLK=3120,
     MLIBRFM=FB,
     MLIBLRC=80,
     TABLUNT=3350,
     TABLVOL=SYSRES,
     TABLSPE=,
     TABLBLK=3120,
```

```
TABLRFM=FB,
TABLLRC=80,
ACLIUNT=3350,
ACLIVOL=SYSRES,
ACLISPE=,
ACLIBLK=3120,
ACLIRFM=FB,
ACLILRC=80,
CCLIUNT=3350,
CCLIVOL=SYSRES,
CCLISPE=,
CCLIBLK=3120,
CCLIRFM=FB,
CCLILRC=80,
APLUNT=3350,
APLVOL=SYSRES,
APLSPE=,
APLBLK=3120,
APLRFM=FB,
APLLRC=80,
ABXTUNT=3350,
ABXTVOL=SYSRES,
ABXTSPE=,
ABXTBLK=3120,
ABXTRFM=FB,
ABXTLRC=80,
SAMPUNT=3350,
SAMPVOL=SYSRES,
SAMPSPE=,
SAMPBLK=3120,
SAMPRFM=FB,
SAMPLRC=80
```

JCLASS
specifies job class.

OCLASS
specifies output class.

WRKUNT
specifies the work data space for IEBCOPY.

LIBPRE
specifies the prefix for the Information Center Facility target data sets (ICQPLIB, ICQSLIB, ICQMLIB, ICQTABL, ICQACLIB, ICQCCLIB, ICQAPL, ICQABTXT, and ICQSAMP).

CATLG
specifies that the system should not catalog or allocate the data sets using DISP=(,CATLG). Use this parameter when the data sets are already allocated.

CMDUNT
specifies the unit type of the device on which the CMDLIB data set resides.

CMDVOL
specifies the volume serial number of the device on which the CMDLIB data set resides.

CMDPRE
specifies the prefix for the CMDLIB data set.

PLIBUNT
specifies the unit type of the device on which you are allocating space for the ICQPLIB data set or on which the ICQPLIB data set resides.

PLIBVOL
specifies the volume serial number of the device on which the ICQPLIB data set resides.

PLIBSPE
specifies the space to allocate for the ICQPLIB data set. If you do not specify a value, the system assumes that the ICQPLIB data set is already allocated and cataloged. If you want the ICQPLIB data set to be allocated using SPACE=(TRK,(270,,80)), specify DEFAULTS.

PLIBBLK
specifies the block size of the ICQPLIB data set. Use this parameter if you are allocating the ICQPLIB data set.

PLIBRFM
specifies the record format of the ICQPLIB data set. Use this parameter if you are allocating the ICQPLIB data set.

PLIBLRC
specifies the logical record length of the ICQPLIB data set. Use this parameter if you are allocating the ICQPLIB data set.

SLIBUNT
specifies the unit type of the device on which you are allocating space for the ICQSLIB data set or on which the ICQSLIB data set resides.

SLIBVOL
specifies the volume serial number of the device on which the ICQSLIB data set resides.

SLIBSPE
specifies the space to allocate for the ICQSLIB data set. If you do not specify a value, the system assumes that the ICQSLIB data set is already allocated and cataloged. If you want the ICQSLIB data set to be allocated using SPACE=(TRK,(7,,1)), specify DEFAULTS.

SLIBBLK
specifies the block size of the ICQSLIB data set. Use this parameter if you are allocating the ICQSLIB data set.

SLIBRFM
specifies the record format of the ICQSLIB data set. Use this parameter if you are allocating the ICQSLIB data set.

SLIBLRC
specifies the logical record length of the ICQSLIB data set. Use this parameter if you are allocating the ICQSLIB data set.

MLIBUNT
specifies the unit type of the device on which you are allocating space for the ICQMLIB data set or on which the ICQMLIB data set resides.

MLIBVOL
specifies the volume serial number of the device on which the ICQMLIB data set resides.

MLIBSPE
specifies the space to allocate for the ICQMLIB data set. If you do not specify a value, the system assumes that the ICQMLIB data set is already allocated and cataloged. If you want the ICQMLIB data set to be allocated using SPACE=(TRK,(30,,5)), specify DEFAULTS.

MLIBBLK
specifies the block size of the ICQMLIB data set. Use this parameter if you are allocating the ICQMLIB data set.

MLIBRFM
specifies the record format of the ICQMLIB data set. Use this parameter if you are allocating the ICQMLIB data set.

MLIBLRC
specifies the logical record length of the ICQMLIB data set. Use this parameter if you are allocating the ICQMLIB data set.

TABLUNT
specifies the unit type of the device on which you are allocating space for the ICQTABL data set or on which the ICQTABL data set resides.

TABLVOL
specifies the volume serial number of the device on which the ICQTABL data set resides.

TABLSPE
specifies the space to allocate for the ICQTABL data set. If you do not specify a value, the system assumes that the ICQTABL data set is already allocated and cataloged. If you want the ICQTABL data set to be allocated using SPACE=(TRK,(10,,1)), specify DEFAULTS.

TABLBLK
specifies the block size of the ICQTABL data set. Use this parameter if you are allocating the ICQTABL data set.

TABLRFM
specifies the record format of the ICQTABL data set. Use this parameter if you are allocating the ICQTABL data set.

TABLLRC
specifies the logical record length of the ICQTABL data set. Use this parameter if you are allocating the ICQTABL data set.

ACLIUNT
specifies the unit type of the device on which you are allocating space for the ICQACLIB data set or on which the ICQACLIB data set resides.

ACLIVOL
specifies the volume serial number of the device on which the ICQACLIB data set resides.

ACLISPE
specifies the space to allocate for the ICQACLIB data set. If you do not specify a value, the system assumes that the ICQACLIB data set is already allocated and cataloged. If you want the ICQACLIB data set to be allocated using SPACE=(TRK,(80,,4)), specify DEFAULTS.

ACLIBLK
specifies the block size of the ICQACLIB data set. Use this parameter if you are allocating the ICQACLIB data set.

ACLIRFM
specifies the record format of the ICQACLIB data set. Use this parameter if you are allocating the ICQACLIB data set.

ACLILRC
specifies the logical record length of the ICQACLIB data set. Use this parameter if you are allocating the ICQACLIB data set.

CCLIUNT
specifies the unit type of the device on which you are allocating space for the ICQCCLIB data set or on which the ICQCCLIB data set resides.

CCLIVOL
specifies the volume serial number of the device on which the ICQCCLIB data set resides.

CCLISPE
specifies the space to allocate for the ICQCCLIB data set. If you do not specify a value, the system assumes that the ICQCCLIB data set is already allocated and cataloged. If you want the ICQCCLIB data set to be allocated using SPACE=(TRK,(60,,4)), specify DEFAULTS.

CCLIBLK
specifies the block size of the ICQCCLIB data set. Use this parameter if you are allocating the ICQCCLIB data set.

CCLIRFM
specifies the record format of the ICQCCLIB data set. Use this parameter if you are allocating the ICQCCLIB data set.

CCLILRC
specifies the logical record length of the ICQCCLIB data set. Use this parameter if you are allocating the ICQCCLIB data set.

APLUNT
specifies the unit type of the device on which you are allocating space for the ICQAPL data set or on which the ICQAPL data set resides.

APLVOL
specifies the volume serial number of the device on which the ICQAPL data set resides.

APLSPE
specifies the space to allocate for the ICQAPL data set. If you do not specify a value, the system assumes that the ICQAPL data set is already allocated and cataloged. If you want the ICQAPL data set to be allocated using SPACE=(TRK,(7,,1)), specify DEFAULTS.

APLBLK
specifies the block size of the ICQAPL data set. Use this parameter if you are allocating the ICQAPL data set.

APLRFM
specifies the record format of the ICQAPL data set. Use this parameter if you are allocating the ICQAPL data set.

APLLRC
specifies the logical record length of the ICQAPL data set. Use this parameter if you are allocating the ICQAPL data set.

ABTXUNT
specifies the unit type of the device on which you are allocating space for the ICQABTXT data set or on which the ICQABTXT data set resides.

ABTXVOL
specifies the volume serial number of the device on which the ICQABTXT data set resides.

ABTXSPE
specifies the space to allocate for the ICQABTXT data set. If you do not specify a value, the system assumes that the ICQABTXT data set is already allocated and cataloged. If you want the ICQABTXT data set to be allocated using SPACE=(TRK,(5,,3)), specify DEFAULTS.

ABTXBLK
specifies the block size of the ICQABTXT data set. Use this parameter if you are allocating the ICQABTXT data set.

ABTXRFM
specifies the record format of the ICQABTXT data set. Use this parameter if you are allocating the ICQABTXT data set.

ABTXLRC
specifies the logical record length of the ICQABTXT data set. Use this parameter if you are allocating the ICQABTXT data set.

SAMPUNT
specifies the unit type of the device on which you are allocating space for the ICQSAMP data set or on which the ICQSAMP data set resides.

SAMPVOL
specifies the volume serial number of the device on which the ICQSAMP data set resides.

SAMPSPE
specifies the space to allocate for the ICQSAMP data set. If you do not specify a value, the system assumes that the ICQSAMP data set is already allocated and cataloged. If you want the ICQSAMP data set to be allocated using SPACE=(TRK,(5,,1)), specify DEFAULTS.

SAMPBLK
specifies the block size of the ICQSAMP data set. Use this parameter if you are allocating the ICQSAMP data set.

SAMPRFM
specifies the record format of the ICQSAMP data set. Use this parameter if you are allocating the ICQSAMP data set.

SAMPLRC
specifies the logical record length of the ICQSAMP data set. Use this parameter if you are allocating the ICQSAMP data set.

Coding SGIKJICQ in a Job Outside the SYSGEN Process

To code SGIKJICQ in a job outside of the SYSGEN process, do the following:

1. Assemble SGIKJICQ.
2. Submit the output from the assembly as a job.

The following example assumes that, in the SMP4 environment, the SMPACDS has been copied to the SMPCDS and the SMPACRQ has been copied to the SMPCRQ after the Information Center Facility was accepted. (If the SMPACDS has not been copied to the SMPCDS, the SMPCDS does not indicate that the Information Center Facility is installed.) The job in the example causes SMP4 to build entries in the SMPCDS showing the structure of the target system library modules.

```
//JOB1      JOB   'ACCOUNT #','NAME',MSGLEVEL=(1,1)
//JCLIN     EXEC  SMPPROC
//SMPJCLIN DD     output from assembly of macro SGIKJICQ
//SMPCNTL   DD    *
  JCLIN.
/*
```

The program directory provides a similar example for an SMP/E environment.

Coding SGIKJICQ in a Stage 1 System Generation

The following example shows where to place SGIKJICQ in a stage 1 system generation input definition:

```
STAGE macro definitions
  .
  .
  .
GENERATE parameters
SGIKJICQ parameters
END
```

Printer Support Considerations

You can use TSO/E printer support to standardize the connections between applications and printers at your installation. An application can invoke a printer selection CLIST, allowing a user to choose from a list of printer definitions. The printer definitions are sets of characteristics that identify physical printers and define the way data sets are to be printed on them. The application can use the characteristics provided by the printer definition that the user chooses, or it can override some or all of them. An application can also print a data set directly, without permitting the user to choose a printer definition.

Defining Printers

Before applications can use printer support, your installation must set up printer definitions. The Information Center Facility includes panels for defining printers and their characteristics. One physical printer can have more than one printer definition; however, the combination of printer location and print format must be unique.

For more information about creating or modifying printer definitions, see *TSO/E Information Center Facility Administrator's Guide* and Volume 2.

Using Printer Support

Your installation's print applications can use CLIST ICQCPC00, the printer selection CLIST, to display to users part or all of the list of available printer definitions. The application can also permit the user to select or change the order of the fonts to be used for printing.

If your application specifies PRINT when it invokes ICQCPC00, ICQCPC00 calls the print function given by the printer definition that the user selects. The print function can be the IBM-supplied print CLIST ICQCPC10, or an installation-supplied print CLIST, program, or command. If your application calls the print function directly, the application can override values from the print definition that the user selects by specifying different values on the invocation of the print function.

As shipped, printer support is limited to printers that ICQCPC10 can access directly through TSO/E. ICQCPC10 supports only parameters of the TSO/E

ALLOCATE command, and ignores other parameters. To use the 6670 pre-processor field-developed program (FDP) or any other printer interface, your installation must supply a print CLIST for the interface. In addition, the administrator must identify such a CLIST when providing information about printers that cannot be accessed through TSO/E directly.

Users of printer support must be made aware that they receive an error message if they enter an asterisk (*) within the location field on panel ICQAPE30. An asterisk has a special meaning in the Information Center Facility. You can never include an asterisk within a data entry field; if you include one within a data entry field, you receive an error message. However, you can, in special cases, enter an asterisk at the end of a data entry field, and then use the field to request a partial list. For example, if you enter SP* in the location field of panel ICQAPE30, you receive a partial list of locations (those beginning with SP). The Information Center Facility panels on which you can use the asterisk indicate the fields in which you can use it.

For more information about the printer selection CLIST, ICQCPC00, or the print CLIST, ICQCPC10, see *TSO/E User's Guide.*

Space Management Considerations

When a user accesses a data set, space management checks to see if the data set exists. If it does not exist, space management allocates it. If it exists and is nearly full, space management increases the free space in the data set before the data set reaches its space limit, either by compressing or reallocating the data set.

If the user accesses an existing sequential data set that is nearly full, space management allocates a new data set, and copies any RACF protection the old data set had to the new data set. Space management then copies the data from the old data set to the new one, deletes the old data set, and renames the new data set to the old data set name.

If the user accesses an existing partitioned data set that is nearly full, space management first compresses it. If compressing the data set does not increase the free space sufficiently, space management reallocates the data set as described for a sequential data set.

Information Center Facility functions that add data to files, such as names processing, invoke space management automatically. Installation-written error routines or edit macros can also invoke space management.

You should be aware of the following when using space management:

- Space management updates RACF accounting data when it reallocates a data set.

- If you do not have RACF 1.7 installed, space management cannot copy the RACF universal access (UACC) from the old data set profile to the new data set profile. Instead, space management gives the new data set the default UACC, which might not match that of the old data set. For any release of RACF, however, space management copies the RACF access list from the old profile to the new profile.

- If the old data set is password-protected, and password-protected data sets are allowed (whether password-protected data sets are allowed is specified when space management is invoked), the password is lost if space management must reallocate the data set.

- When space management enlarges a data set that has a high-level qualifier that is not the user's userid, the user must have the authority to create a RACF profile for the temporary data set used during reallocation.

IIPS

The Information Center Facility uses a default high-level qualifier of IIPS for IIPS data sets. If your installation specifies a different qualifier when installing IIPS, change the Information Center Facility default. To change the default:

1. Select the COURSES option on the primary selection panel for Information Center Facility administrators.

2. Select the DEFAULTS option on the next panel you see (the INFORMATION CENTER COURSES panel).

3. When the COURSES-MODIFY ADMINISTRATION DEFAULTS panel is displayed, enter the high-level qualifier your installation uses.

The Information Center Facility uses the IISBATCH program to process registration requests. The BCONFIG member of the IIPS.OS.CTLCARD data set contains control statements for IISBATCH. (Your installation might use a qualifier other than IIPS in the data set name.) Before you can register students in a course using the Information Center Facility, BCONFIG must contain the statement DISKnn=YES, where nn corresponds to the number of the data set in which the course resides. The DISKnn statement identifies the course data set to IISBATCH. If the statement is omitted, registration fails. Therefore, your installation needs to ensure that BCONFIG contains a DISKnn=YES statement for each data set that might contain a course in which students can request registration using the Information Center Facility.

Date Format

Variables &QCCDFMTX and &QCCDFMT in the non-display panel ICQSIE00 contain a date format used in date processing for a number of functions. Set &QCCDFMT to the correct date format, using any characters required by the national language. &QCCDFMTX should be set to the equivalent date format using English characters ("mm" for month, "dd" for day, and "yy" for year). Check &QCCDFMTX to make certain it is set correctly. An error message is issued if the first two characters of the date format in &QCCDFMTX are not set to English characters.

APLDI-II and ADRS-II

The CLIST ICQGCC20 contains variables to provide the Information Center Facility interface to APLDI and ADRS. Two of the variables, TESTDATE and TESTTIME, must be set by the installation. The requirement to modify any of the other variables depends on how APLDI and ADRS were installed and how an installation wants the products to function.

Additional Product Support

The Information Center Facility supports the following IBM licensed programs with selection panels, HELP panels, tutorials, and dialogs:

APL and APL-based associated products:

- APL2 (5668-899): A Programming Language 2
- VS/APL (5748-AP1): Virtual Storage/A Programming Language
- APLDI-II (5796-PNJ): APL Data Interface II
- ADRS-II (5796-PLN): A Departmental Reporting System II with Business Graphics (ADRS-II/BG)
- Info Center/1 (5668-897): Information Center/1
- FPS (5798-CXP): Financial Planning System - TSO

  When using the Information Center Facility, you can access the FPS routines only through ADRS-II or Info Center/1. Note that IBM no longer provides service for FPS. Consult the following publications for information about installing the FPS routines or ADRS with the BG feature and the FPS routines under ADRS-II:

  - A Departmental Reporting System Users Guide
  - A Departmental Reporting System Systems Guide
  - A Departmental Reporting System II Business Guide
  - The Financial Planning System - TSO Systems Guide

Other associated products:

AS (5767-001): Application System

GDDM/PGF: Graphical Data Display Manager (5748-XXH)/Presentation Graphics Feature (5748-XXH-02)

IBM BASIC/MVS (5665-948)

IIPS/IIAS: Interactive Instructional Presentation System (5668-012)/Interactive Instructional Authoring System (5668-011)

PS/TSO (5665-346): Personal Services/TSO

QMF (5668-972): Query Management Facility

TIF (5665-339): The Information Facility

Modifications for Installation

If you install one of the following products for use with the Information Center Facility, you must reset the associated variables as described below:

APL2: Set variable &QCLAPL2 in panel ICQGCM05 to "Y."

VS/APL: Set variable &QCLVSAPL in panel ICQGCM05 to "Y."

AS: Set variable &QCLMVSAS in panel ICQGCM05 to "Y."

If &QCLMVSAS = Y, the Information Center Facility tutorial includes the Application System help and tutorial panels. If &QCLMVSAS = N, only the Information Center Facility panels describing Application System are displayed.

Info Center/1: Set variable &QCLIC1 in panel ICQGCM01 to "Y."

GDDM/PGF: Set the following variables in the non-display panel ICQSIECG to the appropriate library names:

Set:	*To:*
&QCGICU	Library containing the ICU load module
&QCGISE	Library containing the ISE load module
&QCGVSE	Library containing the VSE load module
&QCGSYMBL	Library containing the default symbols

PS/TSO: Set variable &QCLPSTSO in panel ICQGCM03 to "Y."

QMF: Set variable &QCLQMF in panel ICQGCM01 to "Y."

TIF: Set variable &QCLTIF in panels ICQGCM01 and ICQGCM05 to "Y."

IBM BASIC/MVS: Set variable &QCLBASIC in panel ICQGCM05 to "Y."

Names Service Considerations

The Information Center Facility names service maintains a system names directory and users' private names directories. These names directories help users locate other users, and assist them in transmitting messages through the system.

The names service keeps directory information in ISPF tables. There are two types of table entries: single user entries, and group entries that apply to more than one user. The size and type of an entry can vary, along with the total number of directory entries. As a result, the total size of the personal and master directories can vary. The maximum size for a single user entry is 715 bytes, with the average size being 350-400 bytes. Group entries consist of 300 bytes per group, plus an additional 15 bytes per group member. A copy of the user's personal directory table and a copy of the concatenated personal and master directory tables reside in private area virtual storage. Each copy requires 1K bytes of additional virtual storage.

When users update the master directory, the names service requires an additional table to hold the updates. Each update requires a maximum of 800 bytes, with the average size being 400-450 bytes. A single table holds the updates for all users and empties as the administrator processes the updates. The total size of the additional table depends on the rate at which users update the master directory and the rate at which the administrator processes those updates. A copy of the table resides in private area virtual storage, requiring an additional 1K bytes of virtual storage.

With ISPF Version 2, Release 2, the names table and all ISPF tables are loaded above 16 Mb in virtual storage.

By default, the names directories are open and kept in virtual storage only until users are finished accessing them. You can specify that the names directories remain open and in storage until they are either explicitly closed, or until users exit their logical ISPF/PDF sessions.

Specifying that the directories remain open:

- Reduces the I/O overhead of opening and closing the directories each time they are accessed.
- Simplifies application processing of the directories: when applications terminate they do not have to explicitly close a directory.

However, you should be aware that leaving the directories open will:

- Require additional user storage. (This storage is above 16 megabytes in virtual storage if your installation is using ISPF Release 2.2 or later.)
- Prevent users from always obtaining the directory they want:
  - Users may not obtain the latest changes the administrator makes to the master directory. To obtain the latest changes, users must close and open the directory.
  - When users request just the private directory, they will receive both the private and the merged directory.
  - Rarely, when users request the master directory, they will receive the merged directory, and vice versa.

Note: If your system is storage-constrained, you should not specify that the names directories will remain open. Leaving the directories open could degrade performance.

To specify whether the names directories will remain open, use the QCASTAT variable of the ICQSIECA panel. For information on how to customize information center facility variables, see *SPL: TSO Extensions User Exits and Modifications Volume 2.*

Establishing a Session Manager Environment

Before the session manager can be used in an installation, the installation may have to:

- Modify a SYS1.PARMLIB member, either TSOKEY00 or IKJPRM00
- Modify or create logon procedures tailored for the session manager
- Modify the TSO message handler (MH) and the message control program (MCP) if TSO is running under control of TCAM.

The required modifications are presented in this section, together with other considerations for activating the session manager.

SYS1.PARMLIB Changes

To support a session manager environment, an installation may have to modify SYS1.PARMLIB member TSOKEY00 (or an alternate member) for VTAM or member IKJPRM00 (or an alternate member) for TCAM.

VTAM

Two defaults in member TSOKEY00 are:

- SCRSIZE - 480
- BUFRSIZE - 132.

Recommendations:

1. Increase SCRSIZE to correspond to the type of terminals installed. (For example, specify 3440 for a 3278 Model 4.)
2. Increase BUFRSIZE to 1000 or greater. (Although the buffer size is not critical, a larger size should improve performance.)

TCAM

Defaults in member IKJPRM00 are:

- BUFSIZE - 64
- BUFFERS - 6*USERMAX
- INLICKHI - 4
- INLOCKLO - 1
- OWAITHI - 20
- OWAITLO - 4
- RESVBUF - .1 (6*USERMAX).

Recommendations

1. Set BUFSIZE = 2 (UNITSZ) but not greater than 252
2. Set OWAITHI = $\underline{\text{1.2M}\quad}$ BUFSIZE
3. Set Buffers = 1.2*t*OWAITHI
4. Set RESVBUF = $\frac{\text{BUFFERS}}{7}$
5. Set OWAITLO = $\frac{\text{OWAITHI}}{4}$
6. Set INLOCKHI = OWAITHI
7. Set INLOCKLO = OWAITLO

In the preceding algorithms:

- UNITSZ is an operand on the INTRO macro (Refer to *ACF/TCAM Version 2 Installation Reference*)
- *M* is the maximum value specified on any BUFSIZE parameter in a DCB or TERMINAL macro used for TSO
- *t* is the total number of TERMINAL macros used for TSO
- When a computation results in a remainder, round up.

  Note: Ensure that the value of OWAITHI is large enough to allow a full-screen write before swap out. The value of OWAITHI times the value of BUFSIZE should not be less than SCRSIZE to prevent performance degradation.

Default Environment

When being initialized, the session manager loads the default environment module specified in the PARM field on the EXEC statement of the logon procedure. IBM supplies a default module, ADFMDFLT, that is loaded if a module name is not specified in the PARM field. The default module contains the tables and data necessary for building a TSO user's default screen layout, PFK definitions, etc.

If an installation wants to use a single default environment for all its TSO users, it can modify the IBM-supplied default module to set up the installation-dependent environment. If a single environment is not satisfactory for all its TSO users, an installation can create multiple logon procedures, each one specifying a different default module.

LOGON Procedure Changes

To enable the use of the session manager, an installation may have to create new LOGON procedures or modify the EXEC statement in existing procedures. The EXEC statement can have the following format:

```
//[stepname]  EXEC  PGM=ADFMDF03,

[// PARM= 'SM ( tmpname   , Y , default-module-name )[,] tso-cmd
                IKJEFT01    N   ADFMDFLT
```

Explanation:

- PGM=ADFMDF03 - attach the session manager initialization task, ADFMDF03, instead of the TMP. After initialization is complete, attach the TMP.

- *tmpname* - the name of an installation-written TMP.

- IKJEFT01 - the name of the IBM-supplied TMP.

- Y - the TMP is attached APF authorized.

- N - the TMP is attached non-APF authorized.

- *default-module-name* - the name of an installation-written default environment module.

- ADFMDFLT - the name of the IBM-supplied default environment module.

- *tso-cmd* - a TSO command with any associated operands and parameters.

A sample logon procedure for session manager is:

```
//SMPROC   EXEC   PGM=ADFMDF03,DYNAMNBR=30,
//  PARM='SM(IKJEFT01,Y)  EXEC  "SYS1.PRD.CLIST(SMLOGON)"'
//SYSPROC   DD    DSN=SYS1.TSO.CLIST,DISP=SHR
     .
     .
     .
/*
```

Note: The data set SYS1.PRD.CLIST is an installation-defined CLIST data set.

Message Handler (MH) and Message Control Program (MCP) Changes

An installation may have to make modifications to the MH and MCP to support a session manager environment. The modifications are unnecessary if:

- EXPFLS=YES was specified on the FULLSCR macro for some other full-screen application, for example, ISPF or IPCS

- Session manager runs under control of TSO/VTAM.

Creation of the MH normally requires two assemblies (stage 1 and stage 2), where the output from the first assembly is the input to the second. (Refer to *ACF/VTAM Version 2 Base Installation Guide.*)

Stage 1 Modification

If the full screen (FULLSCR=YES) operand was not originally specified during MCP generation, make the following change and reassemble stage 1. (Alternatively, step 2 under stage 2 modifications may be used.)

- Add the FULLSCR=YES operand to the LINEGRP and/or LISTTA macros that describe the terminals being used and reassemble the MCP generation deck. The output is an MCP source deck with:
  - The generation of the OPTION macro for IEDQFSCR
  - The allocation and initialization to zero of IEDQFSCR in the appropriate TERMINAL macros using the OPDATA operand
  - The generation of the FULLSCR macro in the message handler, immediately preceding the SIMATTN macro.

Stage 2 Modifications

1. If the FULLSCR=YES operand was originally specified in stage 1, make the following two modifications to the stage 1 output:
   - Add the EXPFLS=YES operand to the FULLSCR macro in the MH
   - Insert the changed FULLSCR macro in the output side of the MH between the macros OUTBUF and CODE.
2. If the FULLSCR=YES operand was not originally specified during stage 1, make the following modifications to the stage 1 output:
   - Insert the macro FULLSCR EXPFLS=YES in the MH immediately preceding the SIMATTN macro
   - Insert the macro FULLSCR EXPFLS=YES in the output side of the MH between the macros OUTBUF and CODE
   - Insert the following OPTION statement at an appropriate place in the option table:

     ```
     IEDQFSCR OPTION XL1
     ```

   - Ensure that every TERMINAL macro representing terminals to be used with session manager has an OPDATA operand with a subfield of zero corresponding to the IEDQFSCR option

     Note: The subfields of the OPDATA operand are 'positional' in that their order corresponds to the order of the OPTION macros that make up the option table.

   - Ensure that every other TERMINAL macro with an OPDATA operand also has a null subfield corresponding to the IEDQFSCR option

Stage 2 Assembly

Before proceeding with the stage 2 assembly, examine the current operand values for BUFSIZE. CUTOFF, and LNUNITS.

- Ensure that the value of the BUFSIZE operand is at least 2100 on both the TERMINAL and DCB macros for all terminals to be used with session manager. BUFSIZE must be an even multiple of the UNITSZ operand value on the INTRO macro.
- Ensure that the operand value of the CUTOFF macro is at least 2048
- Ensure that the value of the LNUNITS operand of the INTRO macro is large enough. The value should at least equal (for all TCAM DCBs):

  $$\frac{B * T}{U}$$

  where:

  B
  is the BUFSIZE value for a DCB macro

  T
  is the number of TERMINAL macros for that DCB

  U
  is the UNITSZ value on the INTRO macro

After making the modifications, reassemble stage 2.

System Interlock Condition

In a session manager environment with the session manager active (not running in a full-screen environment, for example, ISPF not active) a system interlock occurs if a task running above the session manager task sets the session manager task nondispatchable and issues TGETs or TPUTs.

For example, the SMF time limit installation exit, IEFUTL, should not issue TGETs or TPUTs because the session manager task is set nondispatchable prior to the exit getting control.

CLISTs

TSO/E installs session manager CLISTs ADFSPLT, ADFSETUP, and ADFVSPLT into SYS1.ADFMAC1.

Recommendation

Copy the CLISTs into the installation-defined production CLIST data set. If your production CLIST has a RECFM of VB, run CLIST ICQSMC00. ICQSMC00, located in ICQ.ICQSAMP, converts the CLIST data set members shipped to you from a RECFM of FB to a RECFM of VB.

SYS1.SMLIB

SYS1.SMLIB is no longer required. The session manager is link edited into SYS1.LINKLIB.

Recommendation

Delete SYS1.SMLIB and remove all linklist library and similar references to it.

Including Session Manager CLISTs in a SYSGEN

If you want to SYSGEN MVS, you must use the SYSGEN macro SGIKJSM3 in SYS1.AGENLIB to include the session manager CLISTs in the SYSGEN. (See also the section "Doing an Information Center Facility SYSGEN.")

After you load the distribution libraries, do one of the following:

- Code SGIKJSM3 to build a job outside of the SYSGEN process.
- Code SGIKJSM3 and add it to the stage 1 system generation deck.

Assembling SGIKJSM3 with its Parameters

Before assembling SGIKJSM3, consider the following:

- You must catalog the distribution library data sets.
- If you specify a value for MAC1SPE on the SGIKJSM3 macro, the macro allocates ADFMAC1 with the space you specify. To use the default allocation values for a device type 3350 (D/T3350), specify DEFAULTS for MAC1SPE. If you do not specify MAC1SPE, or if you specify a null value (MAC1SPE=), the system assumes that ADFMAC1 is allocated and cataloged. Any other parameters you supply are ignored.

  The following example shows the default values of the parameters for SGIKJSM3. All parameters are optional.

  ```
  SGIKJSM3 JCLASS=A,
      OCLASS=A,
      WRKUNT=SYSDA,
      SYSPRE=SYS1,
      CATLG=NO,
      MAC1UNT=3350,
      MAC1VOL=SYSRES,
      MAC1SPE=,
      MAC1BLK=3120,
      MAC1RFM=FB,
      MAC1LRC=80
  ```

JCLASS
specifies job class.

OCLASS
specifies output class.

WRKUNT
specifies the work data space for IEBCOPY.

SYSPRE
specifies the prefix for the ADFMAC1 data set.

CATLG
specifies that the system should not catalog or allocate the data set using DISP=(,CATLG). Use this parameter when the data set is already allocated.

MAC1UNT
specifies the unit type of the device on which you are allocating space for the ADFMAC1 data set or on which the ADFMAC1 data set resides. You can specify any device type that is valid to code on JCL.

MAC1VOL
specifies the volume serial number of the device on which the ADFMAC1 data set resides.

MAC1SPE
specifies the space to allocate for the ADFMAC1 data set. If you do not specify a value, the system assumes that the ADFMAC1 data set is already allocated and cataloged. If you want the ADFMAC1 data set to be allocated using SPACE=(TRK,(5,,2)), specify DEFAULTS.

Enter a value for MAC1SPE in the same format as for JCL. For example:

```
MAC1SPE=(TRK,(5,,1))
```

MAC1BLK
specifies the block size of the ADFMAC1 data set. Use this parameter if you are allocating the ADFMAC1 data set.

MAC1RFM
specifies the record format of the ADFMAC1 data set. Use this parameter if you are allocating the ADFMAC1 data set.

MAC1LRC
specifies the logical record length of the ADFMAC1 data set. Use this parameter if you are allocating the ADFMAC1 data set.

Coding SGIKJSM3 in a Job Outside the SYSGEN Process

To code SGIKJSM3 in a job outside of the SYSGEN process, do the following:

1. Assemble SGIKJSM3.
2. Submit the output from the assembly as a job.

The following example assumes that, in the SMP4 environment, the SMPACDS has been copied to the SMPCDS and the SMPACRQ has been copied to the SMPCRQ after the session manager CLISTs were accepted. If the SMPACDS has not been copied to the SMPCDS, the SMPCDS does not indicate that the session manager CLISTs are installed. The job in the example causes SMP4 to build entries in the SMPCDS showing the structure of the target system library modules.

```
//JOB1      JOB   'ACCOUNT #','NAME',MSGLEVEL=(1,1)
//JCLIN     EXEC  SMPPROC
//SMPJCLIN  DD    output from assembly of macro SGIKJSM3
//SMPCNTL   DD    *
  JCLIN.
/*
```

The program directory provides a similar example for an SMP/E environment.

Coding SGIKJSM3 in a Stage 1 System Generation

The following example shows where SGIKJSM3 should be placed in a stage 1 system generation input definition:

```
STAGE macro definitions
.
.
.
GENERATE parameters
SGIKJSM3 parameters
END
```

Allocating Space for the Utility Work Data Sets that EDIT Uses

If a user is editing a new or an existing data set using the MOVE or COPY subcommand, EDIT uses:

1. temporary utility work data sets when the user's profile indicates NORECOVER, when the user is executing commands in the background, or when the user specifies NORECOVER.

2. permanent utility work data sets when the user is authorized to use the EDIT recovery facility and specifies RECOVER on the EDIT command.

For both temporary and permanent utility work data sets, an installation has two options for space allocation:

1. default space allocation
2. controlled space allocation

Default Space Allocation

The following paragraphs explain the algorithms used by EDIT to calculate the default space allocation for utility work data sets.

Editing a New Data Set: A utility work data set is allocated four blocks with a default block size of 4096 bytes for primary space (16384 bytes) and one half that amount (8192 bytes) for secondary space. (If the data set being edited contains 80-character records, the utility data set has a total capacity of approximately 300 records.)

Editing an Existing Data Set: A utility work data set is allocated:

```
2 ((((x + y)b)/4096)+2) = primary space in 4K blocks
```

where:

x = number of records in existing data set

y = additional number of records to be added to the data set (This number is variable and depends on the user specifications on a MOVE or COPY subcommand. The value may be 0.)

b = number of bytes (characters) per record

```
Secondary space = primary-space/2
```

Example:

- an existing data set contains 6000 records
- 200 additional records are to be copied into the data set
- each record contains 120 characters

```
2 ((((6000 + 200) 120)/4096) +2) = 366 (primary space:
                                         number of 4K blocks)

    Secondary space =183 (number of 4K blocks)
```

The utility data set has a capacity of approximately 18,600 records.

Controlled Space Allocation

If an installation wants to control the allocation of the utility work data sets and the direct access space used for those data sets, it can preallocate the data sets.

Note: An installation may also need to preallocate data sets when using VIO due to the compounding effect of TSO and VIO allocation algorithms.

Temporary Utility Work Data Sets — Include in the user's LOGON procedure two DD statements that define the data sets.

```
//SYSEDIT    DD    DSN=&EDIT,UNIT=SYSDA,SPACE=(2048,(20,10))
//SYSEDIT2   DD    DSN=&EDIT2,UNIT=SYSDA,SPACE=(6144,(50,20))
```

- SYSEDIT is a sample ddname for the temporary utility work data set (&EDIT) that is allocated to a user when editing a new data set.

- SYSEDIT2 is a sample ddname for the temporary utility work data set (&EDIT2) that is allocated to a user when editing an existing data set.

If a user allocates a temporary data set from the LOGON procedure, TSO EDIT still tries to allocate a workfile with the user's user ID appended to the same data set name. If the user sets the user ID as the high-level qualifier when allocating the data set, an error message is issued.

Permanent Utility Work Data Sets — Name the data sets

DSN = userid.EDITUTL1
and
DSN = userid.EDITUTL2

Catalog the data sets prior to the edit session. No DD statements are required in the user's logon procedure because the catalog is searched for their location.

- userid.EDITUTL1 is allocated to a user when editing a new data set.
- userid.EDITUTL2 is allocated to a user when editing an existing data set.

Changing the BRODCAST Limit, LOGON Limits, and EDIT Defaults with Macros (MVS/XA Only)

Changes to the broadcast message processing parameter, LOGON limit parameters, and EDIT data set type parameters do not require a system generation. You can change these parameters simply by submitting jobs that perform the updates.

If you used the TSO, EDIT, or SCHEDULR SYSGEN macros to modify the parameters noted above, IBM defaults for TSO/E Release 3 equivalent parameters overlay your changes. If you want to change the IBM defaults, do so after system generation. SYS1.SAMPLIB contains sample macros and JCL that you can use to run background jobs to change the defaults. (If you use the IKJBCAST, IKJTSO, and IKJEDIT macros to change the IBM-supplied defaults, future SYSGENS and IOGENS will not replace your new defaults.)

Broadcast Message Processing Parameter

To change the number of 130-byte records set aside for broadcast messages in the SYS1.BRODCAST system data set, use the IKJBCAST macro. (THE IKJBCAST macro replaces the BCLMT operand on the SCHEDULR SYSGEN macro.)

For the syntax of the IKJBCAST macro, see Volume 3. A sample IKJBCAST macro appears in SYS1.SAMPLIB, member BCSTSAMP (for SMP/4) or BCSTSMPE (for SMP/E). For examples of changing the sample macros, see Volume 2.

LOGON Limit Parameters

To change the number of lines you can enter in a LOGON attempt before TSO cancels the attempt, or the number of seconds you can wait during a LOGON attempt before you see "LOGON proceeding", use the IKJTSO macro. (The IKJTSO macro replaces the TSO SYSGEN macro.)

For the syntax of the IKJTSO macro, see Volume 3. A sample IKJTSO macro appears in SYS1.SAMPLIB, member TSOSAMP (for SMP/4) or TSOSMPE (for SMP/E). For examples of changing the sample macros, see Volume 2.

EDIT Data Set Type Parameters

To change the default physical characteristics and processing attributes of different data set types, use the IKJEDIT macro. (The IKJEDIT macro replaces the EDIT SYSGEN macro.)

For the syntax of the IKJEDIT macro, see Volume 3. A sample IKJEDIT macro appears in SYS1.SAMPLIB, member EDITSAMP (for SMP/4) or EDITSMPE (for SMP/E). For examples of changing the sample macros, see Volume 2.

Chapter 4. Additional Modifications: An Overview

The following sections briefly describe modifications covered in detail in Volume 2. The section "Transmitting Data Through a Network" outlines the modifications required to permit users to transmit data through a network. The sections "Exit Routines" on page 4-4 and "User Modifications" on page 4-14 give an overview of the exit routines and user modifications covered in Volume 2.

Transmitting Data Through a Network

A TSO user can transmit data through a network by two different methods:

- Submitting a background (batch) job using the JES2 /*XMIT or the JES3 // XMIT facility
- Using the Interactive Data Transmission Facility.

JES Facilities: the JES2 /*XMIT and JES3 // XMIT Statements

JES2 and JES3 provide facilities that allow a job stream or data to be transmitted through a network, from a sending node to a target node. The target node can be a valid VM, JES2, or JES3 system.

The basic format of the JES2 job stream is:

```
//jobname JOB ...
/*XMIT    dest [DLM=xx]
          data
           or
          a job
/*   (or the delimiter specified by the DLM= operand)
```

If there are intermediate nodes in the network, the statements are passed through those nodes without modification or checking. To ensure that the target node (*dest*) is valid, you can write a SUBMIT exit routine that sets bit 5 in byte 0 of word 5. Exit routines for SUBMIT are described briefly in "Exit Routines" on page 4-4. For more detailed information, see Volume 2. For additional information on the /*XMIT function, see *System Programming Library: JES2 Initialization and Tuning*.

The basic format of the JES3 job stream is:

```
//jobname JOB ...
//(name) XMIT DEST=node-name(.VMuserid) (,DLM=xx)
        data
        or a
        job
/*    (or the delimeter specified by the DLM= operand)
```

DEST and DLM are keyword parameters and may be coded in any sequence. DEST identifies the target node name and must be specified. See *MVS/XA JCL* for more information about the // XMIT function.

When a user submits a job stream for transmission, the SUBMIT exit (either the IBM-supplied exit or a user-written exit) is passed only the initial JOB statement. Statements between the /*XMIT statement and the delimiter, or the // XMIT statement and delimiter, are never passed to the exit routine.

At the target node, you can write a JES2 or JES3 exit routine to make sure that the transmitted job statements are valid. For example, the JES exit routine might check for valid accounting information or DD statements. The exit might then approve, reject, or modify any of the statements.

Installations must provide the JES2 or JES3 exit to notify a user that transmitted data has arrived. The JES2 exit is exit 13; the JES3 exit is exit IATUX42.

For additional information on JES2 or JES3 exits, see *System Programming Library: JES2 User Modifications and Macros* or *System Programming Library: JES3 User Modifications and Macros.*

Interactive Data Transmission Facility

The Interactive Data Transmission Facility allows users to send data to each other. You can use two commands, TRANSMIT and RECEIVE, to send or receive sequential or partitioned data sets.

For example, when you issue the TRANSMIT command, the data set you requested to transmit is read, the records are converted to a format suitable for transmission, header information is added, and the data is routed to a class B SYSOUT PUNCH file that is directed to the receiving node and userid. The data routed to the receiving node is queued on the JES SPOOL.

When a user at the receiving userid issues the RECEIVE command, that user is told that a data set has arrived, and is given the name of the data set and the node and userid of the sender. The user might then specify the name of a data set in which the data should be stored. The data is restored to its original format, and is written into the data set indicated by the user.

Considerations

When you are deciding whether to use the Interactive Data Transmission Facility, consider the following:

- The requirements for writing an installation options CSECT
- The requirements for running the RECEIVE command in background mode.
- The considerations for writing TSO exit routines
- The requirements for writing certain utility programs
- The additional programming requirements for JES2 or JES3.

Installation Options CSECT: Before you can use the Interactive Data Transmission Facility, you must supply an installation options CSECT. This CSECT, INMXPARM, contains installation-defined defaults and controls for both the TRANSMIT and RECEIVE commands. The format and syntax of the macros needed to write the CSECT are presented in Volume 3.

Running the RECEIVE Command in the Background: Before users can issue the RECEIVE command in the background, you must define their user IDs to either RACF or a functionally equivalent program. (The user ID issuing the RECEIVE command must be placed in the ASXBUSER field of the ASXB control block before the TMP Initialization Routine receives control; RACF does this for you.)

Exit Routines: A number of exit routines can give your installation a way to monitor transmission activity or alter the way TRANSMIT and RECEIVE perform some operations. For more information, see "Exit Routines" on page 4-4.

Utility Programs: If you intend to transmit data set types not supported by the Interactive Data Transmission Facility (ISAM and VSAM data sets, or data sets with user labels or keys), you must write two utilities - one to unload the data into a sequential data set, and another to reload the data to restore it to its original form.

JES2 and JES3 Programming Considerations: If you are currently using the JES2 initialization statement, &TSU NOOUTPUT or TSUCLASS OUTPUT=NO, the TRANSMIT command does not work. The SYSOUT data resulting from the use of the TRANSMIT command is deleted. See *JES2 Initialization and Tuning* for information about the JES2 initialization statement.

If you are using the JES2 initialization parameter, &ESTPUN, to limit punch-card output, the limit cannot be altered for TSO sessions. Therefore, set the value of &ESTPUN high to avoid abends during TRANSMIT command usage.

JES2 and JES3 are unaware that data sets being routed among nodes in a network are enroute to TSO terminals. Thus, no checking for valid user IDs is performed during transmission. It is recommended that you check periodically for transmissions to user IDs that are not valid.

- If you have JES2, use the $DF command to list all data sets in transit. The system displays a separate line for each external writer name (target user ID).

Then, use the RECEIVE command with the USERID parameter to delete or reroute data sets with invalid destinations.

- If you have JES3, you should write a JES3 exit routine to check for invalid destinations. For more information about the JES3 exit routine, see *JES3 User Modifications and Macros.*

For additional considerations relative to JES2/JES3 exits, see "Exit Routines."

Using the Interactive Data Transmission Facility Within a Single Node

You can use the facilities of TRANSMIT and RECEIVE within a single node (no networking with other systems) to allow easy transmission of messages or files between TSO users. In a single-node environment, there are no requirements for special levels of JES. Standard interfaces to system allocation and the external writer TSO interface are used, and these interfaces are independent of the level of JES.

When sending and receiving messages, the TRANSMIT and RECEIVE commands have the following advantages over the SEND command:

- You can compose multiple line messages and send them to other users.

- You can transmit messages using nicknames or distribution lists.

- The messages you send and receive are logged in a disk data set, with a date and time stamp, so that you can recall them.

- You can receive messages at your convenience.

When a file is transmitted, TRANSMIT sends the characteristics of the file. This allows RECEIVE to restore the file format without requiring pre-allocation or prior information about the file format.

In a security environment (RACF), users can exchange data without the need to provide access to their data sets. This reduces the risk of unauthorized access or change.

Exit Routines

You can write exit routines to control the use of TSO resources and functions by TSO users. This section:

- Presents an overview of each exit routine

- Describes the purpose of each exit routine

- Describes what actions each exit routine can take based on installation options.

Exit routines for the following commands, functions, and subcommands are described in this section:

- LOGON Pre-prompt
- RENUM, MOVE, and COPY subcommands of EDIT
- OUTPUT, STATUS, and CANCEL commands
- SUBMIT command
- TRANSMIT and RECEIVE commands
- Syntax checkers
- Session manager
- ADRS (VS APL)
- Names
- Termination CLIST.

Detailed information on writing each exit routine is in Volume 2.

For some of the exit routines, IBM supplies a default exit. Each of the default exits is also discussed in this section.

LOGON Pre-prompt

The LOGON command initiates a terminal session by supplying the system with certain basic information: userid, password, account number, procedure name, region size, and performance group. You can write an exit routine to specify these values for the LOGON command and thereby customize the LOGON procedure for your installation's terminal users. The exit routine can supply system and user attributes for the protected step control block (PSCB), the generic group name, the performance group, and the default SYSOUT destination. The exit routine can also provide JCL statements to be used instead of the JOB and EXEC statements constructed by the LOGON processor.

If your LOGON pre-prompt exit routine requests that the UADS data set not be opened (for verifying the user ID, account number, and other attributes), change your exit routine to get the I/O reduction for the LISTBC command. Set the CTLNUADE bit so the logon program can open the UADS and retrieve the address of the mail directory.

Notes:

1. *The LOGON processor constructs a standard JOB card and, if SMF audit-exits are being taken, passes the JOB card directly to SMF. If a terminal user must insert installation-dependent information, a LOGON pre-prompt exit is required.*

2. *Do not activate the ? Prompt Help Facility from the exit routine or the exit will abend with a 0B0 system completion code.*

3. *By default, user passwords defined to RACF are not stored in the TSB. If users at your installation do not route jobs to other nodes to execute, your installation's SUBMIT exit need not refer to the TSBPSWD field for the password. The password is not required on the JOB card for verification when the job executes in the same node in which it was submitted. If your installation has RACF users that are defined with the same userid and password on more than one node in a network, and those users use the JES control statement //*ROUTE XEQ to route jobs from one node to another, you should code a*

LOGON pre-prompt exit routine that sets bit 3 of byte 2 of the control switches. This indicates to the LOGON processor that it should store passwords in the TSB. Your installation's SUBMIT exit can then retrieve the password from the TSBPSWD field, and the exit can modify the JOB card to include the password as needed.

RENUM, MOVE and COPY Subcommands of EDIT

The RENUM subcommand of EDIT assigns a line number to each record in data sets that do not have line numbers. It also renumbers records in data sets that have line numbers. The MOVE subcommand of EDIT moves lines in a data set from one location to another, renumbering the lines in the data set as needed. The COPY subcommand of EDIT copies lines in a data set, renumbering the lines in the data set as needed.

However, the RENUM, MOVE and COPY subcommand processors cannot handle certain data set types (for example, VSBASIC data sets) that can have embedded line references (using a line number as a destination or statement "label" in a statement that passes control, such as a GOTO statement). Thus, your installation may want an exit routine that renumbers, moves, and copies records in a data set in storage, adjusting line references as needed. You can also use an exit routine to flag a statement (for example, by adding a comment at the end of the statement) or to allow a numbering scheme different from the standard.

For the types of data sets to which the exit routines apply, the name of the exit routine must be supplied as the DATEXIT value on the:

- EDIT macro instruction, during system generation (for MVS/370)
- IKJEDIT macro instruction (for MVS/XA).

For more information about the IKJEDIT macro, see "Changing the BRODCAST Limit, LOGON Limits, and EDIT Defaults with Macros (MVS/XA Only)." Volume 2 contains examples of changing EDIT defaults with the IKJEDIT macro. Volume 3 provides the syntax of the macro.

OUTPUT, STATUS, and CANCEL Commands

You can write an exit routine to control the conditions under which OUTPUT, STATUS, and CANCEL commands are allowed. The exit routine can restrict a terminal user from obtaining status for a job, from canceling another terminal user's job, or from receiving the output of another terminal user's job. The exit routine can restrict a terminal user from directing a job's output to a specific data set or from changing a job's output class. Also, the exit routine can send a message to the user's terminal and can request a response. In determining whether to allow one of the commands to execute, the exit routine can verify the userid, jobname, and jobid. The exit routine is not entered for a STATUS command with no operands.

User exit IKJEFF53 (OUTPUT/STATUS/CANCEL exit routine) receives control in key 8, rather than key 1. If necessary, this user-written exit routine can change to a different key upon receiving control, but **must** return to key 8 upon completion.

IBM-Supplied Exit Routine

If you do not supply an exit routine, an IBM-supplied exit routine common to all three command processors receives control. The IBM-supplied exit routine allows a terminal user to obtain the status of any job in the system. However, the exit routine restricts a terminal user from canceling, or obtaining the output of, any other user's job.

The IBM-supplied exit routine checks the userid specified on the LOGON command against the jobname or jobnames entered on the CANCEL or OUTPUT command. (The jobname must equal the userid plus at least one character for CANCEL, and must be the userid or must start with the userid for OUTPUT.) If the terminal user enters an invalid jobname, the IBM-supplied exit routine sets up an error message to be issued by the appropriate command processor and passes a return code that tells the command processor to ignore the CANCEL or OUTPUT request for that jobname. If any other jobnames are listed on the same command, the IBM-supplied exit routine processes them in order.

SUBMIT Command

The SUBMIT command allows a user to initiate a batch job from a terminal. The SUBMIT command processor writes the user-specified data set(s), which consist of JCL statements and input data, into an internal reader of the job entry subsystem. Using an exit routine, your installation can accept, reject, or modify the JCL statements being submitted.

When the first JOB statement is read or generated, the SUBMIT command processor invokes the exit routine. As a default, only JOB cards and JOB card continuations are presented to the exit routine. The routine can indicate dynamically additional types of JCL statements that it wishes to inspect as the statements are read from the input data set(s). The routine can delete or change the current statement, and can generate new statements or continuations to follow the current one. The routine can also send a message to the user's terminal and can request a response. In addition, the exit routine can verify the jobname and userid, and can cancel a SUBMIT request.

By default, user passwords defined to RACF are not stored in the TSB. If users at your installation do not route jobs to other nodes to execute, your installation's SUBMIT exit need not refer to the TSBPSWD field for the password. The password is not required on the JOB card for verification when the job executes in the same node in which it was submitted. If your installation has RACF users defined with the same userid and password on more than one node in a network, and those users use the JES control statement //*ROUTE XEQ to route jobs from one node to another, you should code a LOGON pre-prompt exit routine that sets bit 3 of byte 2 of the control switches. This indicates to the LOGON processor that it should store passwords in the TSB. Your installation's SUBMIT exit can then retrieve the password from the TSBPSWD field, and the exit can modify the JOB card to include the password as needed. For more information about the LOGON pre-prompt exit, see "LOGON Pre-prompt."

User exit IKJEFF10 (SUBMIT exit routine) receives control in key 8, rather than key 1. If necessary, this user-written exit routine can change to a different key upon receiving control, but **must** return to key 8 upon completion.

IBM-Supplied Exit Routine

If you do not supply an exit routine, an IBM-supplied exit receives control. The IBM-supplied exit does not check JCL. It is entered once for each SUBMIT command (JOB statement). It turns off all switches that cause it to receive control, and sets the return code to zero.

TRANSMIT and RECEIVE Commands

A number of user exits can give your installation a way to monitor transmission activity or to alter the way TRANSMIT and RECEIVE perform some operations. The individual exits are:

- **TRANSMIT startup exit (INMXZ01)**: receives control after the TRANSMIT command is scanned and a list of addressees is built
- **TRANSMIT encryption exit (INMXZ03)**: receives control just before the Access Method Services REPRO command is invoked to encipher a data set
- **TRANSMIT termination exit (INMXZ02)**: receives control after all processing of the TRANSMIT command is complete
- **RECEIVE initialization exit (INMRZ01)**: receives control after the RECEIVE command processor parses the RECEIVE command, but before it takes any other action
- **RECEIVE data set pre-processing exit (INMRZ11)**: receives control just before the prompt to the receiver asking where the data is to be stored
- **RECEIVE data set decryption exit (INMRZ13)**: receives control just before the Access Method Services REPRO command is invoked to decipher the data set
- **RECEIVE notification exit (INMRZ04)**: receives control when a sender is to be notified that transmitted data has been received
- **RECEIVE data set post-processing exit (INMRZ12)**: receives control after all processing for a data set (except deletion) is complete
- **RECEIVE termination exit (INMRZ02)**: receives control after all processing of the RECEIVE command is complete.

Possible Uses of Exits

Some possible uses of the TRANSMIT and RECEIVE exit routines are described in the following paragraphs.

Checking Authorization: You can use the TRANSMIT startup exit, INMXZ01, and the RECEIVE startup exit, INMRZ01, to determine which TSO users are authorized to use the Interactive Data Transmission Facility. Further, you can use INMXZ01 and the RECEIVE data set pre-processing exit, INMRZ11, to control which users can use a particular network path. INMXZ01 is passed a list of all the addressees of a message by node and userid, and INMRZ11 can check the source of a message before delivering it to the user. The RECEIVE startup exit, INMRZ01, can also determine who has the authority to clean up stray messages by overriding the default, which is to receive only messages for one's own userid.

Alternate Data Format: To support data types not supported by the Interactive Data Transmission Facility, you need to use three exits in addition to two user-written utilities for unloading and reloading the data sets. The three exits are the TRANSMIT startup exit (INMXZ01), the RECEIVE data set pre-processing exit (INMRZ11), and the RECEIVE data set post-processing exit (INMRZ12).

INMXZ01 must detect that a special data set type is being processed and invoke your unload utility to copy the data into a pre-allocated sequential data set. (Allocate the data set as a temporary data set.) INMXZ01 passes the DDNAME of the temporary data set back to TRANSMIT and the data is transmitted. INMXZ01 may also pass up to 10 local control records, which are transmitted with the data.

When the transmission reaches the RECEIVE data set pre-processing exit, INMRZ11, the exit is passed the original DSNAME entered by the sender and any local control records created by INMXZ01. INMRZ11 instructs RECEIVE to ignore the target DSNAME entered by the sender, and writes the received data into a pre-allocated temporary data set. This data set is passed to the RECEIVE data set post-processing exit, INMRZ12, which invokes your reload utility to restore the data to its original format and write it into the receiving user's target data set.

This process need not be limited to transmitting entire data sets. It can also be used to transmit segments of a data base. INMXZ01 can extract the required data from the data base and write it into a temporary sequential data set for transmission. On the receiving end, the RECEIVE exits could re-build the data and write it into the receiving user's data base.

Accounting: The TRANSMIT termination exit, INMXZ02, and the RECEIVE data set post-processing exit, INMRZ12, provide information you can use for recording network usage. At both these exits, you have the information required to determine the volume and direction of network traffic. Use the SMFWTM or SMFEWTM macro to write your own SMF records from either of the exits.

IBM-Supplied Exit Routine

For each of the TRANSMIT and RECEIVE exit routines, IBM supplies an exit that sets register 15 to zero and returns to the caller.

Checking for Invalid Destinations

JES2 and JES3 are unaware that data sets being routed among nodes in a network are enroute to other nodes. Thus, no checking for invalid destination is performed during transmission. If you have JES3, you should write a JES3 exit routine to check for invalid destinations. See *JES3 User Modifications and Macros* for more information about the JES3 exit. If you have JES2, you can check for invalid destinations using the $DF command. See "JES2 and JES3 Programming Considerations" on page 4-3 for more information about the $DF command.

To notify an addressee that transmitted data has arrived, JES invokes an installation-written exit routine. If the exit routine does not exist, the addressee is not notified. See *JES2 User Modifications and Macros* or *JES3 User Modifications and Macros* for additional information on the exit.

Syntax Checkers

For IBM-supplied data set types and the associated syntax checkers, each syntax checker determines the attributes of the associated data set type by referring to information that EDIT initialization sets in the option word, a fullword in the parameter list of the syntax checker. However, for an installation's own data set types and syntax checkers, EDIT initialization does not place the attribute information in the option word. You can write an exit routine to fill in the option word for the syntax checker according to information entered by the user.

When a user specifies your installation's data-set-type keyword on the EDIT command, the user can also specify a subfield. The subfield can contain any valid alphameric data not exceeding 256 characters. This information is passed to the exit routine, to be interpreted and encoded into the option word.

For the types of data sets to which the exit routine applies, the name of the exit routine must have been supplied as the USREXT value on the:

- EDIT macro instruction, during system generation (for MVS/370)
- IKJEDIT macro instruction (for MVS/XA).

For more information about the IKJEDIT macro, see "Changing the BRODCAST Limit, LOGON Limits, and EDIT Defaults with Macros (MVS/XA Only)." For examples of changing EDIT defaults with the IKJEDIT macro, see Volume 2. For the syntax of the IKJEDIT macro, see Volume 3.

Session Manager

Your installation can use three types of installation-written exit routines to monitor users' interaction with the system:

- An exit allowing your installation to indicate which session manager streams are to be monitored for users' sessions, and whether the session manager should log line-mode output while users are executing full-screen programs. The session manager initialization routine (ADFMDF01 in MVS/370 or ADFMDF0A in MVS/XA) calls this exit.

- An exit monitoring streams. Two routines call this exit: ADFIMPUT, which puts a line into a stream, and ADFIMGET, which gets a line of input from a stream.

  Whenever a line is to be put into a stream (excluding the TSO input stream), ADFIMPUT determines if the stream is being monitored. If it is, then ADFIMPUT calls the exit routine and passes the line, an indication of the stream being monitored, a timestamp, and other information. The line is not put into the stream until after the exit routine is called.

  Whenever the next line of input is being retrieved from a stream, the session manager determines if the installation is monitoring the stream, and calls the exit routine if it is.

- An exit allowing your installation to do required processing during termination of the session manager. The session manager initialization routine (ADFMDF01 in MVS/370, ADFMDF0A in MVS/XA) calls this exit when the user enters either the session manager END command or the TSO LOGOFF command.

Depending upon how and where you supply the exits, you can provide a wide range of monitoring capabilities. You can monitor all users in the same way, monitor each class of users differently, or group various classes of users together and monitor the groups differently. For example, you can monitor all the users with the same LOGON procedure in the same way.

Exit Usage Considerations

This section describes some considerations for writing session manager exit routines.

- You can monitor TSO input to intercept any commands that a particular user or group of users is restricted from issuing. When the exit intercepts one of these commands, it can change the command to blanks or some other characters, which effectively discards the command. If you choose to eliminate commands in this way, do not change the length of the line.

- When a TSO input line is intercepted, it is being placed in the TSO input stream. The command may not be ready for execution, since other commands may be stacked ahead of it. Therefore, if the exit routine is to perform some special action when a particular command is entered, the command must be detected as it is copied to the TSO output stream. At that point the session manager is returning that line of input to TSO for execution.

- If the exit routine is to intercept a particular command as it is copied to the output stream, you do not have to monitor every line of TSO output. Monitor only TSO input until the exit detects the command. Then the exit routine can begin monitoring TSO output until the command is copied to the output stream. Then the exit routine can perform some special action and return to monitoring only TSO input.

  To change dynamically which streams are monitored, have ADFEXIT1 save the address of the stream mapping as part of the installation data. Then ADFEXIT2 can change the stream mapping that defines which streams are being monitored. For example, if you are monitoring TSO input and detect a particular command, you can start monitoring the TSO output stream. When the next command is retrieved for TSO input, you can go back to monitoring only TSO input.

- ADFIMPUT is called each time a line is put into any stream, and ADFIMPUT calls ADFEXIT2. Although this is a mainline path, there is a negligible increase in the number of instructions in ADFIMPUT to determine if ADFEXIT2 is to be called and to call it. However, you should consider the following:

  - Depending upon what streams you are monitoring, ADFEXIT2 could be called every time through ADFIMPUT. If this happens, and ADFEXIT2 has a considerable amount of function, then performance could be degraded.

    - ADFEXIT2 can change dynamically the mapping that defines which streams are to be monitored, to reduce the number of unnecessary calls to it.

    - If one group of users is to be monitored differently, then the overhead of monitoring one group does not have to affect others. Different sets of exit routines can be supplied in the groups' default environment modules.

- You can have ADFEXIT2 retain a log of a user's TSO session by monitoring the TSO input and output streams and writing each line to a data set. When the termination exit (ADFEXIT3) is called, the routine can create a hardcopy of the data set.

  ADFEXIT2 should not do any I/O operations. To prevent performance degradation, ADFEXIT1 can attach instead another task during initialization to handle I/O requests from ADFEXIT2. Whenever ADFEXIT2 is ready to write a record, it can post the I/O task and pass it a buffer with the record. ADFEXIT2 can then return to ADFIMPUT, allowing the session manager to continue while the I/O task is asynchronously writing the record.

  The task attached by ADFEXIT1 runs without holding the local lock; therefore, ADFEXIT2 never has to release it. Releasing the local lock in ADFEXIT2 causes unpredictable serialization problems within the session manager.

- If the user has the same stream defined as the input or output stream of two or three functions, the following can happen:

    - ADFIMPUT calls ADFEXIT2 for *every* line going to a monitored stream, even if the line is being put into the stream as the result of another function. For example, if MSG output is being monitored, and the user has TSOOUT as both the TSO and MSG output stream, then ADFEXIT2 is called for all TSO output, since all such lines have a target stream equal to the MSG output stream.

    - The bit mapping passed to ADFEXIT2 indicates only the stream for which a match was found between the target stream and the function's stream. (In the last example, the bit mapping always indicates MSG output, even though many of the lines were TSO output lines).

- A user can fabricate the entire TSO output stream by issuing session manager commands to put the expected lines into the TSO output stream. If this is a potential problem, the exit routine can save all TSO input in a data set. Later, you can verify which TSO commands the user issued.

- You can use the time stamp provided with each line passed to ADFEXIT2 to monitor the execution time of particular commands or to provide a time stamp on lines saved for subsequent hardcopy listings:

    - ADFEXIT2 can monitor TSO input and, on intercepting a particular command, save the time stamp and begin monitoring TSO output. When ADFEXIT2 detects the READY mode message, it can calculate the difference between the saved time stamp and the time stamp provided with the READY mode message line to determine the execution time of the command.

- If a TSO session is being saved for subsequent hardcopy, then the time stamp passed to ADFEXIT2 for each line can be converted to a printable time stamp and written with each line. In reviewing the session, you can then see at what time certain operations were performed.

Location of Session Manager Exits

You can put each of the exit routines in either or both of the following load modules:

- You can link edit any or all of the exits into the ADFMDF01 load module. If you link edit exits into ADFMDF01, name the exit called during session manager initialization ADFEXIT1. Name the exit called by ADFIMPUT and ADFIMGET ADFEXIT2. Name the termination exit ADFEXIT3.

- You can link edit any or all of the exits into any default environment module. The IBM-supplied default environment module is named ADFMDFLT. If you link edit exits into a default environment module, there is no restriction on the names of the exits or on the name of the default environment module, but offset X'5C' into the defaults CSECT must contain the address of the exit that the session manager initialization routine calls. Offset X'60' must contain the address of the exit that ADFIMPUT and ADFIMGET call. Offset X'64' must contain the address of the termination exit.

If you supply an initialization exit routine in the defaults module, it overrides the exit named ADFEXIT1 in the ADFMDF01 load module. If the exit routine to be called by ADFIMPUT and ADFIMGET is in the defaults module, it overrides ADFEXIT2 in the ADFMDF01 load module. Likewise, if a termination exit is in the defaults module, then it overrides ADFEXIT3 in the ADFMDF01 load module.

You can place the exits in the same or different load modules, but in MVS/XA, be sure that all exits in the same load module have the same AMODE.

ADRS

When a user selects the ADRS option on the DATA ANALYSIS/REPORT CREATION SERVICES PANEL, VS APL is invoked to load ADRS. An exit routine can be invoked before VS APL is invoked. You can use the exit routine to display panels, allocate data sets, or perform any other services required.

If the return code from the exit routine is zero, VS APL is invoked, and the workspace is loaded. If the return code is not zero, the ADRS option terminates.

Space Allocation for Data Sets Allocated by VM/PC Servers (MVS/XA Only)

VM/PC servers dynamically allocate MVS data sets with a TSO ALLOCATE command. All of the data sets are allocated with the same space parameters. The default values for these space parameters are:

- For sequential data sets, TRACKS SPACE(10,2)
- For partitioned data sets, TRACKS SPACE(15,3) DIR(27).

You can change the values of these parameters from an installation-written exit.

Names

The Information Center Facility names service is used to maintain a system names directory and users' private names directories. These names directories can help users locate other users, and assist them in transmitting messages through the system. The names user exit is invoked each time a name or group is added, modified or deleted in either the master directory or a private directory. The exit can be used to create and maintain a parallel data base of names, such as a SCRIPT/VS names macro library. The exit can access a table containing the following information:

- The dialog from which the exit is being invoked (administrator or user)
- The type of update just performed
- The name of the table that was updated
- The name of the library in which the table can be found
- The entry that was added, modified, or deleted.

Termination CLIST

When the Information Center Facility or another ISPF application invokes CLIST ICQGCC11 to perform termination processing, a termination user exit can be invoked. The user exit can be used to perform such tasks as freeing data sets used during the session, or executing commands such as RECEIVE after leaving the Information Center Facility.

User Modifications

To provide particular installation-dependent function and control, you may have to modify or add to certain segments of IBM-supplied code before placing TSO/E into production. This section presents an overview of such modifications and additions. See Volume 2 for more information about modifications and additions and how to code them.

Executing Authorized Programs and Commands

To allow users in problem program state or programs using the TSO service routine to execute authorized commands and programs, link edit installation access lists APFPTABL (authorized programs), APFTTABL (authorized programs), and APFCTABL (authorized commands) into load module IKJTABLS (for MVS/XA) or load module IKJEFT02 (for MVS/370). Afterwards, TSO users can execute authorized and nonauthorized programs and commands within a single TSO session. APFPTABL is in CSECT IKJEFTE8. APFTTABL is in CSECT IKJEFTAP. APFCTABL is in CSECT IKJEFTE2. SYS1.LPALIB contains the load modules IKJTABLS and IKJEFT02.

If your installation does not supply program names in lists APFPTABL or APFTTABL, then users can execute only nonauthorized programs. The restriction on execution of authorized programs includes the utility programs IEBCOPY, IEHMOVE, IEHATLAS, IEHINITT, IEHPROGM, and DFDSS. (TSO users can execute these programs using the SUBMIT command.) To use the FIB commands (STATUS, SUBMIT, OUTPUT, and CANCEL), your installation must also place IKJEFF76 in an access list.

If your installation does not supply command names in list AFPCTABL, then users can execute only nonauthorized commands. The restriction on execution of authorized commands includes TRANSMIT, XMIT, and RECEIVE. If an authorized program calls an authorized CLIST, the EXEC command must also be authorized.

Each entry in lists APFCTABL, APFTTABL, and APFPTABL must be 8-bytes long. If necessary, pad entries with blanks.

Adding Commands Not Supported by the TMP in the Background

The OPERATOR and TERMINAL commands, and their aliases OPER and TERM, are not supported in the background. The names of the unsupported commands are in a list distributed by IBM in CSECT IKJEFTNS, which is in module IKJTABLS for MVS/XA and in module IKJEFT02 for MVS/370. You can add additional command names and aliases to the list. Also, any command processors that issue a TGET or a TPUT will not function correctly in the background. (The I/O operations are effectively treated as no-ops.)

When a command is executed in the background, the TMP scans the list and compares the command name to the entries in the list. If a match is found, the command is unsupported. The TMP issues a message identifying the invalid command, terminates its own processing, and returns control to the system.

Adding Subcommands to the EDIT Command

When a terminal user enters an EDIT subcommand, the EDIT controller (load module IKJEBEMA) invokes the appropriate subcommand processor. IKJEBMA9, a CSECT within load module IKJEBEMA, is reserved as a table of installation-written EDIT subcommand names. Thus, if your installation requires an editing function not provided by the IBM-supplied EDIT subcommands, you can write one or more subcommand processors and add their names to the table using the IKJEBEST macro instruction.

Building Translation Tables for TSO/VTAM Users

Translation tables allow TSO/VTAM users to replace internally characters that are not available on a keyboard with characters that are available. For example, the characters "<," ">," and "|," which are not available on correspondence keyboards, can be represented by other characters that are available, such as "ffl," "“," and "!." The CHAR and TRAN operands of the TERMINAL command are used to specify replacement characters. The TERMINAL command invokes the STTRAN macro instruction to set up the translation tables.

When you specify character translation, installation-written translation tables or default (IBM-supplied) translation tables are used. Default translation tables are a part of the TSO/VTAM program; they translate each character to itself. The different combinations of the CHAR and TRAN operands that can be specified by users are:

- CHAR (*characters*) alone: causes a copy of the default translation tables (in the user's storage) to be updated according to the *characters* specified. The updated tables are used to translate all inbound and outbound characters, for as long as the terminal session lasts, or until NOCHAR or NOTRAN is specified.

- TRAN (*name*) alone: causes a copy of the translation tables located in *name* to be used to translate all inbound and outbound characters, for as long as the terminal session lasts, or until NOTRAN is specified.

- CHAR (*characters*) and TRAN (*name*): causes a copy of the translation tables located in *name* to be updated according to the *characters* specified. The updated tables are used to translate all inbound and outbound characters, for as long as the terminal session lasts, or until NOCHAR or NOTRAN is specified.

When translation tables are in use, input translations take place after TSO/VTAM translates the line code to EBCDIC characters, and output translations take place before TSO/VTAM translates the EBCDIC characters to line code.

Writing a Syntax Checker

Each IBM-supplied syntax checker is associated with one or more IBM-supplied data set types. If your installation defines its own data set type(s) for the EDIT command, you can write a syntax checker for each new data set type. Each syntax checker and its associated data set type must be defined using the:

- EDIT macro instruction during system generation, for MVS/370 (see *System Generation Reference*).

- IKJEDIT macro instruction, for MVS/XA.

For more information about the IKJEDIT macro, see "Changing the BRODCAST Limit, LOGON Limits, and EDIT Defaults with Macros (MVS/XA Only)." Volume 2 contains examples of changing parameters with the IKJEDIT macro. Volume 3 contains the syntax of the macro.

Then when users enter lines of input to installation-defined data set types, they can request that each of the lines entered in input mode be scanned immediately for syntax errors.

Modifying the Session Manager

IBM supplies a default environment module, ADFMDFLT, for the session manager. This module contains the tables and data needed for building the TSO user's default screen layout, PF key definitions, and so on.

If you want to use one default environment for all your TSO users, you can modify the IBM-supplied defaults module to refine the environment for your installation's particular requirements. If you find that one environment does not satisfy the requirements of all TSO users, you can create multiple LOGON procedures, each one specifying a different installation-supplied default environment module.

Customizing Information Center Facility Functions

The Information Center Facility allows easy customization. You can:

- Add installation-supplied functions, or delete or replace options that are not available at your installation

- Modify the start-up processing, or add processing for termination

- Add commands to the command table

- Change the defaults for space management

- Change the default values of variables for different functions in the Information Center Facility.

Modifying APLDI II for END PF Key Support

On a system-wide basis, you can modify APL Data Interface II (APLDI-II) to alter the way users leave APLDI and return to an invoking program, for example the Information Center Facility. The unmodified version for APLDI requires users to press the END PF key twice when responding to an APLDI prompt. After installing the modification, users press the END PF key only once to leave APLDI, save the workspace, terminate APL, and return to the invoking program.

Defining Output Descriptors for the ALLOCATE Command

You can define output descriptors for your installation by placing OUTPUT JCL statements in LOGON procedures. The output descriptors associate printer locations and output characteristics with names. Users specify the names of output descriptors when allocating SYSOUT data sets rather than coding the output parameters themselves.

Chapter 5. Diagnosing Problems in the Information Center Facility

This chapter describes commands you can use to help diagnose the causes and locations of problems you encounter in the Information Center Facility. You can find charts of the Information Center Facility CLISTs and panels in Volume 2. For additional information about error messages, see the message help online or *TSO Messages*. For information about CLISTs, see *TSO/E CLISTs: Implementation and Reference*. For information about ISPF dialog and table services, see *ISPF Dialog Management Services*.

ISPF PANELID Command

If there is a problem with a panel in the Information Center Facility, you can display the ID of the panel by entering the command "PANELID" or "PANELID ON" on the COMMAND or OPTION line of the panel. After you enter this command, all the panels displayed show the panel identifier in the upper left corner of the screen. You can suppress the display of the panel ID again by entering "PANELID OFF."

TRACE Commands

You can use the trace commands provided with the Information Center Facility to diagnose problems in the facility's CLISTs. The commands provide three levels of tracing:

Level 1 - activated by the TRACE1 command
Level 2 - activated by the TRACE2 command
Level 3 - activated by the TRACE3 command

You can turn tracing off by entering the TRACEOFF command.

When a CLIST is invoked for execution and tracing is active, the CLIST writes its name to the screen using a CLIST WRITE statement. The information appears as follows:

```
************************
**  CLIST=clistname  ***
************************
```

The ISPF dialog service LOG is issued at the beginning and termination of a CLIST. The following message is written to the ISPF log when a CLIST begins execution:

```
CLIST clistname beginning execution.  ICQGA000
```

At the termination of a CLIST, the following message is written:

```
CLIST clistname exiting; final condition code n.  ICQGA001
```

The executing CLIST determines the final condition code.

Note: You can enter the TRACE commands on the OPTION line of any Information Center Facility selection panel except the following:

ICQABM30
ICQCLM00

TRACE1 Command — Level 1

Level 1 traces the control flow between (or among) nested CLISTs and shows the order of CLIST invocation and execution. As each CLIST is invoked, it writes its name to the screen as described previously.

To activate level 1 tracing, enter TRACE1 on the OPTION line of an Information Center Facility selection panel. The selection panel is redisplayed with the command still on the OPTION line, and the following message is displayed:

```
CLIST tracing will be at level 1.  ICQGA006
```

TRACE2 Command — Level 2

Level 2 provides the same information as level 1. In addition, it displays, for all the nested CLISTs, each CLIST statement and TSO command and subcommand before execution. Each CLIST executes with the following options in effect:

```
CONTROL SYMLIST CONLIST LIST MSG
```

SYMLIST Display each executable statement before it is scanned for symbolic substitution. Executable statements include CLIST statements, and TSO commands and subcommands.

CONLIST Display CLIST statements after symbolic substitution, but before execution.

LIST Display TSO commands and subcommands after symbolic substitution, but before execution.

MSG Display informational messages from the CLIST.

To activate level 2 tracing, enter TRACE2 on the OPTION line of an Information Center Facility selection panel. The selection panel is redisplayed with the command still on the OPTION line, and the following message is displayed:

```
CLIST tracing will be at level 2.  ICQGC006
```

TRACE3 Command — Level 3

For all CLISTs, level 3 tracing provides the same trace as level 1. In addition, for a single explicitly-named CLIST, level 3 tracing provides the same trace as level 2. It displays each CLIST statement and TSO command and subcommand before execution. The explicitly-named CLIST executes with the options described for the level 2 trace.

To activate level 3 tracing, enter TRACE3.*clistname* on the OPTION line of an Information Center Facility selection panel, where *clistname* is the name of a nested CLIST. The selection panel is redisplayed with the command still on the OPTION line, and the following message displayed on it:

```
CLIST member clistname will be traced on the screen.  ICQGA005
```

Entering the TRACE3 command without specifying *clistname* turns off all tracing. The following message is displayed:

```
CLIST member name required.  TRACEOFF is set.  ICQGA004
```

TRACEOFF command

To deactivate tracing, enter TRACEOFF on the OPTION line of an Information Center Facility selection panel. The selection panel is redisplayed with the command still on the OPTION line, and the following message is displayed:

```
CLIST trace is turned off.  ICQGA003
```

If you enter TRACEOFF when tracing is not active, the following message is displayed:

```
CLIST trace is not active.  ICQGA002
```

Index

System Programming
Library: TSO Extensions
Installation and Planning
Volume 1

READER'S
COMMENT
FORM

SC28-1379-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity   Accuracy   Completeness   Organization   Coding   Retrieval   Legibility

If you wish a reply, give your name, company, mailing address, and date:

____

____

____

____

What is your occupation? ______________________________

How do you use this publication? ______________________________

Number of latest Newsletter associated with this publication: ____________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

System Programming Library: TSO Extensions Installation and Planning Volume 1

SC28-1379-2 S370-39

Reader's Comment Form

Cut or Fold Along Line

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department D58, Building 921-2
PO Box 390
Poughkeepsie, New York 12602

Fold and tape Please Do Not Staple Fold and tape

IBM®

Printed in U.S.A.

IBM /Technical Newsletter

This Newsletter No.	SN28-5066
Date	July 27, 1987
Base Publication No.	SC28-1379-2
File No.	S370-36
Prerequisite Newsletters/ Supplements	None

System Programming Library: TSO Extensions Installation and Planning Volume 1

This newsletter contains replacement pages for *SPL: TSO/E Installation and Planning* to update the Information Center Facility names directory.

Before inserting any of the attached pages into *SPL: TSO/E Installation and Planning*, read ***carefully*** the instructions on this cover. They indicate when and how you should insert pages.

Pages to be Removed	Attached Pages to be Inserted*
Cover - Edition Notice	Cover - Edition Notice
ix - x	ix - x
xv - xvi	xv - xvi
3-41 - 3-42	3-41 - 3-42.2
X-1 - X-6	X-1 - X-6

*If you are inserting pages from different Newsletters/Supplements and ***identical*** page numbers are involved, always use the page with the latest date (shown in the slug at the top of the page). The page with the latest date contains the most complete information.

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

This newsletter documents changes to the Information Center Facility names directory. This technical newsletter applies only when the PTF for APAR OY05782 is installed on your system.

Note: Please file this cover letter at the back of the publication to provide a record of changes.

**IBM Corporation, Information Development, Dept. D58, Building 921-2,
P.O. Box 390, Poughkeepsie, New York 12602**

Printed in U.S.A.

System Programming Library: TSO Extensions Installation and Planning Volume 1

SC28-1379-2 S370-39

Printed in U.S.A.