



PIE/TSO Dynamic Steplib Administrator Guide

Release 3.2.1

UNICOM
SYSTEMS, INC.

PTSOIG321-01

Printed: October 2002

©1984-2002 UNICOM Systems, Incorporated
All Rights Reserved

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without written permission from UNICOM Systems, Inc.

This manual applies to PIE/TSO release 3.2.1 and to all subsequent releases of the product until otherwise indicated by new editions or updates to this publication.

All products referenced are trademarks or registered trademarks of their respective companies.

UNICOM Systems, Inc.
Unicom Plaza - Suite 310
15535 San Fernando Mission Blvd.
Mission Hills, California 91345

Contents

Chapter 1 Overview	1
Changing STEPLIBs	1
Changing ISPLLIBs	2
Dynamic Concatenation	2
Stacking ISPF Applications	3
Technical Details	4
Release Notes.....	4
Chapter 2 Installation	5
Installation Without MultiTSO	5
Installation With MultiTSO	14
Chapter 3 Using Dynamic Steplib Commands	19
Dynamic STEPLIBs and ISPLLIBs	19
Resetting Your STEPLIB or ISPLLIB.....	21
Displaying Your Current Settings	22
Dynamic Concatenation.....	23
Adding Datasets to an Existing Allocation	23
Resetting Your Allocation.....	24
Displaying Your Current Allocation	24
Chapter 4 Messages	25
PTC Messages—Dynamic Concatenation	25
300 Messages	25
400 Messages	25
500 Messages	27
Return Codes	28
PTD Messages—Dynamic Steplib	29
000 Messages	29
100 Messages	31
200 Messages	32
300 Messages	32
400 Messages	33
500 Messages	35
Return Codes	36
Abend Completion Codes	36
Index	39



Chapter 1

Overview

Dynamic Steplib allows you to change library concatenations while you are logged on, using a simple command. You can change your

- STEPLIB
- ISPLLIB
- any current allocation

You can add datasets before or after your current concatenation. You can completely replace your STEPLIB or ISPLLIB concatenation. You can even eliminate your STEPLIB or ISPLLIB altogether.

Customizing STEPLIB and ISPLLIB concatenations can improve performance and reduce the administration required to install new applications and migrate to new versions.

Changing STEPLIBs

Improving Performance

You can use Dynamic Steplib to improve performance, both for your logon and for the system as a whole. Searching the STEPLIB is a resource intensive process. So it is critical to make the STEPLIB as small as possible. Yet you have scores of applications. You certainly don't want all of them in the LPA and link list. So, to be run, these applications have to be in the STEPLIB.

With TSO alone, these applications have to be added to the STEPLIB in the logon procedure. If you put the datasets for every application in the STEPLIB of the general logon procedure, it can take a long time and a lot of resources to finally find an application toward the bottom of the list.

You can alleviate this problem by creating a customized logon procedure for each user that includes only the datasets the user needs. But this is a maintenance headache most installations aren't willing to suffer.

That's where Dynamic Steplib comes in. With Dynamic Steplib, you can change, add to, or even eliminate your STEPLIB on the fly. Dynamic Steplib can maintain separate STEPLIBs in different ISPF split screens. It can even maintain separate STEPLIBs for each position in a stack of ISPF applications (see "Stacking ISPF Applications" on page 3). If you run PIE/TSO MultiTSO, you can have different STEPLIBs for each session. As you switch from session to session, application to application, Dynamic Steplib activates the appropriate STEPLIB for that environment.

This means, every application can have its own customized STEPLIB. Simply include Dynamic Steplib commands in the CLIST or REXX EXEC you use to start the application. At the start of the CLIST, you can replace the current STEPLIB with the datasets this application uses. Then, after the application is ended, the CLIST can reset the STEPLIB to its previous setting. If the modules the application uses are not in the STEPLIB, you can turn off the STEPLIB altogether. If you eliminate

the STEPLIB, MVS goes right to the LPA and link list and bypasses the overhead involved in a STEPLIB search. The time it takes to locate a module is drastically reduced.

Streamlining STEPLIBs improves performance for each user, because the search times are minimized. It also improves the performance of your whole system, because these small decreases compound to a substantial savings.

Facilitate Administration

Right now, with all your datasets allocated in the general logon procedure, one person is responsible for administering all dataset changes for all applications. Using Dynamic Steplib, you can allocate these datasets in individual CLISTs or REXX EXECs, and delegate dataset administration to the person responsible for maintaining each application.

Dynamic Steplib also simplifies testing. It allows you to test new versions and product PTFs without logging off and back on. Simply change the STEPLIB concatenation when you need to go from one version to another.

Changing ISPLLIBs

Dynamic Steplib provides the same benefits for ISPLLIB.

Of course, ISPF provides the LIBDEF command to provide a special “ISPLLIB” (ISPLUSR) for a particular user. However LIBDEF has an important restriction: Once an ISPF application is started, it is considered to be part of the standard ISPLLIB only. If the application tries to link to a module in the special ISPLLIB, it will abend with an 806-04. This means all applications that link to a module in their own library must be included in the ISPLLIB.

Dynamic Steplib overcomes this limitation by changing the ISPLLIB concatenation itself (instead of adding a ISPLUSR concatenation). There is only one library, so there is no difficulty when an application attempts to link to a module.

With Dynamic Steplib, you can add datasets before or after your current ISPLLIB concatenation, or you can completely replace the concatenation. An ISPLLIB change applies only to the ISPF split screen (or place in the stack) in which it is executed, so you can have a different ISPLLIB for every split screen. You can completely customize your ISPLLIB.

Dynamic Concatenation

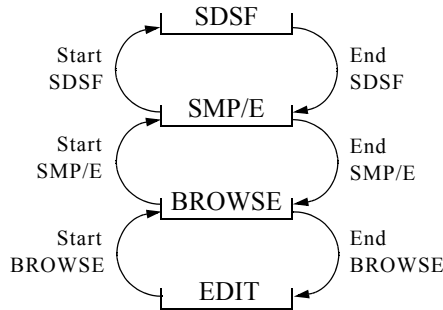
Dynamic concatenation allows you to add your own personal datasets to any existing allocation. You don't have to free and reallocate the datasets. You don't have to know the dataset names. You don't have to change existing CLISTs or EXECs when dataset names change with a new version.

All you have to do is specify the ddname you are changing and the datasets you want to add to the concatenation.

Using dynamic concatenation also offers benefits over using ALTLIB statements for SYSPROC and SYSEXEC and LIBDEF statements for ISPF datasets. Dynamic concatenation is easier to use. It is also more efficient. When you use ALTLIB and LIBDEFs, TSO must open and close two separate allocations. With LIBDEFs this is even more significant because ISPF must issue a BLDL twice each time it must locate an ISPxLIB/ISPxUSER allocation pair. When you use dynamic concatenation, there is only one open and one close, and ISPF needs to issue only one BLDL to locate an ISPx LIB/ISPxUSER allocation pair.

Stacking ISPF Applications

ISPF allows you to execute applications one on top of the other within a split screen. So you can execute, for example, EDIT, then BROWSE, then SMP/E, then SDSF, and so on. ISPF just places the new application right over the old. The old still exists in the background, and you can go back to it by exiting out of the applications “above” it in the stack.

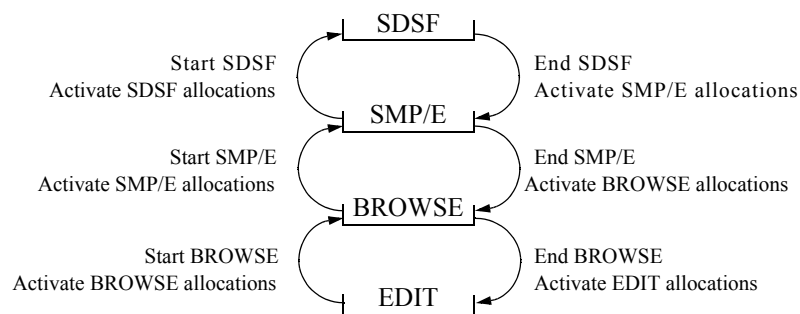


To do this, you must write a CLIST or REXX EXEC to execute the application you want to run next and start the application with the SELECT service. For example, you could create a CLIST that contains the following command:

```
ISPEXEC SELECT PGM(ISREDIT)
```

We call this type of execution stacking. We call the application you see on your screen the top of the stack. So in our example, above SDSF would be the top of the stack. If you ended it, SMP/E would be the top of the stack.

Dynamic Steplib allows you to have different allocations for each position in the stack. As you exit out of each application, going down the stack, Dynamic Steplib will invoke the allocations previously specified for that position. So if EDIT, BROWSE, SMP/E, and SDSF all have their own, customized allocations, they will still have their customized allocations going up and down a stack.



Technical Details

You do not have to change your logon procedure to use Dynamic Steplib. It works with any of the following TMPs:

- PCSEFT01 (MultiTSO),



- IKJEFT01 (IBM)
- ADFMDF03 (IBM's Session Manager)
- TSOSES (CMA's TSOPLUS)

Dynamic Steplib does not interfere with any existing tasklibs, for example ISPLLIB and those used by SAS.

Dynamic Steplib works with or without PIE/TSO MultiTSO. It will not affect the PIE/TSO job steps for PCSEFT01, PNAVL01, and PCSSTAT.

Dynamic Steplib requires:

- MVS version 3.1.0 or higher
- MVS DFP version 3.1 or higher
- RACF 1.9 or higher. ACF2 or TOPSECRET must support the RACROUTE macro specifying release 1.9

Release Notes

Version 1.2.0 has the following changes and enhancements.

- PTSSTEP has the new keyword AUTH. Authorized users can use the AUTH keyword to APF authorize the changed STEPLIB or ISPLLIB for their TSO address space.
- You may now specify volume overrides for PTSSTEP and PTSCAT datasets using the VOL parameter.
- PTSCAT RESET no longer resets the concatenation to the default. Now it functions like PTSSTEP and resets the concatenation to its last setting. To reset the concatenation to the default, use the new parameter STACK. You may also free the concatenation with the new FREE parameter.

To upgrade from version 1.1.0 to 1.2.0, follow all installation procedures in [“Chapter 2 Installation”](#) on page 5.

Chapter 2

Installation

Steps to install Dynamic Steplib differ depending on whether you also run PIE/TSO MultiTSO. If you do not run MultiTSO, follow the instructions in [“Installation Without MultiTSO,”](#) on page 5”. If you run MultiTSO, follow the instructions in [“Installation With MultiTSO,”](#) beginning on page 14.

Installation Without MultiTSO

Step 1: Load File 1

1. Unload the tape using the job below.

```
//LOADJCL JOB 1234,'load installation dataset',
//          MSGLEVEL=(1,1)
//LOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD VOL=SER=PTD120,
//          UNIT=3480,          <== Correct device type
//          DSN=TSCPIE.PTD120.CNTL,DISP=OLD,
//          LABEL=(1,SL,EXPDT=98000)
//SYSUT2 DD UNIT=SYSALLDA,DISP=(,CATLG,DELETE)
//          DSN=pie.PTD120.CNTL,<==Correct HLQ
//          VOL=SER=volser,    <==Supply the volume
//          SPACE=(6160,(60,60,4),,ROUND),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSIN DD DUMMY
//
```

Step 2: Load the Remaining Files

In this step you will build and run a job to load the rest of the files from the tape, using the `INSTALLS CLIST`. The `INSTALLS CLIST` will prompt you for job variables—job statements, qualifiers, etc. When you have completed the questions, the `INSTALLS CLIST` will build a customized installation job and present it for your review. You may then modify the job, if you like. It will be saved as member `SINSTJCL` in the `CNTL` dataset. The following sections detail the steps required to run the `INSTALLS CLIST`.

CHOOSING A LIBRARY

Dynamic Steplib modules must execute from an explicitly authorized APF library. That is, the library must have an entry in an active `IEAAPFxx` member of `SYS1.PARMLIB`. If it does not, you will receive an `S306-C` abend when you try to start Dynamic Steplib.

WARNING

Do not place the `PTSSTEP` module in a `STEPLIB`. If you use `PTSSTEP OVERRIDE`, the module will be “lost,” and you will be unable to issue another `PTSSTEP` command.

Install the `PTSSTEP` module in a link list library or an LPA library.


You may install all other Dynamic Steplib modules in a link list library or an LPA library, or you may install into a new library and authorize it. If the library is not in the link list or LPA, you will need a `STEPLIB DD` statement in your logon procedures. If you want Dynamic Steplib modules in the LPA, you will have to IPL with a `CLPA`.

RUNNING THE INSTALLS CLIST

If you like, complete the `INSTALLS` worksheet on the following page, before running the `INSTALLS CLIST`.

You may run the `CLIST` multiple times, if necessary. To cancel the `CLIST` enter `.END` at any prompt.

1. From ISPF Option 6 or TSO `READY` enter: `EXEC 'pie.PTD120.CNTL(INSTALLS)'`
Substitute the correct high level qualifier for `pie`.
2. Respond to the instructions on the screen.
3. Review the job and make any changes you like.
4. Submit your `SINSTJCL` job.

 If you use the Dynamic Steplib load library as the APF authorized library, the `COPYAPF` step will end with a condition code of 8. That is because `IEBCOPY` will think that a “member select” is being attempted during a compress. If this occurs the `IEBCOPY` will fail with a parameter validation error before doing the compress or the copy. In the above case, 8 is an acceptable return code for this job step.

INSTALLS CLIST WORKSHEET

The CLIST will prompt you for the following information. If you want, you can prepare your answers in advance with this worksheet.

What is the name of the dataset you loaded file 1 into (e.g., PIE.PTD120.CNTL)?

What APF library will you load Dynamic Steplib datasets into and what is its DISP (e.g., SYS3.PIEAUTH and SHR)?

What JES procedure library will you use for your Dynamic Steplib procedures and what is its DISP (e.g., SYS3.PROCLIB and SHR)?

What disk volume serial number will you use for Dynamic Steplib product datasets (e.g., MVS230)?

What esoteric unit name do you want to use for disk (e.g., SYSALLDA)?

What are the first and second level qualifiers for your Dynamic Steplib datasets (e.g., PIE.PTD120)?

If you are migrating or re-installing, do you want to overwrite old Dynamic Steplib datasets (YES or NO)?

What is the SYSOUT class (e.g., X)?

What esoteric unit name do you want to use for temporary disk datasets (e.g., SYSALLDA)?

What esoteric unit name do you want to use for tape units (e.g., 3480)?

What esoteric unit name do you want to use for VIO units (e.g., VIO)?

What job statement information do you want used (e.g., accounting information and programmer name)?

What special JES control statements do you want to add to the job stream (e.g., /* ROUTE statements, etc.)?

Step 3: Implement the Dynamic Steplib Library

1. Perform one of the following steps to implement your Dynamic Steplib load library.
 - If you loaded Dynamic Steplib into a link list library, perform an LLA refresh so that MVS can find the Dynamic Steplib load modules. Enter: F LLA,REFRESH
 - If you loaded Dynamic Steplib into an LPA library, IPL with a CLPA so that Dynamic Steplib load modules will be loaded into the LPA.
 - If you loaded Dynamic Steplib into another library, edit an IEAAPFxx member in SYS1.PARMLIB. Add an entry for the Dynamic Steplib load library. For example:
SYS3.DSAUTH MVS230. Then IPL.

Step 4: Modify the Dynamic Steplib Start-Up Procedure

You are now ready to modify the Dynamic Steplib start-up procedure.

1. Edit member PTDINIT from the Dynamic Steplib PROCLIB.

2. Modify the following parameters:

APF Identify the Dynamic Steplib load library.

DUMP Specify a SYSOUT class for dumps generated if there is a problem with PTDINIT. The default is A.

PARMDS Identify the dataset that will contain the Dynamic Steplib start-up options. The default is PIE.PTD120.PARMLIB.

SOUT Specify a SYSOUT class for PTDINIT output. The default is *.

Step 5: Set Start-Up Options

The STEPPARM member of the PARMLIB dataset contains Dynamic Steplib start-up options. These options specify your password, locate files, and tailor Dynamic Steplib processing.

The PTDINIT procedure locates and reads the STEPPARM member. Be sure to identify the correct dataset in PTDINIT's PARMDS parameter.

Syntax

Each statement in the STEPPARM member must be an 80 character statement, beginning in column one. Sequence numbers are not allowed.

To code a comment statement, place an asterisk (*) in column one. You may place a comment within a statement if you separate the parameter and value from the comments with one or more spaces.

You may not code continuations.

Updates

1. Edit your PARMLIB(STEPPARM) member.
2. Add EXPUSR=YES or NO. If you specify NO, Dynamic Steplib will suppress messages regarding invalid or expiring passwords from the user's terminal. These messages will still be displayed once a day on the console. If you specify YES, Dynamic Steplib will display these messages both at the user terminal and the console.
3. Add STPCMP=YES to indicate that you are using the Dynamic Steplib component of PIE/TSO.
4. Update the following start-up options using the information in the letter sent with your Dynamic Steplib product tape.

EXPDT Identify your Dynamic Steplib expiration date in yyyy.ddd format.

CPUID Each licensed CPU ID is paired with a password. Specify your CPU ID and password pairs. Code one pair per line. You may use as many lines as necessary. Example:

```
CPUID  
000000000000=XXXXXXXXXXXX  
111111111111=YYYYYYYYYYYY
```

CPUID must be the last option in the STEPPARM member.

Step 6: Authorize Commands

1. The PTSSTEP and PTSCAT commands are authorized commands, and you must add them to IKJTSOxx SYS1.PARMLIB. To do so, edit IKJTSOxx. Add PTSSTEP and PTSCAT to AUTHCMD and to AUTHTSF NAMES. Use the TSO PARMLIB command to update the table dynamically:

PARMLIB UPDATE(xx)

You may want to secure these commands using your external security system.

Step 7: Define Security Resources

To secure PTSSTEP or to use the PTSSTEP AUTH keyword, you must define PTSSTEP resources to your security system.

☞ The AUTH keyword APF authorizes the modified STEPLIB or ISPLLIB. Dynamic Steplib checks for authorization using a SAF call.

1. To secure PTSSTEP, define the resource PIEDYN.CMD.PTSSTEP in the FACILITY class. The access intent is READ.
2. To authorize the AUTH keyword, define the following resources in the FACILITY class. The access intent is UPDATE.
 - PIEDYN.CMD.PTSSTEP.AFTER.AUTHORIZE—AUTH keyword used with the AFTER keyword.
 - PIEDYN.CMD.PTSSTEP.BEFORE.AUTHORIZE—AUTH keyword used with the BEFORE keyword.
 - PIEDYN.CMD.PTSSTEP.OVERRIDE.AUTHORIZE—AUTH keyword used with the OVERRIDE keyword.

Step 8: Start Dynamic Steplib

1. Enter START PTDINIT from the console.

Installation With MultiTSO

Use this installation procedure if you also run PIE/TSO MultiTSO. If you are a new PIE/TSO customer, install MultiTSO before installing Dynamic Steplib.

Step 1: Installation

1. Unload the tape using the job below.

```
//LOADJCL JOB 1234,'load installation dataset',
//          MSGLEVEL=(1,1)
//LOAD     EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD VOL=SER=PTD120,
//          UNIT=3480,          <== Correct device type
//          DSN=TSCPIE.PTD120.CNTL,DISP=OLD,
//          LABEL=(1,SL,EXPDT=98000)
//SYSUT2   DD UNIT=SYSALLDA,DISP=(,CATLG,DELETE)
//          DSN=pie.PTD120.CNTL,<==Correct HLQ
//          VOL=SER=volser,    <==Supply the volume
//          SPACE=(6160,(60,60,4),,ROUND),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSIN    DD DUMMY
//
```

2. Edit your current PIE/TSO PCSINIT procedure. Add the new Dynamic Steplib load library to its STEPLIB concatenation, before the MultiTSO load library.

Step 2: Set Start-Up Options

The SESSIONS member of the PARMLIB dataset contains Dynamic Steplib start-up options. These options specify your password, locate files, and tailor Dynamic Steplib processing. You will share a SESSIONS member between MultiTSO and Dynamic Steplib.

SYNTAX

Each statement in the SESSIONS member must be an 80 character statement, beginning in column one. Sequence numbers are not allowed.

To code a comment statement, place an asterisk (*) in column one. You may place a comment within a statement if you separate the parameter and value from the comments with one or more spaces.

You may not code continuations.

UPDATES

1. Edit your current PARMLIB(SESSIONS) member.
2. Add EXPUSR=YES or NO. If you specify NO, Dynamic Steplib will suppress messages regarding invalid or expiring passwords from the user's terminal. These messages will still be displayed on the console. If you specify YES, Dynamic Steplib will display these messages both at the user terminal and the console.
3. Add STPCMP=YES to indicate that you are using the Dynamic Steplib component of PIE/TSO.
4. Update the following start-up options using the information in the letter sent with your PIE/TSO product tape.

EXPDT Identify your PIE/TSO expiration date in yyyy.ddd format.

CPUID Each licensed CPU ID is paired with a password. Specify your CPU ID and password pairs. Code one pair per line. You may use as many lines as necessary. Example:

```
CPUID  
000000000000=XXXXXXXXXXXX  
111111111111=YYYYYYYYYYYY
```

CPUID must be the last option in the SESSIONS dataset.

Step 3: Authorize Commands

1. The PTSSTEP and PTSCAT commands are authorized commands, and you must add them to IKJTSOxx SYS1.PARMLIB. To do so, edit IKJTSOxx. Add PTSSTEP and PTSCAT to AUTHCMD and to AUTHTSF NAMES. Use the TSO PARMLIB command to update the table dynamically:

PARMLIB UPDATE(xx)

You may want to secure these commands using your external security system.

Step 4: Define Security Resources

To secure PTSSTEP or to use the PTSSTEP AUTH keyword, you must define PTSSTEP resources to your security system.

☞ The AUTH keyword APF authorizes the modified STEPLIB or ISPLLIB. Dynamic Steplib checks for authorization using a SAF call.

1. To secure PTSSTEP, define the resource PIEDYN.CMD.PTSSTEP in the FACILITY class. The access intent is READ.
2. To authorize the AUTH keyword, define the following resources in the FACILITY class. The access intent is UPDATE.
 - PIEDYN.CMD.PTSSTEP.AFTER.AUTHORIZE—AUTH keyword used with the AFTER keyword.
 - PIEDYN.CMD.PTSSTEP.BEFORE.AUTHORIZE—AUTH keyword used with the BEFORE keyword.
 - PIEDYN.CMD.PTSSTEP.OVERRIDE.AUTHORIZE—AUTH keyword used with the OVERRIDE keyword.

Step 5: Start Dynamic Steplib

1. Enter `START PCSINIT ACT=NEWPARM` from the console.

Chapter 3


Using Dynamic Steplib Commands

Dynamic STEPLIBs and ISPLLIBs

To change a STEPLIB or ISPLLIB, issue the PTSSTEP command using the parameters below. If you are using ISPF, PTSSTEP will change the concatenation for the active split screen. If you are stacking, it will change the concatenation for the top of the stack. If you are using PIE/TSO MultiTSO, you can change the STEPLIB for any MultiTSO session. However you can change the ISPLLIB only for the session you are in.

If PTSSTEP is executed at TSO READY, the setting will remain in effect until the TSO session ends or you log off. If PTSSTEP is executed within ISPF, the setting will remain in effect until the ISPF split screen or stack position ends, abends, or restarts.

PTSSTEP	$\left[\begin{array}{c} \text{SESSION}(ids \text{ALL}) \\ \text{ISPLLIB} \end{array} \right]$
	$\left[\begin{array}{c} \text{BEFORE}[(\text{CURRENT})] \\ \text{AFTER}[(\text{CURRENT})] \\ \text{OVERRIDE} \end{array} \right]$
	DA(<i>dsn1, dsn2, ...</i>) [APFCK NOAPFCK] [AUTH] [QUIET] [VOL(<i>volser, volser, ...</i>)]

 The PTSSTEP command is a TSO command processor. You can execute it directly from TSO READY. If you enter it from the MultiTSO menu or using PIEEXEC, MultiTSO will start a new session and execute PTSSTEP from that session (if IMPTSO is set ON).

You may specify either SESSION or ISPLLIB. If you omit both, PTSSTEP will default to SESSION(*), that is, the current session.

ISPLLIB Indicates that you are changing the ISPLLIB.

SESSION Indicates that you are changing the regular STEPLIB and specifies the IDs of the MultiTSO sessions whose STEPLIBs you want to change. Separate multiple IDs with blanks or commas. These sessions must already be open. To change the STEPLIB for all open sessions, specify SESSION(ALL).

You may specify either BEFORE, AFTER, or OVERRIDE. BEFORE is the default.

BEFORE Concatenates the datasets you specify with the DA parameter in front of the datasets in the existing concatenation.

AFTER Concatenates the datasets you specify with the DA parameter following the datasets in the existing concatenation.

(CURRENT) If you specify CURRENT with BEFORE or AFTER, PTSSTEP will add the datasets to the last modification of the STEPLIB or ISPLLIB. If you omit CURRENT, it will restore the concatenation to its original datasets and then add the datasets you specify.

For example, your original concatenation is X Y Z. Using PTSSTEP BEFORE, you add dataset A and change the concatenation to A X Y Z. Then, using PTSSTEP BEFORE, you add dataset B; Dynamic Steplib reverts to the original concatenation (X Y Z) and adds your dataset B (B X Y Z). Then you add dataset C using BEFORE(CURRENT); Dynamic Steplib adds dataset C to the current concatenation (B X Y Z) to give you C B X Y Z.

OVERRIDE Replaces the existing concatenation with the datasets you specify with the DA parameter. If you use the OVERRIDE function, you may turn off the STEPLIB or ISPLLIB by specifying DA('NULLFILE').

DA Specify the new datasets with the DA parameter.

APFCK and NOAPFCK Determine whether Dynamic Steplib will check for APF authorization consistency among the datasets in the STEPLIB.

Remember that the ISPLLIB concatenation must contain only non-APF authorized datasets. Also, if any non-APF authorized dataset is placed in a STEPLIB concatenation, MVS will consider all datasets in the concatenation to be non-APF authorized.

It is easy to mix APF and non-APF authorized datasets in a concatenation by mistake, but this simple mistake can cause serious trouble. If APF check is on (the default), Dynamic Steplib will check all datasets in the new concatenation to ensure that they are either all APF authorized or all non-APF authorized. If the new concatenation mixes datasets that are APF authorized with datasets that are not authorized, Dynamic Steplib will prevent the change and issue an error message to your screen. If you want to force a change, specify NOAPFCK.

AUTH Forces APF authorization for the concatenation, for your TSO address space only. AUTH is available to authorized users only. Unauthorized users will receive an error message.

QUIET Suppresses error messages on the terminal, except for messages regarding invalid or expiring passwords. You can determine if there is an error by checking the return code in a CLIST or REXX EXEC.

VOL Specifies volume serial number overrides for datasets listed. If you supply a volser for a dataset, PTSSTEP will use the dataset on this volume, rather than using the information in the MVS catalog. Specify VOL values in the same order as DA values:

DA(A,B,C,D),VOL(A,B,C,D)

If there are fewer VOL values than DA values, the unpaired datasets will be accessed according to their MVS catalog entries:

DA(A,B,C,D),VOL(A,B)

To indicate the default volume in the middle of a VOL list, code an asterisk (*) in that position:

DA(A,B,C,D),VOL(A,B,*,D)

EXAMPLES

You will probably want to place the PTSSTEP command in the CLIST or REXX EXEC that starts an application. You can add any datasets needed by the application to the current concatenation. In this example, we change the ISPLLIB for DFSORT.

```
PTSSTEP BEFORE ISPLLIB DA('SYS1.ICEDGTLB')
ISPEXEC SELECT PGM(ICEISP)
```

If you are changing the STEPLIB for a non-ISPF application, you can override the current STEPLIB and add only the datasets the application needs. This gives you the most efficient STEPLIB possible. After the application has executed, reset the concatenation to its previous setting. In this example, we change the STEPLIB for SDSF.

```
PTSSTEP OVERRIDE DA('ISF.V1R3M3.ISFLOAD')
SDSF
PTSSTEP RESET
```

In this example, we change the STEPLIB in another session. We do not use the CURRENT option, so we are reverting to our original STEPLIB and adding this dataset to it.

```
PTSSTEP BEFORE DA('HLASM.HMQ4100.SASMMOD1') SESSION(4)
```

In this example, we change the ISPLLIB, adding the new datasets to the last modification of the ISPLLIB.

```
PTSSTEP BEFORE(CURRENT) DA('SYS4.FDR52.LOAD')
```

Resetting Your STEPLIB or ISPLLIB

To reset your library back to its defaults, use the following command format:

<pre>PTSSTEP RESET [SESSION(ids ALL)] [STACK] ISPLLIB</pre>

You may specify either SESSION or ISPLLIB. If you omit both parameters, Dynamic Steplib will reset the STEPLIB of the current session.

- SESSION** Resets the STEPLIB to its previous settings. Specify the IDs of the sessions you want this command to affect; it defaults to the current session. To reset the library for all sessions, specify SESSION(ALL). (You can always specify ALL, whether or not all of your session libraries have been changed previously.)
- ISPLLIB** Resets the ISPLLIB of the current session to its previous settings.
- STACK** Resets the entire stack. If you omit this parameter, only the top of the stack will be reset.

Displaying Your Current Settings

To display your current settings, use the following command.

```
PTSSTEP DISPLAY [ISPLLIB] [STACK]
```

ISPLLIB Displays the ISPLLIB concatenation. If you omit this parameter, it will display the STEPLIB concatenation.

STACK Displays the libraries for all applications in the stack.

Dynamic Concatenation

Changing an allocation affects all sessions. The allocation must not be open.

Adding Datasets to an Existing Allocation

To add datasets to the allocation, issue the PTSCAT command with one of the modification functions below. You can change a single allocation many times. PTSCAT will continue to add your changes to the last modification.

```
PTSCAT [ BEFORE ]
       [ AFTER ]
       FI(ddname)
       DA(dsn1, dsn2, ...)
       [VOL(volser, volser, ...)]
```

☞ The PTSCAT command is a TSO command processor. You can execute it directly from TSO READY. If you enter it from the MultiTSO menu or using PIEEXEC, MultiTSO will start a new session and execute PTSCAT from that session (if IMPTSO is set ON).

- | | |
|--------|--|
| BEFORE | The default, concatenates the datasets you specify with the DA parameter in front of the existing datasets in the allocation. |
| AFTER | Concatenates the datasets you specify with the DA parameter following the existing datasets in the allocation. |
| FI | Specify the ddname of the allocation you want to change. |
| DA | Specify the new datasets with the DA parameter. |
| VOL | Specifies volume serial number overrides for datasets listed. If you supply a volser for a dataset, PTSCAT will use the dataset on this volume, rather than using the information in the MVS catalog. Specify VOL values in the same order as DA values: |

DA(A, B, C, D), VOL(A, B, C, D)

If there are fewer VOL values than DA values, the unpaired datasets will be accessed according to their MVS catalog entries:

DA(A, B, C, D), VOL(A, B)

To indicate the default volume in the middle of a VOL list, code an asterisk (*) in that position:

DA(A, B, C, D), VOL(A, B, *, D)

Resetting Your Allocation

When you reset your allocation, all datasets added to the allocation with the PTSCAT command will be removed from the allocation and deallocated.

To reset your allocation, use the following command:

```
PTSCAT RESET FI(ddname) [STACK|FREE]
```

- | | |
|-------|---|
| FI | Specify the ddname of the allocation you want to reset. |
| STACK | Resets the entire stack. If you omit this parameter, only the top of the stack will be reset. |
| FREE | Frees the concatenation. No datasets will be allocated to it. |

Displaying Your Current Allocation

To display your current settings, use the following command.

```
PTSCAT DISPLAY FI(ddname)
```

- FI Specify the ddname of the allocation you want to display.

Chapter 4

Messages

PTC Messages—Dynamic Concatenation

300 Messages

PTC300I The current dynamic concatenation environment is:

ddname group name dataset name

This message is in response to a PTSCAT DISPLAY command.

Column 1 resents the allocation's ddname.

Column 2 presents the group names for the allocations. The original allocation's group name will be the same as the ddname. Groups added with the PTSCAT command will have incremental group names.

Column 3 presents the dataset names added to the allocation.

400 Messages

For the following messages PTSSTEP also sets a return code. The return code is the last two digits in the error message number.

PTC401E Error encountered during parse.

Contact UNICOM Systems, Inc. Customer Services.

PTC402E Acronym check for @PT@BLOK failed.

Contact UNICOM Systems, Inc. Customer Services.

PTC403E Acronym check for PTSCATB failed.

Contact UNICOM Systems, Inc. Customer Services.

PTC404E Dsname specified is too long.

Specify a dsname that is 44 characters or less.

PTC405E Too many dsnames specified. Limit total exceeded.

The total number of datasets in the allocation must be 64 or less. Eliminate unnecessary datasets.

PTC406E Dsnames must be specified.

The total number of datasets in the allocation must be 64 or less. Eliminate unnecessary datasets.

PTC407E Ddname specified is too long.

Specify a ddname that is eight characters or less.

PTC408E Ddname must be specified.

Specify a ddname using the FI(ddname) parameter.

PTC410E Dynamic allocation failed for the specified dsnames.

Verify that the datasets you specified with the DA parameter exist and are cataloged.

PTC411E Dynamic concatenation failed. Phase 1.

Verify that the datasets you specified with the DA parameter exist and are cataloged. If they are, contact UNICOM Systems, Inc. Customer Services.

PTC412E Dynamic concatenation failed. Phase 2.

Verify that the datasets you specified with the DA parameter exist and are cataloged. If they are, contact UNICOM Systems, Inc. Customer Services.

PTC413E PTSCAT internal logic error encountered.

Contact UNICOM Systems, Inc. Customer Services.

PTC414E The file specified for reset was not found.

Check the ddname you specified with the FI parameter. It must refer to an existing allocation.

PTC415E The file specified for reset is not a concatenation.

The ddname you specified with the FI parameter has not been changed using the PTSCAT command.

PTC416E The file specified for reset is OPEN.

First close the allocation. Then reissue PTSCAT RESET.

PTC417E Dynamic deconcatenation of specified file failed.

Contact UNICOM Systems, Inc. Customer Services.

PTC418E Unable to enq SYSZTIOT.

Try reissuing the PTSSCAT command. If this fails, contact UNICOM Systems, Inc. Customer Services.

PTC419E Unable to deq SYSZTIOT.

Contact UNICOM Systems, Inc. Customer Services.

PTC421E Incomplete concatenation performed. Duplicate DDNAME encountered.

Contact UNICOM Systems, Inc. Customer Services.

PTC422E Unable to perform deconcatenation. Duplicate DDNAME would result.

Contact UNICOM Systems, Inc. Customer Services.

PTC423E DDNAME not found.

Contact UNICOM Systems, Inc. Customer Services.

PTC424E Ddname specified is too long.

The ddname must be eight characters or less.

PTC425E Volser specified is too long.

You specified a volume serial number that is greater than six characters. Correct the statement and resubmit.

PTC426E Too many volsers specified. Limit total exceeded.

You have specified more than 64 volsers. The limit for dataset concatenation is 64 datasets. Correct the statement and resubmit.

PTC427E More volsers than dsnames were specified.

Correct the statement and resubmit.

PTC428E Unable to locate DSAB for a DD group.

We found a TIOT entry for the DD group, but there is no corresponding DSAB. Get an SVC dump and contact UNICOM Systems, Inc.. To resolve the problem, you may log off and back on.

PTC429E Dynamic concatenation failed. Phase 3.

You have issued PTSCAT RESET and Dynamic Concatenation was unable to reset the concatenation to its last setting. Dynamic Concatenation resets the ddname to the original concatenation. Get an SVC dump and contact UNICOM Systems, Inc..

500 Messages

PTC500W Your password for PTSCAT will expire in nn days.

Contact the person responsible for maintaining PIE/TSO Dynamic Steplib on your system.

PTC510E Your password for PTSCAT has expired. PTSCAT exiting.

No PTSCAT commands will be processed. Contact the person responsible for maintaining PIE/TSO Dynamic Steplib on your system and update the password.

Dynamic Steplib sets return code 500 with this message.

PTC520E PTSCAT is not licensed for this CPU. PTSCAT exiting.

No PTSCAT commands will be processed. Contact the person responsible for maintaining PIE/TSO Dynamic Steplib on your system and update the password.

Dynamic Steplib sets return code 500 with this message.

Return Codes

For an explanation of return code 500, see the PTSCAT 500 series messages. For other return code explanations, see the PTSCAT 400 series message that ends with the return code number. For example, for an explanation of return code 10, see PTSCAT message PTC410.

PTD Messages—Dynamic Steplib

000 Messages

PTD000I

PTDINIT was executed without a SYSPRINT DD statement. Message PTD013I will also be issued, as well as the headings, PTD010I, PTD011I and PTD12I. PTDINIT will list all parameter dataset statements on the console using this message identifier.

```
PTD010I DYNAMIC STEPLIB RELEASE rr.vv.mm PAGE nn
      PTD011I ON CPU cpuid model
      PTD012I AT hh:mm:ss ON yy.ddd
```

PTDINIT was executed without a SYSPRINT DD statement. Message PTD013I was already issued. These are the heading lines for the output listing.

PTD013I UNABLE TO OPEN SYSPRINT DATASET

The OPEN for the output dataset failed. PTDINIT will begin using the operator console to output all messages. Check the allocation for the SYSPRINT dataset.

PTD020E UNABLE TO OPEN SYSIN DATASET, TERMINATING

PTDINIT was unable to OPEN the parameter dataset. PTDINIT will terminate without initializing Dynamic Steplib. Check the allocation for the SYSIN dataset.

PTD021E NO DYNAMIC STEPLIB PARAMETERS FOUND, TERMINATING

The parameter dataset pointed to by the SYSIN DD statement was empty. PTDINIT will terminate without initializing Dynamic Steplib. Check the allocation for the SYSIN dataset.

PTD022E PIE/TSO DYNAMIC STEPLIB REQUIRED PARAMETERS MISSING

PTDINIT must find two valid parameter keywords in PARMLIB(STEPPARM). They are the CPU ID/password parameter (CPUID) and the expiration date parameter (EXPDT). One or more of the above parameters were missing from the parameter dataset. PTDINIT will terminate without initializing Dynamic Steplib. Supply all required parameters.

PTD023E PTDINIT TERMINATED DUE TO PARAMETER ERRORS

PTDINIT detected an error in the PARMLIB(STEPPARM) dataset. More information will appear in the SYSPRINT output dataset. PTDINIT will terminate without initializing Dynamic Steplib. Check the output from PTDINIT and correct any errors flagged.

PTD024E EXPIRATION DATE OF yy.ddd HAS EXPIRED, PIE TERMINATING

The license period for Dynamic Steplib has expired. Do not attempt to tamper with the EXPDT parameter in the parameter dataset. Any attempt to do so will result in Dynamic Steplib becoming unusable. PTDINIT will terminate without initializing Dynamic Steplib. Contact UNICOM Systems, Inc. to extend your license.

PTD025E PIE/TSO DYNAMIC STEPLIB NOT LICENSED ON THIS CPU

Dynamic Steplib has not been licensed on this CPU. Do not attempt to tamper with the CPUIDs in the parameter dataset. Any attempt to do so will result in Dynamic Steplib becoming unusable. PTDINIT will terminate without initializing Dynamic Steplib. If the primary CPU is unusable, alternate CPUs can be substituted on the license agreement, either temporarily or permanently. To do so, contact UNICOM Systems, Inc. to modify your license.

PTD026E PTDINIT EXECUTED AS A JOB OR TSO COMMAND

PTDINIT was executed as a batch job or from TSO. PTDINIT will terminate without initializing Dynamic Steplib. Do not attempt to execute PTDINIT except as a started task.

PTD027E PTDINIT MUST BE AN AUTHORIZED PROGRAM

PTDINIT was executed from a non-APF authorized library or has been link edited without the AC=1 parameter. PTDINIT will terminate without initializing Dynamic Steplib. Move PTDINIT to an APF authorized library and ensure that it has been link edited with APF authorization.

**PTD037E CSA OVERLAY DETECTED AT @SMPARMS ADDRESS hhhhhhhh CONTENTS
IS yyyyyyyyy**

PTDINIT has found that a Dynamic Steplib read-only control block, @SMPARMS at address hhhhhhhh, has been overlaid in CSA. PTDINIT will abend with a user completion code of 900. A storage overlay of CSA has probably occurred, corrupting a Dynamic Steplib read-only control block. Use the DUMP taken to check for a CSA overlay.

PTD040E UNABLE TO OPEN SYSLIB DATASET, TERMINATING

PTDINIT has found that the SYSLIB dataset required start Dynamic Steplib could not be opened. PTDINIT will terminate without initializing or replacing Dynamic Steplib. If APF was entered with the "S PTDINIT" command, ensure there was no typing error. If there was no error, check the SYSLIB DD statement in the PTDINIT procedure.

PTD041I PIE/TSO DYNAMIC STEPLIB rr.vv.mm SUCCESSFULLY INITIALIZED

PTDINIT has initialized Dynamic Steplib. rr.vv.mm is the release level. PTDINIT will terminate. Dynamic Steplib is initialized.

**PTD043E CSA OVERLAY DETECTED AT @SMPARMS ADDRESS hhhhhhhh CONTENTS
IS vvvvvvvv**

An attempt was made to start Dynamic Steplib while Dynamic Steplib was already initialized. At this time a CSA overlay of vvvvvvvv at address hhhhhhhh was detected in a Dynamic Steplib read-only control block. PTDINIT will abend with a user 900 completion code. Use the dump provided to check the CSA overlay.

PTD045E SQA/CSA CRITICAL, REQUEST ABORTED

An attempt was made to start Dynamic Steplib. There was not enough SQA or CSA available. PTDINIT will terminate with a user 120 abend. Correct the CSA or SQA shortage problem.

**PTD046E LOAD UNSUCCESSFUL REASON CODE n - PIE/TSO DYNAMIC STEPLIB
TERMINATING**

An internal error occurred attempting to load Dynamic Steplib modules during PTDINIT execution. PTDINIT will terminate with a user 220 abend. Save the dump provided and contact UNICOM Systems, Inc..

PTD047E CVT MODIFIED - PIE/TSO DYNAMIC STEPLIB TERMINATING

The MVS CVT was modified during PTDINIT execution. PTDINIT will terminate with a user 600 abend. Save the dump provided and contact UNICOM Systems, Inc..

PTD048E CVT MODIFIED DURING LOAD - PIE/TSO DYNAMIC STEPLIB TERMINATING

The MVS CVT was modified during PTDINIT execution. PTDINIT will terminate with a user 690 abend. Save the dump provided and contact UNICOM Systems, Inc..

PTD050E PTDINIT INTERNAL ERROR ON ccccccc PIE/TSO DYNAMIC STEPLIB TERMINATING

An internal error was detected in PTDINIT. ccccccc is the internal error designation. PTDINIT will terminate with a user 710 abend. Save the dump provided and contact UNICOM Systems, Inc..

PTD051E CAN'T FIND ccccccc MODULE

A required module, ccccccc, was not in the dataset pointed to by the PTDINIT procedure SYSLIB DD statement during a PTDINIT execution. PTDINIT will terminate without initializing Dynamic Steplib. If APF was entered with the "S PTDINIT" command ensure there was no typing error. If there was no error, check the validity of the SYSLIB DD statement.

PTD052E BLDL ERROR ACCESSING MODULE ccccccc RETURN CODE n REASON CODE y

An error occurred during a BLDL for module ccccccc. The return code n and reason code y are provided. PTDINIT will terminate without initializing Dynamic Steplib. Check the validity of the SYSLIB DD statement. The BLDL return codes and reason codes can be found in the IBM Data Management Services Guide.

PTD054I @SPARM LOADED AT X'address'

This message indicates where @SPARM has been loaded in global storage. It is issued after first time Dynamic Steplib activation.

PTD055I @SPARMX LOADED AT X'address'

This message indicates where @SPARMX has been loaded in global storage. It is issued after first time Dynamic Steplib activation.

100 Messages

PTD120I END OF PARAMETERS

This message will appear only on the SYSPRINT dataset. The end of the parameter dataset has been reached.

PTD121W EXPIRATION DATE OF yy.ddd WILL EXPIRE IN n DAYS

The Dynamic Steplib license agreement will expire on yy.ddd. Do not attempt to modify the EXPDT parameter or Dynamic Steplib will become unusable. Contact UNICOM Systems, Inc. for a license extension.

200 Messages

PTD220E UNKNOWN KEYWORD

This message will appear only on the SYSPRINT dataset. The previous statement contained a keyword not known to PTDINIT. PTDINIT will continue to process input parameters, and then terminate. Correct or delete the statement in error.

PTD221E *

This message will appear only on the SYSPRINT dataset. The previous statement contained an error. The asterisk (*) flags the column where the error was detected. PTDINIT will continue to process input parameters, and then terminate. Correct or delete the statement in error.

PTD222E DUPLICATE PARAMETER, IGNORED

This message will appear only on the SYSPRINT dataset. The previous statement contained a keyword already processed from the parameter dataset. It will be ignored. PTDINIT will continue to process input parameters, and then terminate. Correct or delete the statement in error.

PTD227E cccc INVALID EXPIRATION DATE FORMAT

This message will appear only on the SYSPRINT dataset. The EXPDT parameter was not in the format yy.ddd. Do not tamper with the EXPDT parameter or Dynamic Steplib will become unusable. PTDINIT will continue to process input parameters, and then terminate. Correct or delete the statement in error.

PTD234E INVALID INPUT CARD

This message will appear only on the SYSPRINT dataset. An input statement did not begin with a keyword, asterisk, or CPU serial number. PTDINIT will continue to process input parameters, and then terminate. Correct or delete the statement in error.

300 Messages

PTD300I The current dynamic STEPLIB environment is:

PTD300I Session nn: dsnames

This message is in response to a PTSSTEP DISPLAY command. The dsnames are the datasets you have added to your concatenation using the PTSSTEP command. Session nn identifies your current session. If you specify PTSSTEP DISPLAY STACK, each new position in the stack will be identified with another Session nn.

PTD310E * ERROR * - Dsnames not found.

This message is in response to a PTSSTEP DISPLAY command. Contact UNICOM Systems, Inc. Customer Services.

PTD350I The dynamic ISPLLIB environment for the current split screen is:

PTD351I module dsnames

This message is in response to a PTSSTEP DISPLAY ISPLLIB command. The dsnames are the datasets you have added to your ISPLLIB concatenation using the PTSSTEP command. Module is the name of the active module. For instance, ISPTASK is the module name for some ISPF menus. If you specify PTSSTEP DISPLAY STACK, each new element in the stack will be identified with another module name.

PTD360E * ERROR * - Dsnames not found.

This message is in response to a PTSSTEP DISPLAY ISPLLIB command. Contact UNICOM Systems, Inc. Customer Services.

400 Messages

For the following messages PTSSTEP also sets a return code. The return code is the last two digits in the error message number.

PTD401E Error encountered during parse.
Contact UNICOM Systems, Inc. Customer Services.

PTD402E Acronym check for @PT@BLOK failed.
Contact UNICOM Systems, Inc. Customer Services.

PTD403E Acronym check for PTSSTEPB failed.
Contact UNICOM Systems, Inc. Customer Services.

PTD404E DSNAME specified is too long.
Specify a dsname that is 44 characters or less.

PTD405E Too many dsnames specified. Limit total exceeded.
The total number of datasets in the STEPLIB concatenation must be 64 or less. Eliminate unnecessary datasets.

PTD406E Dsnames must be specified.
Specify dataset names using the DA(dsname,dsname,...) parameter.

PTD407E Session ID specified is too long.
Session IDs must be eight characters or less.

PTD408E Too many session IDs specified.
Only specify open sessions. You cannot specify more than twelve sessions.

PTD409E Invalid session ID specified.
Only specify open sessions. Check the session ID.

PTD410E Dynamic Steplib stack full.
You already have 384 dynamic steplibs active. Reset a STEPLIB or close an active application to create a new dynamic steplib.

PTD411E Dynamic allocation failed for the STEPLIB. Phase 1.
Verify that the datasets you specified with the DA parameter exist and are cataloged. If they are, contact UNICOM Systems, Inc. Customer Services.

PTD412E A STEPLIB specified was not a PDS.
All datasets must be PDS datasets. Re-enter the command with valid datasets only.

PTD413E Dynamic allocation failed for the STEPLIB. Phase 2.

Verify that the datasets you specified with the DA parameter exist and are cataloged. If they are, contact UNICOM Systems, Inc. Customer Services.

PTD414E Dynamic concatenation failed for the STEPLIB.

This message may indicate that there is not enough space in the TIOT. Try freeing some allocated datasets that you do not need. If you still get this message, contact UNICOM Systems, Inc. Customer Services.

PTD415E Open failed for the STEPLIB.

This message may indicate that you are not authorized to read one of the datasets in the new STEPLIB. Ensure that you have security access to all datasets in the concatenation. If you do, contact UNICOM Systems, Inc. Customer Services.

PTD416E Logic error encountered for action BEFORE or AFTER.

Contact UNICOM Systems, Inc. Customer Services.

PTD417E Scan for ddname in TIOT failed - Not Found.

Contact UNICOM Systems, Inc. Customer Services.

PTD418E Dsname table limit exceeded. No more slots.

The total number of datasets in the STEPLIB concatenation must be 64 or less. Eliminate unnecessary datasets.

PTD419E Unable to enq SYSZTIOT.

Try reissuing the PTSSTEP command. If this fails, contact UNICOM Systems, Inc. Customer Services.

PTD420E Unable to deq SYSZITOT.

Try reissuing the PTSSTEP command. If this fails, contact UNICOM Systems, Inc. Customer Services.

PTD421E Concatenation of APF and non-APF authorized libraries prevented.

Dynamic Steplib checks to ensure that there is no mixture of APF authorized and non-APF authorized datasets in the concatenation. If you want to force a change that would create a mixture, use the NOAPFCK parameter.

 **WARNING**

If there is a mixture of APF authorized and non-APF authorized datasets in a concatenation, MVS will consider all datasets in the concatenation non-APF authorized. Use the NOAPFCK parameter advisedly.

PTD422E ISPLLIB option specified outside of an ISPF environment.

You must execute PTSSTEP ISPLIB within ISPF. If you get this message from within ISPF, contact UNICOM Systems, Inc. Customer Services.

PTD423E ISPLLIB option logic error.

Contact UNICOM Systems, Inc. Customer Services.

PTD424E Environmental conflict detected. ISPF steplib swapping disabled.

Contact UNICOM Systems, Inc. Customer Services.

PTD425E ISPLLIB table limit exceeded. No more slots.

The total number of datasets in the ISPLLIB concatenation must be 48 or less. Eliminate unnecessary datasets.

PTD426E ISPLLIB TCBJLB mismatch error.

Contact UNICOM Systems, Inc. Customer Services.

PTS427E You may only specify active sessions.

Only specify open sessions. Check the session ID.

PTD428E You have insufficient access authority to execute PTSSTEP.

You are not authorized to use PTSSTEP. To obtain authorization, contact your security administrator.

PTD429E You have insufficient access authority to use the AUTH keyword.

You are not authorized to use the AUTH keyword of PTSSTEP. To obtain authorization, contact your security administrator.

PTD430E Volsr specified is too long.

You specified a volume serial number that is greater than six characters. Correct the statement and resubmit.

PTD431E Too many volsers specified. Limit total exceeded.

You have specified more than 64 volsers. The limit for dataset concatenation is 64 datasets. Correct the statement and resubmit.

PTD432E More volsers than dsnames were specified.

Correct the statement and resubmit.

500 Messages

These messages will be displayed on the user's terminal and on the operator console. The message is sent to the console only once per day. All 500 messages are non-scrollable. They are sent to the console using a WTO with route code 2 and descriptor code 11.

These messages are suppressed from the user's terminal if you specify start-up option EXPUSER=NO.

PTD500W Your password for PTSSTEP will expire in xx days.

Contact the person responsible for maintaining PIE/TSO Dynamic Steplib on your system.

PTD510E Your password for PTSSTEP has expired. PTSSTEP exiting.

No PTSSTEP commands will be processed. Contact the person responsible for maintaining PIE/TSO Dynamic Steplib on your system and update the password.

Dynamic Steplib sets return code 500 with this message.

PTD520E PTSSTEP is not licensed for this CPU. PTSSTEP exiting.

No PTSSTEP commands will be processed. Contact the person responsible for maintaining PIE/TSO Dynamic Steplib on your system and update the password.

Dynamic Steplib sets return code 500 with this message.

Return Codes

For an explanation of return code 500, see the PTSSTEP 500 series messages. For other return code explanations, see the PTSSTEP 400 series message that ends with the return code number. For example, for an explanation of return code 10, see PTSSTEP message PTS410.

Abend Completion Codes

The codes listed below are all user abend codes issued by PTDINIT.

120

PTDINIT was not able to GETMAIN the CSA or SQA required to initialize Dynamic Steplib. This abend is always preceded by message PTD045E. If SQA was specified, the amount requested was less than 128 bytes.

If CSA was specified, the amount requested was less than 4K.

Check to see if there are any problems with CSA or SQA. If not save the dump provided, and contact UNICOM Systems, Inc..

220

An internal error occurred attempting to load Dynamic Steplib modules during PTDINIT. This abend is always preceded by message PTD046E.

Save the dump provided, and contact UNICOM Systems, Inc..

600

The CVT was dynamically modified before the execution of PTDINIT. This abend is always preceded by message PTD047E.

Save the dump provided, and contact UNICOM Systems, Inc..

690

The CVT was dynamically modified before the execution of PTDINIT. This abend is always preceded by message PTD048E.

Save the dump provided, and contact UNICOM Systems, Inc..

710

An internal error occurred during PTDINIT. This abend is always preceded by message PTD050E.

Save the dump provided, and contact UNICOM Systems, Inc..

900

A Dynamic Steplib read only control block in CSA has been overlaid. This abend will always be preceded by a PTD037E message, or a PTD042E message and a PTD043E message.

Check for a probable CSA overlay. Save the dump, and contact UNICOM Systems, Inc..



Index

A

- Abend completion codes
 - PCSINIT 36
- APF authorized library
 - installation CLIST 6
 - loading PIE/TSO datasets 6
- AUTHCMD and the ACCESS command 11, 16
- Authorized APF library
 - installation CLIST 6
 - loading PIE/TSO datasets 6
- AUTHTSF NAMES and the ACCESS command 11, 16

C

- Completion codes
 - PCSINIT 36
- Condition code 8, PCSDFLD@ 6
- CPUIDS start-up option 10, 15
- Customization
 - PIE/TSO processing 10, 15

E

- EXPDT start-up option 10, 15

I

- IEAAPFxx member 6
- IEBCOPY
 - and the installation job 6
- IKJTSOxx, updating 11, 16
- Initialization procedure 9
- Installation
 - installation job, creating automatically 6
 - installation job, creating manually 6
 - IPL 8
 - start up procedure 9
- INSTALLS CLIST 6
 - and the load library 6
 - running 6

- worksheet 7
- IPL 8

L

- Link list 6
- LLA refresh 8
- Load library
 - implementing 8
 - installation CLIST 6
- LPA library 8

M

- Modifying the PIE/TSO start-up procedure 9

P

- Passwords
 - PIE/TSO 10, 15
- PCS000I 29
- PCS010I 29
- PCS013I 29
- PCS020E 29
- PCS021E 29
- PCS022E 29
- PCS023E 29
- PCS024E 29
- PCS025E 30
- PCS026E 30
- PCS027E 30
- PCS040E 30
- PCS041I 30
- PCS043E 30
- PCS045E 30
- PCS046E 30
- PCS047E 31
- PCS048E 31
- PCS050E 31
- PCS051E 31
- PCS052E 31
- PCS054I 31
- PCS055I 31
- PCS120I 31
- PCS121W 31
- PCS220E 32
- PCS221E 32
- PCS222E 32
- PCS227E 32
- PCSINIT messages
 - abend completion codes 36

- PCSINIT procedure 9
 - and start-up options 10
- PIE/TSO
 - library
 - implementing 8
 - installation CLIST 6
 - start-up options 10, 15
- PROCLIB (PCSINIT) 9

R

- Refresh LLA 8
- Running the INSTALLS CLIST 6

S

- S306-C abends 6
- SESSIONS member 10, 15
- SINSTJCL installation job output 6
- Start-up options 10, 15
 - syntax 10, 15
- Start-up procedure, PCSINIT 9
- Syntax
 - start-up options 10, 15
- SYS1.PARMLIB
 - IEAAPFxx member 6
 - IKJTSoxx member 11, 16

T

- Tailoring PIE/TSO processing 10, 15
- Troubleshooting
 - S306-C abends 6
- TSC PCS026E 30

U

- User abend codes
 - PCSINIT 36

W

- Worksheet, INSTALLS CLIST 7