

TASKLIB

Manage STEPLIB, LIBDEF and task libraries for TSO/ISPF



■ **Dramatically improve performance of the TSO LOGON procedure**

■ **Use one TSO LOGON procedure for all TSO users**

■ **Dynamically reconfigure STEPLIB while in TSO or ISPF**

■ **Seamless support for the ISPF LIBDEF ISPLLIB service**

■ **Support for stacking task libraries and recursive invocation**

■ **Comprehensive display of TSO/ISPF tasks**

What is TASKLIB?

TASKLIB is a TSO command processor that lets you dynamically reconfigure STEPLIB or task library concatenations in the TSO/ISPF environment.

While the TSOLIB command is only available at the TSO READY prompt, TASKLIB lets you change STEPLIB and task library concatenations while executing in TSO or ISPF.

TASKLIB uses the familiar syntax of the TSOLIB and ALTLIB commands.

The TASKLIB ACTIVATE command establishes a new task library at the task level you specify. These levels can be TSO subtasks such as IKJEFT01, IKJEFT02, IKJEFT09 or the ISPF logical screen task ISPTASK. Every ACTIVATE request saves the current library concatenation on the stack until a DEACTIVATE command makes the previous search order active again.

Systems programmers can issue the TASKLIB APF command to dynamically APF authorize their task library in order to test authorized commands.

The TASKLIB ISPF command lets you execute ISPF applications that issue ATTACH, LINK, LOAD and XCTL macros without the need to specify a private load library. This eliminates the S806 abends that are otherwise inevitable.

Your ISPF applications can now effectively manage their DB2 library allocations within ISPF — without re-LOGON or returning to the TSO READY prompt.

Benefits

- Dramatically reduce CPU utilization and EXCP counts of TSO/ISPF logon processing — in some cases by as much as 90% or more.
- Improve end-user response time and satisfaction to achieve better productivity.
- Use one very simple LOGON procedure for all TSO/ISPF users to bring order and uniformity to your ISPF application environment.
- Achieve granular ISPF application dataset allocations to enable use of RACF to restrict application access to authorized users only and improve overall system security.
- Make ill-behaved ISPF applications well-behaved, seamlessly and with little extra effort.
- Installation of new ISPF applications becomes simple and uniform to simplify product installation and maintenance.
- Eliminate superfluous dataset ENQs that interfere with library maintenance.

Benefits *(continued)*

- Using the DISPLAY TASKS command, you can immediately see the amount of storage required for resident programs and data areas. This helps you plan the region size of the TSO LOGON procedure to improve overall system performance.

Features

- ACTIVATE, DEACTIVATE and DISPLAY commands
- Task library stacking on STEPLIB and ISPTASK levels, for every ISPF logical screen
- Dynamic APF authorization of task libraries by authorized systems programmers
- Concatenate new task libraries BEFORE or AFTER an existing allocation or REPLACE the allocation altogether
- Support for the ISPF LIBDEF ISPLLIB service
- Eliminate duplicate libraries from dataset concatenations to further reduce search times for commands and programs

Prerequisites

TASKLIB runs with all releases of z/OS, OS/390 and TSO/ISPF whose IBM support is current.

Bring order and speed to your TSO/ISPF environment

Contact the TSO/ISPF experts at 800 776-0771 www.relarc.com

DISPLAY TASKS command output

Figure 1 shows sample output from a TASKLIB DISPLAY TASKS command.

```

TLB021I - Library = TLB00005 (active ) LEVEL(1) BEFORE  TCB=8FF6D0
TLB034I - DSNNAME = RAI.RAI100.LOAD
TLB034I - DSNNAME = RLX.V721.RLXLOAD
TLB034I - DSNNAME = DSN7.SDSNEXIT
TLB034I - DSNNAME = DSN7.SDSNLOAD
TLB018I - Library = TLB00010 (active ) LEVEL(4) BEFORE  TLB00005 TCB=8DE2A0
TLB034I - DSNNAME = RTE.V314.RTELOAD
TLB034I - DSNNAME = RAI.RAI100.LOAD
TLB034I - DSNNAME = RLX.V721.RLXLOAD
TLB034I - DSNNAME = DSN6.SDSNEXIT
TLB034I - DSNNAME = DSN6.SDSNLOAD
-----
TCB Address          Program  IC STAB FL CD Tasklib      Data Program
-----
8FF6D0              *JS*IKJEFT01 01 0083 42 00 TLB00005      40K  88K
8F9E88              IKJEFT02 01 0083 46 00 TLB00005        0K   0K
8F9CF0              IKJEFT09 01 0083 42 00 TLB00005        4K   0K
8F9A20              EXEC      6D 0082 00 00 TLB00005        8K   0K
                   04ADA6C0 01 0003 42 14
8DE540              ISPF      06 0082 00 00 TLB00005     212K   0K
                   ISPMAIN  01 0003 42 00
890AA0              ISPTASK   06 0082 00 00 TLB00005     96K   0K
                   ISFMAIN  0C 0002 02 00
                   0A241DEE 01 0003 40 14
86DD90              EXEC      6D 0082 00 00 TLB00005        8K   0K
                   04ADA6C0 01 0003 42 14
86DA78              >TASKLIB  6D 0082 00 00 TLB00005     40K   0K
890810              EXEC      6D 0082 00 00 TLB00005        4K   0K
                   04ADA6C0 6D 0002 02 14
                   04ADAFAA 01 0002 42 14
8DE2A0              ISPTASK   01 0082 40 00 TLB00010     120K   0K
893E88              EXEC      6D 0082 00 00 TLB00010        4K   0K
                   04ADA6C0 6D 0002 02 14
                   04ADAFAA 01 0002 42 14
8A2760              RLXS      06 0082 00 00 ISP09573     712K   0K
                   RLXM     01 0003 42 00
8934A8              CAFD      01 0083 42 00 ISP09573        0K   0K
893170              RLXE      01 0082 40 00 ISP09573     12K   0K
896680              RLX       01 0082 40 00 ISP09573        0K   0K
896A78              RLXT      01 0083 42 00 ISP09573        0K   0K
896D90              RFPM      01 0082 42 00 ISP09573        0K   0K
8A29F0              EXEC      6D 0082 00 00 TLB00010        4K   0K
                   04ADA6C0 01 0002 42 14
8A2D08              TASKLIB   01 0082 40 00 ISP09573     40K   0K
8F90C8              EXEC      6D 0082 00 00 TLB00010        4K   0K
                   04ADA6C0 6D 0002 02 14
                   04ADAFAA 01 0002 42 14
-----

```

Figure 1: Output of the TASKLIB DISPLAY TASKS command

Figure 1 shows the subtasks in a split screen environment with two ISPF applications active. The task library with DDNAME TLB00005 is installed in all TSO subtasks and is propagated to all subtasks without their own task library. The DB2 V7 load libraries are base (or default) libraries for all DB2 applications.

In the ISPTASK with TCB address 890AA0, the SDSF application is active. Since the SDSF load library is defined in LINKLIST, no special task libraries need to be installed for it.

A DB2 application is active in the ISPTASK with TCB address 8DE2A0. This application requires a DB2 V6 load library that is installed under DDNAME TLB00010.

You can readily see data and program memory requirements for this TSO/ISPF session — it is the sum of all values shown in the Data and Program columns.