Batch Installation and Operation

This chapter covers the following topics:

- CALLDLI Interface
- Consistency Interface
- User Exit
- z/OS JCL Requirements
- z/VSE JCS Requirements

CALLDLI Interface

The ADL CALLDLI Interface processes all data base requests originating from DL/I calls or command level programs. For the application, it provides an interface identical to original DL/I. The CALLDLI Interface decides dynamically, whether a request is to be routed to DL/I or to be processed by ADL. In the first case, it simply acts as a front-end to DL/I.

ADL may be executed in batch in two different modes:

- **Normal Mode**
  This is used for application programs using PSBs which only reference DBDs describing previously converted DL/I data bases.

- **Mixed Mode**
  This is used for application programs using PSBs which reference at least one DBD describing an unconverted DL/I data base.

In normal mode, all requests are processed by ADL. In mixed mode, requests are routed to DL/I if they refer to non-converted DL/I data bases. GU and ISRT calls on the IOPCB and CHKP calls will be routed to ADL as well as to DL/I.

Positional Parameters for DAZIFP

All DL/I parameters will be ignored, with the exception of the positional parameters given below. These will be accepted and interpreted by DAZIFP in the same way as DL/I.

The syntax for the input statement is as follows:

```
runmode,pgmname,psbname[,keyword]
```

These parameters are explained on the next page.

Parameter	Description
runmode	A three-character string indicating the run mode. The possible values are described in the following table below.
pgmname	The name of the application program to be loaded and started.
psbname	The name of the PSB used by the application program (if omitted, it is assumed that the PSB name is the same as the application program name).

Run Mode	Description
BMP/MPS	Indicates a multi-user batch run. Programs using this mode of operation can run concurrently with other programs (batch or online). An IO-PCB is added in front of the PCBs. Automatic ET calls may be issued by the ADL. Checkpoints may be issued and will result in Adabas ET calls. See also the section *Recovery and Restart Procedures*. Compatible with DL/I BMP/MPS.
ELO	Indicates an Establish Logical Relationship utility run.
IMS/DLI	Indicates a normal run. Batch update programs using this mode of operation cannot run concurrently with any other program (batch or online) using the same data base(s). Read-only programs can. Checkpoints may be issued and will result in Adabas unsynchronized checkpoints being taken (C1 calls). See also the section *Recovery and Restart Procedures*. Compatible to DL/I IMS/DLI.
MIX	Indicates that the batch program is to be run in mixed mode. The same rules apply as for IMS/DLI mode.
MPX/SDX	Indicates an MPS/SDB batch run in mixed mode.
MSG	Indicates an IMS/TP message region processing run.
PRE	Indicates an ADL precompiler run for translating command level programs.
PRT	Indicates a Print Trace utility run.
REF	Indicates a Reformat utility run.
SDB	Indicates a shared-data base run. Programs running in this mode may access data bases concurrently with other programs (batch and online). This mode of operation is provided for compatibility with the DL/I "shared data bases" feature. The ET checkpoint logic is like BMP/MPS, but no IO-PCB is added to the PCBs.
SHI	Indicates a DAZSHINE utility run.
STA	Indicates a stand-alone batch run. Programs using this mode of operation can not run concurrently with online programs using the same data base(s), nor with any other batch programs. The ADL directory file is used exclusively. Checkpoints may be issued and will result in Adabas unsynchronized checkpoints being taken (C1 calls). See also the section Recovery and Restart Procedures.
UNL	Indicates an Unload utility run.
UTC	Indicates a CBC utility run.

Keywords for DAZIFP

In addition to the positional parameters mentioned above, you may specify one or more keywords to control operation of ADL.

Any keywords specified must immediately follow the positional parameters. Any keywords not on the list (e.g. the original DL/I parameters) encountered after the program and PSB names will cause scanning to be terminated. In this case, all other parameters will be ignored.

If the parameters are read in from a file (i.e., DAZIN1, DAZIN2 or SYSIPT), more than one line can be used for the parameter specification. A comma followed by a blank after the last keyword of the line indicates that the specification of the parameters is continued in the next line. Any character found after the blank in the initial line is treated as a comment.

The table below gives a short explanation of the keywords which may be specified. For further information (including defaults) on all parameters, see the section *ADL Parameter Module* in the *ADL Installation* documentation.

Keyword	Description
BUFSF	Suffix for the buffer table module name.
CHKPMSG	Checkpoint message destination.
CPID	The checkpoint ID from which the application is to be restarted.
DBD	The size (in kilobytes) of the ADL DBD ICB buffer.
DBDSF	Suffix for the DBD table module name.
DBID	Adabas data base for the ADL directory file.
DUO	Support of CA-DUO under z/OS.
EBUF	The size (in kilobytes) of the ADL ECB buffer.
ET	The number of times a different root segment occurrence may be accessed before an Adabas ET command is issued (BMP/MPS/SDB programs only).
FNR	Adabas file number for the ADL directory file.
FX	z/VSE only. Specifies the input data set for the Print Utility when printing the routine trace or for the ADL precompiler.
IMSY	IMS/TP syncpoint/Adabas ET synchronization.
LANG	Language of the application program. Overwrites the PSB language parameter.
LCS	The size (in kilobytes) of the last call save area.
LOAD	Indicates whether an ISRT against a PCB with PROCOPT=L is to be written to Adabas or to a sequential file.
MFT	Defines the ADL Multifetch Table.
OPENRQ	Specifies if an Adabas OPEN is required or not.
PLI	z/OS only. Passes PL/I parameters to PL/I. See the explanation on the next page.
PLILE	z/VSE only. Specifies whether PL/I uses the LE/VSE.

Keyword	Description
PR	z/VSE only. Specifies the number of logical printers.
PSB	The size (in kilobytes) of the ADL PSB ICB buffer.
PSBSF	Suffix for the PSB table module name.
RBE	Specifies the record buffer extension list.
RETRY	The number of times that ADL tries to put a record, which is currently held by another user, into hold status.
SQ	z/VSE only. Specifies the input/output data set for the Unload utility.
STACK	The size (in kilobytes) of the ADL internal subroutine stack.
TRACE	Activates the Trace facility and specifies what is to be traced. This keyword is described in the section Debugging Aids — the ADL Trace Facility.
UTI	Specifies the CBC utility work area.

PLI Parameter (z/OS Only)

This keyword is used to pass PL/I parameters to PL/I. The feature may only be specified if the PL/I application program has been link edited with "PLICALLB" as its entry point. Specifying "PLI" or "PLI=(...)" will cause the PL/I program to be called in conformity with the standards for calling the entry point "PLICALLB".

Syntax

```
PLI =(options)
```

where *options* are PL/I execution options.

The following values may be entered:

REPORT, NOREPORT, SPIE, NOSPIE, STAE, NOSTAE.

For further details, see the IBM documentation *PL/I Optimizing Compiler: Programmer’s Guide*.

Normal Mode Batch Execution

To invoke normal mode batch execution under ADL, execute the DAZIFP Bridge initialization program. This program requires input parameters similar to those for the DL/I initialization program. Both positional and keyword parameters exist (see above).

Neither the JCL describing the original DL/I data bases nor the load libraries containing the DL/Iload modules are required in normal batch execution.

The ADL load library containing the executable ADL batch module and the Adabas load library must be included in the JCL. ADARUN control statements must also be provided, as is the case with any Adabas application program. For a detailed description of JCL requirements, see the end of this section.

Unlike DL/I, ADL opens all files which are referenced in the PSB during initialization. This must be considered, particularly if the file is opened in EXU mode and the application does not access the file at all.

Mixed Mode Batch Execution

To invoke mixed mode batch execution under ADL, execute the normal DL/I initialization program.

You still require all the JCL describing the original DL/I data bases and the load libraries containing the DL/I load modules.

Before running the program, make the following changes in the JCL/JCS:

1. Change the name of the application program in the DL/I parameters from the user program to DAZIFP. No other parameters have to be modified.
2. Insert an extra input statement for DAZIFP:
   - **z/OS Environments**
     Include a `DD` statement with the DD name DAZIN2 in the JCL.
   - **z/VSE Environments**
     The mixed mode control statement is read from SYSIPT.
3. Add a JCL/JCS statement for the ADARUN file "DDCARD".

For a more detailed description of the JCL/JCS requirements for mixed mode, see the end of this section.

The parameters and syntax for the input statement are the same as for normal mode batch execution (see the first topic of this section).

Note:
DL/I and Adabas are not fully synchronized. Therefore, you should carefully consider restart and recovery procedures for programs running in mixed mode, especially when they are running concurrently to other programs (MPX or SDX run mode).

Pseudo Mixed Mode Batch Execution

In the "pseudo" mixed mode batch execution, the ADL interface program DAZIFP calls a copy of itself. This can be useful when changing structures after the conversion. Pseudo mixed mode is described in details in the section Managing ADL Files.

Link-editing of Application Programs

In general, application programs do not have to be re-linked.

The ADL load library provides the language interface module DAZLIBAT as a substitute for the IBM language interface DFSLI000 (z/OS) and DLZLI000 (z/VSE). These modules provide the entry points ASMTDLI, PLITDLI, CBLTDLI and FORTDLI. You might need the ADL provided language interface in case DL/I is not available at your site, and you have to re-link application programs.

Consistency Interface

This paragraph describes how the ADL Consistency Interface intercepts Adabas calls and how you activate the Consistency in a batch environment.

Activating the Consistency Interface

The ADL Consistency Interface intercepts Adabas calls on the Adabas link module level. In batch, this is a module named ADALNK.

A substitute for the Adabas link module in batch, ADALNK, is delivered with the ADL installation tape. You will have to assemble and link-edit this module, as described in the section below.

Operation of the Consistency Interface requires the original Adabas link module to be present. This, however, has to be renamed. The default new name for the original Adabas link module is ADAOLK. When customizing the ADL substitute, you may choose any new unique name for the original Adabas link module.

Consequently, you have to rename the original Adabas link module for batch, ADALNK, to ADAOLK (or to whatever is the new name you choose.)

It is recommended that the ADL supplied substitute and the renamed copy of the original Adabas link module are both stored on the ADL load library.

For the Consistency Interface to become active, the order of the concatenation of the ADL and Adabas load libraries is of importance. For z/OS batch jobs, this means that the ADL load library must be concatenated before the Adabas load library in the JOBLIB or STEPLIB `DD` statements.

Example:

```
//STEPLIB DD DISP=SHR,DSN=ADL.LOAD
//        DD DISP=SHR,DSN=ADABAS.LOAD
//        DD DISP=SHR,DSN=............
```

In z/VSE, the ADL load library must be defined in the search sequence of a `LIBDEF` statement before the Adabas load library.

Example:

```
// LIBDEF *,SEARCH=(ADL,ADABAS, ... .)
```

Customizing the Consistency Interface

The ADL source library contains the source of the substitute for Adabas link module in batch, ADALNK. The name of this source member is DAZLNK.

All parameters for this module are given as assembler variables and their meaning is explained in comment statements in the source.

All parameters correspond to customizable parameters in the original Adabas link module for batch. Note, however, that you do not have to specify an SVC nor a default DBID when you customize the ADL link module substitute. The values for these will dynamically be taken from the original Adabas link module.

The assembly of the ADL link module substitute requires the ADL source library and the Adabas supplied macro library as well as the system macro library to be available.

See the sections *z/OS Installation* and *z/VSE Installation* in the *ADL Installation* documentation for more details on how to assemble and link-edit the ADL link module.

Parameters for the Consistency Interface

This paragraph describes which parameters are of importance for the Consistency Interface and how you may initialize them. The meaning of the individual parameters, their possible values and their default values are described in the section *ADL Parameter Module* in the *ADL Installation* documentation.

Most of the parameters for the *Consistency Interface* will be initialized by the assembly and link-edit of the ADL parameter module. Parameters of particular importance for the *Consistency Interface* are:

FDT	The size of the "file description table"
FSTAC	The size of the internal format buffer stack
PARM	Do (not) read dynamic parameters
RBSIZ	The size of the internal record buffer
SDT	The size of the "segment description table"

When you have specified "PARM=YES" in the assembly of the ADL parameter module, the *Consistency Interface* will read a card containing dynamic parameters from the file DAZIN1 (z/OS) or the logical unit SYSIPT (z/VSE). The layout of this card is:

```
Keyword=value,...
```

Keywords must be separated by commas, no intervening or leading blanks are allowed. Any undefined keyword causes the scanning to be terminated and the following statements to be ignored.

As described in the section *Keywords for DAZIPF* earlier in this section, more than one line can be used for the parameter specification. You may specify the following dynamic parameters:

DBD	The size (in kilobytes) of the internal control block area for DBDs
DBID	The DBID of the ADL directory file
EBUF	The size (in kilobytes) of the external control block area
FNR	The FNR of the ADL directory file
OPENRQ	Adabas "OP" (not) required
PSB	The size (in kilobytes) of the internal control block area for PSBs
STACK	The size (in kilobytes) of the internal subroutine stack
TRACE	Specify trace parameters. This keyword is described in the section Debugging Aids — the ADL Trace Facility in this documentation.

For a detailed explanation of these parameters, see the *ADL Installation* documentation, section *ADL Parameter Module*.

Table of Converted Adabas Files (DAZTCF)

In order to minimize the effort for batch programs accessing only native Adabas files (i.e. files which do not result from a conversion of a DL/I data base), you may assemble a table containing a list of the converted files. Link-edit it together with the ADL substitute for the Adabas batch link module, ADALNK.

You may use the JCL(JCS) in the member ADLTCF (ADLTCF.J) as an example of how to assemble the table under z/OS (z/VSE).

To create a table of converted files, DAZTCF, you assemble a source containing one call to the ADL supplied macro DAZTCF for each DBID to be included in the table.

The two keyword parameters of the DAZTCF macro are:

DBID=	the DBID to be included in the table
FILES=	list of file numbers to be included in the table

For the FILES= parameter, there are five forms of specifying the argument:

FILES=x	a single file to be included in the table
FILES=(x, ...)	a single file as part of a list
FILES=((x), ...)	a single file as part of a list
FILES=((x,y), ...)	a range of file numbers from x to y inclusive
FILES=ALL	all files for a given DBID

Any number of files may be specified with the FILES= parameter.

DBIDs and file numbers must be specified in ascending sequence. The same DBID can be specified more than once in the list.

Which files are to be included in the table of converted files, DAZTCF, should be evaluated very carefully. Any call referencing a DBID/FNR combination not included in the DAZTCF table will be routed directly to Adabas. Thus, if an ADL file is not included in this table, data integrity will *not* be maintained by the Consistency Interface. On the other hand, the ADL directory file and all Adabas, Natural etc. system files should not be included in the DAZTCF table.

The DAZTCF table can also be used under CICS. Refer to the section CICS Installation and Operation for more details.

Note:
DAZTCF replaces the ADL 2.2 table of non-converted files DAZNCF which is no longer supported.

Example:

```
DAZTCF DBID=001,FILES=ALL
DAZTCF DBID=002,FILES=((14,20),25,(100,200))
DAZTCF DBID=005,FILES=(1060)
END
```

Summary

In short, to activate the ADL Consistency Interface for Adabas or Natural batch jobs you have to perform the following steps:

- create the ADL substitute for the Adabas link module,
- rename the original Adabas link module for batch, ADALNK,
- concatenate the ADL load library before the Adabas load library,
- add a DD card for DAZIN1 (z/OS) or an extra input statement on SYSIPT (z/VSE) when you intend to read dynamic parameters.

Besides that, Adabas or Natural batch jobs run unchanged as described in the corresponding Adabas and Natural documentations.

User Exit

ADL Interfaces optionally pass control to a user written routine (user exit) before each call to Adabas. This user exit may be used for monitoring purposes as well as to apply modifications to the call parameters before Adabas receives control. The conventions for this user exit and how it is activated is described in the section *Miscellaneous* in the *ADL Installation* documentation.

z/OS JCL Requirements

The JCL requirements for batch operation are discussed in the following pages. Both normal mode and mixed mode batch operation are described.

Normal Mode Batch Operation

The following table lists the data sets used by the ADL batch monitor when application programs are run in normal mode.

DDname	Medium	Description
DAZOUT1	Printer	Messages and codes.

Example

```
//G           EXEC PGM=DAZIFP,PARM='DLI,pgmname,psbname'
//STEPLIB     DD DISP=SHR,DSN=ADL.LOAD
//            DD DISP=SHR,DSN=ADABAS.LOAD
//DDCARD      DD *
ADARUN PROGRAM=USER,...
//DAZOUT1     DD SYSOUT=X
//*
//*  application program datasets
//*
```

Mixed Mode Batch Operation

The following table lists the data sets used by the ADL batch monitor when an application program is run in mixed mode.

DDname	Medium	Description
DAZIN2	Reader	Control input for the ADL batch monitor, DAZIFP.
DAZOUT1	Printer	Report, messages and codes.

Example

```
//          EXEC DLIBATCH,MBR=DAZIFP,PSB=psbname
//G.STEPLIB DD
//          DD
//          DD DISP=SHR,DSN=ADL.LOAD
//          DD DISP=SHR,DSN=ADABAS.LOAD
//*
//*  datasets describing DL/I data bases
//*
//G.file    DD .......
//*
//G.DDCARD  DD *
ADARUN PROGRAM=USER,...
//G.DAZOUT1 DD SYSOUT=X
//G.DAZIN2  DD *
MIX,pgmname,psbname
//*
//*   application program datasets
//*
```

DLIBATCH is the standard batch procedure provided by IBM as part of the IMS/DB installation.

z/VSE JCS Requirements

The JCS requirements for batch operation are discussed in the following pages. Both normal mode and mixed mode batch operation are described.

Normal Mode Batch Operation

The following table lists the files used by the ADL batch monitor when application programs are run in normal mode.

DTF	Logical Unit	Medium	Description
DAZIN1	SYSIPT	Reader	Control input for the ADL batch monitor, DAZIFP.
DAZOUT1	SYSLST	Printer	Messages and codes.

The control input for the batch monitor (DAZIFP) and for ADARUN is read from SYSIPT. The control statements must be specified in the following order:

```
DLI,pgmname,psbname,...                     input for DAZIFP
/*
ADARUN DB=dbid,MO=MULTI,PROGRAM=USER,...    input for ADARUN
/*
user input
.
.
/*
```

Example

```
// EXEC PROC=ADLLIBS
// EXEC DAZIFP
DLI,pgmname,psbname
/*
ADARUN PROGRAM=USER,...
/*
/&
```

Mixed Mode Batch Operation

The following table lists the files used by the ADL batch monitor when an application program is run in mixed mode.

DTF	Logical Unit	Medium	Description
DAZIN1	SYSIPT	Reader	Control input for the ADL batch monitor, DAZIFP.
DAZOUT1	SYSLST	Printer	Report, messages and codes.

The control input for the batch monitor (DAZIFP), for ADARUN and for the DL/I initialization program, DLZRRC00, is read from SYSIPT. The control statements must be specified in the following order:

```
DLI,DAZIFP,psbname,...                      input for DLZRRC00
MIX,pgmname,psbname,...                     input for DAZIFP
ADARUN DB=dbid,MO=MULTI,PROGRAM=USER,...    input for ADARUN
/*
user input
.
.
/*
```

Example

```
// EXEC PROC=ADLLIBS
// EXEC DLZRRC00
DLI,DAZIFP,psbname
MIX,pgmname,psbname
ADARUN PROGRAM=USER,...
/*
/&
```