z/VSE Installation Tape

The Adabas Bridge for DL/I installation tape is a standard-label tape. The Report of Tape Creation that accompanies the installation tape lists the volume serial number, tape density, media type, data sets and sequence numbers. The tape is compatible with Software AG's System Maintenance Aid (SMA). Refer to the SMA documentation for more details.

The ADL installation tape for systems operating under the IBM z/VSE Operating System contains four files as described in the table below. The first file has been unloaded to tape using the `IBM LIBR BACKUP` utility, and the other files have been copied by the `IBM IEBGENER` utility.

In this table, *`vrs`* indicates the current version, release and system maintenance (SM) level as indicated on the Report of Tape Creation, for example *`123`* for Version 1, Release 2 and SM 3.

Data Set	DSN	Description
1	ADL*vrs*.LIBR	The ADL Core Image Library, containing the ADL core image load modules. The ADL Relocatable Library, containing ADL relocatable load modules used to link edit executable ADL modules. The ADL Source Library, containing JCS examples for installing the ADL and converting DL/I data bases, as well as macros for reassembling DBDs and PSBs, and source code for sample data base unload programs.
2	ADL*vrs*.SYSF	An unloaded Adabas file for use as the ADL directory file.
3	ADL*vrs*.INPL	An unloaded Natural library containing the ADL Online Services and the ADL Installation Verification Package.

The following sections describe the libraries in more detail.

This chapter covers the following topics:

- ADL Libraries
- ADL Source Library
- ADL Directory File
- ADL Natural Programs

ADL Libraries

The following libraries are described in more detail below:

- ADL Core Image Library
- ADL Relocatable Library

ADL Core Image Library

When loaded from the installation tape, the ADL Core Image Library contains executable ADL load modules. The table below gives the names and descriptions of the modules.

Module	Description
DAZCDUMP	CICS - Write ADL tables on dump file.
DAZCEND	CICS - ADL shut-down program (PLT).
DAZCICS	CICS control program.
DAZCINF	CICS - Return status of ADL.
DAZCINIT	CICS - ADL start-up program (PLT).
DAZCTOFF	CICS - Switch off ADL trace.
DAZCTON	CICS - Switch on ADL trace.
DAZELORE	Establish Logical Relationships utility.
DAZIFP	Batch interface program.
DAZMIX	Batch mixed mode module.
DAZREFOR	Reformat utility.
DAZSYNC	CICS - ADL task-related user exit.
DAZUNDLI	Data Base Unload utility (automated procedure).
DAZZLER	Program to test DL/I calls in batch.

ADL Relocatable Library

When loaded from the installation tape, the ADL Relocatable Library contains the relocatable load modules given in the table below. These are used during installation to create executable ADL load modules.

Module	Description
DAZAXES	ADL nucleus module containing the logic for generating and issuing Adabas calls.
DAZBDOKE	Batch doorkeeper for the ADL Consistency Interface.
DAZBENT	Batch entry point module.
DAZBIFP	Batch interface program (object module).
DAZBRQH	Batch request handler.
DAZCAPRI	CICS application program interface.
DAZCCGEN	ADL nucleus module for the CBC utility.
DAZCCOUT	ADL nucleus module for the CBC utility.
DAZCCSUB	ADL nucleus module for the CBC utility.
DAZCDOKE	CICS doorkeeper for the ADL Consistency Interface.
DAZCDPOS	ADL nucleus module containing the logic for retrieval calls and for maintaining positional information.
DAZCIFP	CICS interface program.
DAZCLUB	CICS – Allocate local user blocks.
DAZCONSI	Environment independent routines of the ADL Consistency Interface.
DAZCRQH	CICS request handler.
DAZDEBUG	ADL nucleus module containing tracing and debugging routines.
DAZDREIN	ADL nucleus module containing the logic for `DELETE, REPLACE` and `INSERT` calls.
DAZEXEC	ADL nucleus module containing the logic for the `EXEC` command precompiler.
DAZINICB	ADL nucleus module containing the logic for initializing internal control blocks.
DAZLANP	ADL nucleus module containing the logic for the language processor.
DAZLIBAT	Language interface for batch mode.
DAZLICID	Language interface for CICS.
DAZSERV	ADL nucleus module containing general purpose service routines.
DAZSTUB	CICS — Stub for the ADL task-related user exit.
DAZSYXTB	ADL nucleus module containing the `EXEC` command syntax.
DAZUEX06	Adabas User Exit 6.
DAZUEXMI	Adabas User Exit 6 for migration to ADL 2.3.
DAZZAP	ADL nucleus module containing zaps for nucleus routines.

ADL Source Library

When loaded from the installation tape, the ADL Source Library contains:

- Macros for creating an ADL parameter module;
- Macros for creating Adabas User Exit 6 extensions;
- Macros for creating the CICS runtime control tables;
- Macros for assembling DBDs and PSBs (substitutes for the original `DL/IDBDGEN` and `PSBGEN` macros);
- Macros for assembling the application control table (substitute for the original `DL/I DLZACT` macro);
- Source code for sample unload programs;
- Source code for performing ADL functions under CICS;
- Source code for the ADL supplied Adabas link module substitutes;
- Sample JCS.

The table below lists the other macros contained in this library.

Member	Description
DAZLDT	Macro used to create entries for logical DBID/FNR assignments in the ADL parameter module. This macro is provided for compatibility with ADL 2.2 only.
DAZTCF	Macro used to create the table of converted files for the Adabas link module substitute in batch.
DAZPARM	Macro used to create the ADL parameter module.
ZFNR	Macro used to create Adabas User Exit 6 extensions.
ZPCK	Macro used to create Adabas User Exit 6 extensions.
ZREC	Macro used to create Adabas User Exit 6 extensions.
ZSEG	Macro used to create Adabas User Exit 6 extensions.
ZSEX	Macro used to create Adabas User Exit 6 extensions.
ZVCK	Macro used to create Adabas User Exit 6 extensions.
MGPSTIN	Macro used to create the DAZPSB table.
MGPSTEN	Macro used to create the DAZPSB table.
MGPSTFI	Macro used to create the DAZPSB table.
DLZACT	Macro used to create a DAZPSB or DAZACT table.
DBDMAC	Macro used to create the DAZDBD table.
BUFMAC	Macro used to create the DAZBUF table.

The following table lists the source members used during the assembly of the ADL supplied Adabas link module substitutes:

Member	Description
DAZLNKD	Source member to be assembled as the ADL supplied Adabas link module substitute in batch.
DAZLNK	Operating system independent part of the ADL supplied Adabas link module substitute in batch.
LNKDOS	Operating system dependent part of the ADL supplied Adabas link module substitute in batch.

The following table lists the sample JCS streams for the installation and conversion processes:

Member	Description
ADLINS1.J	JCS for loading the ADL libraries from tape.
ADLINS2.J	JCS for loading the ADL directory file from tape.
ADLINS2A.J	JCS for updating an existing ADL directory file.
ADLINS3.J	JCS for an initial program load of the ADL Online Services and the ADL supplied Natural subprograms ADLERROR and ADLACTIV and an initial program load of the Natural programs for the ADL Installation Verification Package .
ADLINS4.J	JCS for creating an ADL parameter module.
ADLINS5U.J	JCS for creating an executable ADL CBC utility.
ADLINS6P.J	JCS for creating an executable ADL precompiler module.
ADLINS7B.J	JCS for creating an executable ADL batch module.
ADLINS8A.J	JCS for creating an executable ADL Consistency Interface module for batch.
ADLINS9C.J	JCS for creating an executable ADL CICS module.
ADLCSD	Sample CICS system definition file.
ADLCTG1.J	JCS for assembling the DAZPSB table.
ADLCTG2.J	JCS for running the DAZSHINE utility.
ADLCTG3.J	JCS for assembling the DAZBUF table.
ADLCTG4.J	JCS for assembling the DAZDBD table.
ADLCFCT.J	JCL for precompile, assembly and link-edit of DAZCFCT.
ADLTCF.J	JCS for assembly of the DAZTCF table.

Note the naming conventions for the JCL examples:

```
ADLxxxnn    General sample JCL
IVPxxxnn    JCL for the Installation Verification package
```

where *xxx* represents one of the following suffixes

INS	for the installation steps (section *z/VSE Installation* in this documentation);
DPC	for the DBD/PSB conversion steps (section *ADL Conversion Utilities for DBDs and PSBs* in the *ADL Conversion* documentation ;
DBC	for the data base conversion steps (section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation;
CTG	for the CICS table generation (section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation);
TCF	for the generation of the converted Adabas file table (see the section *Batch Installation and Operation* in the *ADL Interfaces* documentation);

and *nn* is the number of the step.

The following table lists the remaining members in the Source Library.

Member	Description
$INFO$	Information about the current ADL release.
ADLEX06	ADACMP User Exit 6 skeleton, for changing the layout of an ADL file.
ADLIMEX	Sample user exit routine for index maintenance.
ADLXPCn	COBOL sources for the ADL IVP.
ADLXPIn	Input streams for the COBOL programs for the ADL IVP.
ADLXPAn	Assembler sources for the ADL IVP to run under CICS.
ADLXPDn	DAZZLER input streams for the ADL IVP.
COURSEDB	Physical DBD definitions for the ADL IVP.
COURSEL	Logical DBD definitions for the ADL IVP.
COURSUNL	PSB definitions for the ADL IVP (DAZUNDLI utility).
DAZUNLOD	Source of the sample unload program.
DAZREFOR	Source of the sample reformat program.
DAZCFCT	Source of the sample program to perform ADL functions under CICS.
DAZEISTG	DSECT used by DAZCFCT
INSTDB	Physical DBD definitions for the ADL IVP.
INSTIDX	Primary Index DBD definition for the ADL IVP.
INSTL	Logical DBD definitions for the ADL IVP.
INSTUNL	PSB definitions for the ADL IVP (DAZUNDLI utility).
INSTELO	PSB definitions for the ADL IVP (DAZELORE utility).
IVPINFO	Abbreviations used by the ADL Installation Verification Package (IVP).
IVPARUN	ADARUN cards for the ADL IVP.
IVPCOB	Execute a COBOL batch program of the ADL IVP.
MAINIDX	Primary Index DBD definition for the ADL IVP.
SCHOOL	PSB definitions for the ADL IVP.
STUDIDX	Secondary Index DBD definition for the ADL IVP.

ADL Directory File

The unloaded Adabas Directory file on the installation tape was created by the Adabas Unload utility, ADAULD. At initialization, the file contains the texts of the ADL error messages. Later, it will also be used for storing the DBDs and PSBs for ADL and any checkpoint information.

ADL Natural Programs

This file contains the unloaded Natural programs comprising the ADL Online Services, together with the ADL supplied Natural subprograms ADLERROR and ADLACTIV and the ADL Installation Verification Package. The ADLERROR subprogram may be used by Natural applications to retrieve the comprehensive error messages of the ADL Consistency Interface. The ADLACTIV subprogram may be used by Natural applications to verify whether the ADL Consistency Interface is active or not.

The files were created with the Natural Object Handler (SYSOBJH).