

# Installing Adabas for OS IV/F4

This section describes how to prepare for and install Adabas on Fujitsu Ltd.'s OS IV/F4 (including MSP and MSP 20/AE/EX) operating systems.

For information about the prerequisite system levels supported by Adabas, refer to the section *Supported Environments*.

## Note:

Adabas 7.4.2 is the most current Adabas release to support MSP. Instructions for using Adabas under MSP are provided in this 7.4.4 documentation only as a convenience for MSP users of Adabas 7.4.2.

- Installation Checklist
- Preparing to Install Adabas
- Installing the Release Tape
- Initializing the Adabas Communication Environment
- Installing an Adabas Database
- SVC Integrity Validation
- Requirements for Cross-Memory Services
- Using EXCPVR
- Creating a Shareable ADARUN
- Storage Above 16 MB
- Adalink Considerations

## Installation Checklist

### Note:

The MSP environment now has two tape versions. The newer tape contains MSP/EX support for cross-memory services.

The following is an overview of the steps for installing Adabas on an OS IV/F4 system.

Step	Description	Additional Information
	<b>Basic Installation Steps</b>	
1	Allocate DASD space for the Adabas libraries.	The libraries are restored from the installation tape. Refer to the section <i>Disk Space Requirements for Libraries</i> .

Step	Description	Additional Information
2	Allocate DASD space for the Adabas database.	For better performance, distribute the database files over multiple devices and channels. Refer to the section <i>Disk Space Requirements for the Database</i> .
3	Specify the partition or OS IV/F4 address space for running the Adabas nucleus.	Refer to the section <i>Adabas Nucleus Address Space Requirements</i> .
4	Restore the Adabas libraries from the installation tape.	Use the tape positioning information that accompanies the tape. Refer to the section <i>Installing the Release Tape</i> .
5	Install the Adabas SVC.	Refer to the section <i>Initializing the Adabas Communication Environment</i> .
	<b>Database Installation Steps</b>	Steps 6-15 require changes to the setup definitions as described in the section <i>Installing an Adabas Database</i> .
6	Allocate and format the Adabas database with the ADAFRM utility job.	
7	Define the global database characteristics with the ADADEF utility job.	
8	Load the demonstration files with the ADALODE, ADALODV, and ADALODM jobs.	
9	Start the Adabas nucleus and test the Adabas communications with the ADANUC job.	
10	If appropriate, test Adabas address space communications by running ADAREP in MULTI mode with the CPEXLIST parameter.	
11	If appropriate, load the Adabas Online System (AOS) selectable unit into a Natural system file by running the AOSINPL job. Alternatively, install the AOS demo version.	See section <i>Installing the AOS Demo Version</i> .
12	Terminate the Adabas nucleus with an ADAEND operator command using the OS Modify command F.	
13	Back up the database by running the ADASAV utility job.	
14	Insert the ADARUN defaults by running the DEFAULTS job.	
	<b>TP Monitor Installation</b>	

Step	Description	Additional Information
15	Install the required TP link routines for Adabas.	See section <i>Installing Adabas with TP Monitors</i> .

## Preparing to Install Adabas

The major steps in preparing for Adabas installation are

- check for the correct prerequisite system configuration; and
- allocate disk and storage space.

The following sections discuss the nominal disk and storage space requirements and how to allocate the space.

### Disk Space Requirements for Libraries

The minimum 3390 disk space requirements for the Adabas libraries are as follows:

Library	3390 Cylinders	3390 Tracks	Directory Blocks
Load	8	120	40
Source	3	45	20
JCL	1	15	20

#### Note:

You can isolate user programs from the Adabas load library by creating a separate load library that contains only those modules needed to execute user programs in multi-user mode and linked with ADAUSER. For Version 7.4, the modules required by user programs are ADAIOR, ADAIOS, ADALNK, ADAMLF, ADAPRF, ADARUN.

### Disk Space Requirements for the Database

The actual database space needed by Adabas depends on user requirements. The minimum 3390 disk space requirements for the database are as follows:

Database Component	3390 Cylinders	3390 Tracks
ASSOR1 (Associator)	20	300
DATAR1 (Data Storage)	60	900
WORKR1 (Work space)	15	225
TEMPR1 (temporary work space)	15	225
SORTR1 (sort work space)	15	225

## Adabas Nucleus Address Space Requirements

The typical Adabas nucleus requires at least 800-1024 kilobytes to operate. The size of the nucleus address space may need to be larger, depending on the ADARUN parameter settings. Parameter settings are determined by the user.

## Installing the Release Tape

The following job control (JCL) restores the Adabas libraries:

```
//ADAREST JOB ...
//STEP1 EXEC PGM=IEBCOPY
//LOADT DD DSN=ADAvrs.LOAD,DISP=(OLD,PASS),
// UNIT=TAPE,VOL=SER=vvvvvv,
// LABEL=(n,SL)
//SOURCET DD DSN=ADAvrs.SRCE,DISP=(OLD,PASS),
// UNIT=TAPE,VOL=SER=vvvvvv,
// LABEL=(n,SL)
//JCLT DD DSN=ADAvrs.JOBS,DISP=(OLD,PASS),
// UNIT=TAPE,VOL=SER=vvvvvv,
// LABEL=(n,SL)
//*
/* ADABAS VERSION v LIBRARIES:
/*
//LOAD DD DSN=ADAvrs.LOAD,DISP=(NEW,CATLG),
// UNIT=dasdev,VOL=SER=volser,
// SPACE=(CYL,(cyl,,blk)),
// DCB=(RECFM=U,BLKSIZE=6447)
//SOURCE DD DSN=ADAvrs.SRCE,DISP=(NEW,CATLG),
// UNIT=dasdev,VOL=SER=volser,
// SPACE=(CYL,(cyl,,blk)),
// DCB=(RECFM=FB,BLKSIZE=6000,LRECL=80)
//JCL DD DSN=ADAvrs.JOBS,DISP=(NEW,CATLG),
// UNIT=dasdev,VOL=SER=volser,
// SPACE=(CYL,(cyl,,blk)),
// DCB=(RECFM=FB,BLKSIZE=6000,LRECL=80)
//SYSPRINT DD SYSOUT=X
//SYSIN DD *
COPY INDD=LOADT,OUTDD=LOAD
COPY INDD=SOURCET,OUTDD=SOURCE
COPY INDD=JCLT,OUTDD=JCL
/*
//
-where
blk is the number of directory blocks for this library
cyl is the number of cylinders for this library
vrs is the Adabas version level
vvvvvv is the volume serial number for the Adabas installation tape
n is the position of the data set on the Adabas installation tape
volser is the volume serial number of the disk used for the Adabas library
dasdev is the DASD device type
```

Refer to the *Report of Tape Creation* for the correct Adabas library sequence. The DCB statements in the above example are not required for the IEBCOPY step.

# Initializing the Adabas Communication Environment

This section describes the installation of the Adabas router (ADASVC). The router uses cross-memory services for communication between the Adabas nucleus and the Adabas users.

The Adabas cross-memory communications service comprises two modules:

- the Adabas router (ADASVC); and
- the Adabas subsystem initialization routine (ADASIR).

ADASIR, executed either during IPL or by the Adabas SVC installation program (ADASIP), initializes the router’s operating environment, particularly the ID table.

ADASIR must be linked into either SYS1.LINKLIB or into a library concatenated to SYS1.LINKLIB by the LNKLSTxx member of SYS1.PARMLIB. This requires re-IPLing the system to install the Adabas SVC.

**Note:**

The CLPA option must first be specified as an IPL-selected option.

## Router Installation Overview

The following steps are required for installing the router:

- Step 1: Restore the Adabas Installation Tape
- Step 2: Apply the Zap To Define the Subsystem ADAB
- Step 3: Re-IPL the Operating System with CLPA To Load and Execute ADASIR

### Step 1: Restore the Adabas Installation Tape

Restore the Adabas Installation Tape as described in the section *Installing the Release Tape*.

### Step 2: Apply the Zap To Define the Subsystem ADAB

Apply the zap to define the subsystem ADAB as described in the section *Executing ADASIR*.

If necessary, also do the following operating-system-dependent steps:

OS IV/F4	Apply the zap to modify the SVC table entry for a type 3 SVC as described in the section <i>Setting the SVC Table</i> .
MSP/EX	Modify the member KAASVCnn to define the Adabas router under MSP/EX as described in the section <i>Setting the SVC Table</i> .
MSP/EX	Modify the member SUBSYSxx to add the subsystem as described in the section <i>Executing ADASIR</i> .

Apply the zap as described in the section *Setting the ID Table Count and SVC Number*.

If necessary, relink the SVC as described in the section *Linking the Adabas SVC*.

### Step 3: Re-IPL the Operating System with CLPA To Load and Execute ADASIR

Re-IPL the operating system with CLPA to load and execute ADASIR.

### Setting the SVC Table

OS IV/F4 systems must define the SVC as a type 3 with no locks.

The VERIFY and REPLACE statement examples in the following sections show how to alter an available SVC entry to define it as type 3.

#### For OS IV/F4 MSP E20

Use the following statements to alter an available SVC entry to define it as type 3:

```
NAMEX KANUC01 SVCTABLE
VER offset xxxx
REP offset C000 (type 3, no locks)
-where
offset is the offset to byte 4 of the SVC entry for the specified SVC.
xx (or) xxxx is the existing contents of SVC bytes 4 and 5 at the specified offset.
```

#### Note:

The SVC table entry is eight bytes long. The fourth byte of each entry specifies the type attribute, and the fifth byte specifies the lock attribute (see bytes 5 and 6 of the following E20 AE example).

#### For OS IV/F4 MSP E20 AE

Use the following statements to alter an available SVC entry to define it as type 3:

```
NAMEX KANUC01 SVCTABLE
VER offset xxxx xxxx
REP offset C000 00yy
-where
offset is the offset to byte 5 of the SVC entry for the specified SVC.
xxxx xxxx is the existing contents of SVC bytes 5-8 at the specified offset.
yy is the value of the selected AMODE option as described below in flag 4 (for
example, B'10nn nnnn' for AMODE=31).
```

#### Note:

The SVC table entry is eight bytes long, and has the following format:

Bytes	Meaning
1 - 4	SVC entry address
5	Flag byte:  Type 1 SVC: B'0000 nnnn' Type 2 SVC: B'1000 nnnn' Type 3 and 4 SVC: B'1100 nnnn' Type 6 SVC: B'0010 nnnn'
6	Flag 2:  Local lock: B'10nn nnnn' CMS lock: B'11nn nnnn'
7	Flag 3 (reserved)
8	Flag 4:  AMODE=24: B'00nn nnnn' AMODE=31: B'10nn nnnn' AMODE=ANY: B'01nn nnnn'

### For OS IV/F4 MSP 20AE/EX

Either create or modify member KAASVCnn in SYS1.PARMLIB. This member describes the attributes of the SVCs used in the MSP/EX system.

The following characteristics can be used for the Adabas SVC:

- Type 3
- No locks

The entry format is:

```
NUM=svc,TYPE=3,EPNAME=entryname,AM=31,APF=NO
-where
svc is the Adabas SVC number.
entryname is the Adabas SVC entry name.
```

### Setting the ID Table Count and SVC Number

ADASIR detects and supports either the MSP/20, or MSP/AE and MSP/EX without cross-memory services, based on PTF level 91212 or above running on a FACOM model M760 or higher.

ADASIR does not accept passed parameters. As a result, the ID table entry count, SVC number, and operating mode must be zapped as follows:

```
NAMEX ADASIR
* ADABAS 7 SVC number
VER 0034 0A00 (this default is invalid)
REP 0034 0Axx (where xx is the SVC number)
```

To enter the allowed maximum count of ID table entries, issue the following zap:

```

NAMEX ADASIR
* number of ID Table entries
VER 0020 0000,000A (default = 10)
REP 0020 0000,00xx (where xx is the maximum entry count)

```

To set the operating environment, make the following ZAP;

```

NAMEX ADASIR ADASIR
VER 0024 00 (this default is invalid)
REP 0024 xx
(where xx is the operating environment, as follows:
01: native mode
02: AVM/EF mode
03: AVM/EX mode)

```

**Note:**

To get the VMID under AVM/EX, you must set the QAL for the AS option of AVM; otherwise, the VMID is NO.

## Using ADASIR

ADASIR receives control when the master scheduler is initialized. The operator is prompted for NRIDTES and SVCNR values that were not correctly zapped. Message ADAS03 indicates that ADASIR ended normally. Message ADAS04 indicates an error condition that prevents Adabas from being used until the problem is corrected and the system is re-IPLed. For specific message information, refer to the *Adabas Messages and Codes*.

- Linking the Adabas SVC
- Executing ADASIR

### Linking the Adabas SVC

Link the Adabas OS IV/F4 type 3 SVC into SYS1.LPALIB with the name JFF00*nnp*, where *nn* are the first two digits of the SVC number and *p* is the character equivalent of the last SVC digit (as an example, when the last SVC digit is 1, *p* must be set to A), which must be non-zero.

### For MSP 20 AE Systems

For OS IV/F4 MSP 20 AE systems, the following example applies:

```

//A42L31 JOB CLASS=A,MSGCLASS=X
//IEWL2 EXEC PGM=IEWL,
// PARM='LET,LIST,NCAL,XREF,RENT,SECTION=(31)',REGION=1024K
//SYSPRINT DD SYSOUT=X
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//ADALIB DD DSN=user.loadlib,DISP=SHR <--ADASVC loadlib
//SYSLMOD DD DSN=SYS1.LPALIB,DISP=SHR
//SYSLIN DD *
CHANGE ADASVC(JFF00nnp)
INCLUDE ADALIB(ADASVC)
NAME JFF00nnp(R)
/*

```



## For Non-MSP 20 AE Systems

The following example shows how to link the SVC for non-MSP 20 AE systems:

```
// (job card)
//LKED EXEC PGM=IEWL,
// PARM='XREF,LIST,LET,NCAL,RENT,REUS'
//SYSPRINT DD SYSOUT=X
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLMOD DD DSN=SYS1.LPALIB,DISP=SHR <- -Target loadlib
//ADALIB DD DSN=user.loadlib,DISP=SHR <--ADASVC loadlib
//SYSLIN DD *
CHANGE ADASVC(JFF00nnp)
INCLUDE ADALIB(ADASVC)
NAME JFF00nnp(R)
/*
```

## Executing ADASIR

### For OS IV/F4 Environments

Zap the ADASIR entry into the subsystem table named KDJSSNT in SYS1.LINKLIB.

Entries in the subsystem table are 80 bytes long. The table begins directly with the first entry.

To find an available entry for Adabas, dump the module KDJSSNT in SYS1.LINKLIB and search for an empty 80-byte entry. Empty entries contain all blanks.

Use the following zap to add ADAB as a valid subsystem:

```
NAMEX KDJSSNT KDJSSNT
VER offset 40404040,4040404040404040
REP offset C1C4C1C2,C1C4C1E2C9D94040
```

—where *offset* is the offset to the beginning of the first available subsystem table entry specifying ADAB, ADASIR, as defined in the REP statement.

#### Note:

The ADASIR program loads a channel appendage KHH019KU from SYS1.LINKLIB.

At this point, you are ready to IPL the ADASIR routine (with CLPA). Any messages or codes occurring while executing ADASIR are described in the *Adabas Messages and Codes*.

### For OS IV/F4 MSP (E20, AE and EX) Environments

Define a SUBSYSTEM entry for Adabas by either creating or modifying the member SUBSYSxx in SYS1.PARMLIB. The format is as follows:

```
SNDSUB SUBNAME=name,PGM=ADASIR
```

—where *name* is the subsystem name (the default is ADAB). The value for *name* must be the same as the name in the Adabas router, offset X'28'.

If you already have an “ADAB” subsystem in the system, apply the following zap:

```
NAMEX ADASVC
VER 0028 C1C4C1C2
REP 0028 subname
```

—where *subname* is a four-character name other than ADAB specified by SUBNAME.

## Installing an Adabas Database

Once you have restored the Adabas installation tape and have installed the ADASVC, you can

- migrate an existing Adabas database to the new version; or
- install a new version of the Adabas database.

Messages or codes that occur during the installation are described in the *Adabas Messages and Codes*; utilities are described in *Adabas Utilities*.

- Migrate an Existing Database
- Install a New Database

### Migrate an Existing Database

Use the ADACNV utility to migrate existing databases to new releases of Adabas (SMA job number I051). See *Adabas Utilities* for more information.

### Install a New Database

#### Step 1: Allocate and Format the Adabas Database

Customize and run the ADAFRM utility job to allocate and format the Adabas database. The following must be customized (for references to Adabas utility parameters; see the specific utility descriptions in the *Adabas Utilities* documentation):

- data set names for the database and libraries;
- volumes for libraries and data sets for the database;
- space allocation for data sets for the database;
- the Adabas SVC number, the database ID, and database device type(s);
- sizes of the data sets for each ADAFRM statement.

#### Step 2: Define the Global Database Characteristics

Customize and run the ADADEF utility job to define the global definition of the database. The following must be customized:

- data set names of the database and libraries;

- the Adabas SVC number, the database ID, and database device type(s);
- ADADEF parameters.

### **Step 3: Load the Demonstration Files**

Customize and run the job:

- ADALODE to load the sample demo file EMPL;
- ADALODV to load the sample demo file VEHI; and
- ADALODM to load the sample demo file MISC.

For each job, the following must be customized:

- data set names for the database and libraries;
- the Adabas SVC number, the database ID, and database device type(s);
- ADALOD parameters.

### **Step 4: Start the Adabas Nucleus and Test the Adabas Communications**

Customize and run the job ADANUC to start up the Adabas nucleus. The following must be customized:

- data set names for the database and libraries;
- the Adabas SVC number, the database ID, and database device type(s);
- ADANUC parameters.

### **Step 5: Test Adabas Address Space Communications, If Appropriate**

Customize and run the job ADAREP in MULTI mode with the CPEXLIST parameter to test Adabas address space communications. The following must be customized:

- data set names for the database and libraries;
- the Adabas SVC number, the database ID, and database device type(s);
- ADAREP parameters.

### **Step 6: Load the Adabas Online System Selectable Unit, If Appropriate**

Customize and run the job AOSINPL to load the Adabas Online System (AOS) into a Natural system file using a batch version of Natural 3.1 or above. The following must be customized:

- data set names of the database and libraries;
- the Adabas SVC number, the database ID, and device type(s);

- the Natural INPL parameters and system file number.

Alternatively, install the AOS demo version delivered with Adabas: see section *Installing AOS Demo Version*.

### Step 7: Terminate the Adabas Nucleus

Communicate with the Adabas nucleus to terminate the session either with an ADAEND operator command using the OS Modify command

```
F jobname ,ADAEND
```

—or

```
P jobname
```

—where *jobname* is the job or task name of the started nucleus.

### Step 8. Back Up the Database

Customize and run the ADASAV utility job to back up the database. The following must be customized:

- data set names of the database and libraries;
- the Adabas SVC number, the database ID, and device type(s);
- ADASAV parameters.

### Step 9: Insert the ADARUN Defaults

The member DEFAULTS in the Adabas JCL library can be modified to set the ADARUN defaults. The following must be customized:

- data set names of the database and libraries;
- ADARUN user defaults:
  - device type(s) (default: 3380)
  - SVC number (default: 249)
  - database ID (default: 1)

Customize and run the DEFAULTS job to set the ADARUN defaults using the OS ZAP utility.

### Step 10: Install the Required TP Link Routines for Adabas

Refer to the section *Installing Adabas with TP Monitors* for a description of the TP link routine procedure.

## SVC Integrity Validation

In the past, the presence of multiple SVCs with the same subsystem ID has resulted in a single ID table being used by different SVCs. This has caused problems, some of them serious (abnormal nucleus termination or corruption of the database).

To eliminate this danger, the Version 7 SVC checks to ensure that the SVC accessing the ID table is the same as the one that was used by ADASIP/ADASIR to initialize the table. If the SVCs are not the same, an abend 650 occurs.

Abend 650 occurs when an incorrect SVC number is specified in the ADARUN parameters for a nucleus. It can occur during Adabas initialization, during the first Adabas call from a user program, or when the ID table is queried by another Software AG server such as Entire Net-Work.

## Requirements for Cross-Memory Services

The following hardware/software is required to use cross-memory services with MSP EX:

- FACOM M760, or above;
- MSP EX level PTF91121 operating system, or above.

With cross-memory services, the Adabas command queue and attached buffers are located in the address space instead of the common storage area (CSA), where all components except the ID table are usually located.

When cross-memory services is used, the number of system linkage indexes can be increased from the default of 55 (X'37') by using the following zap:

```
NAMEX KANUC01 KAACMTBL
VER 004E 0037 (default = 55)
REP 004E 0xxx (maximum: 1024)
```

To get the VMID under AVM/EX, you must set the ASP option of AVM to ON; otherwise, the VMID will be null.

## Using EXCPVR

CPU usage is reduced considerably by using EXCPVR. This requires that ADARUN be APF-authorized and that all Adabas modules be contained in APF-authorized libraries. When placed in an APF-authorized library, ADARUN allows the Adabas nucleus and utilities to use EXCPVR.

All Adabas modules except the Adalink must run with RMODE=24 (the default); the Adalink can run with any RMODE.

When the EXEC statement specifies PGM=ADARUN, the AMODE is determined as follows:

- The default AMODE=31 is used unless ADARUN was relinked using either the AMODE=24 EXEC statement parameter or the MODE AMODE(24) linkage editor control statement.
- An AMODE of 31 is changed to 24 before affected macros (data management, for example) are run, and then changed back to 31 thereafter.

Adabas performance can be improved by using EXCPVR to improve channel program translation time. For Adabas to invoke EXCPVR automatically, the Adabas modules must be in an APF-authorized library and ADARUN must be linked with SETCODE AC(1), the default, as shown in the following example:

```
//LINKRUN EXEC PGM=IEWL,PARM='REUS'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//ADALIB DD DSN=user.loadlib,DISP=SHR
//SYSLMOD DD DSN=user.loadlib,DISP=SHR <==APF-authorized library
//SYSLIN DD *
INCLUDE ADALIB(ADARUN)
SETCODE AC(1)
NAME ADARUN(R)
```

## Creating a Shareable ADARUN

The ADARUN module delivered in the Adabas load library is neither reentrant nor reusable.

A reusable ADARUN permits several programs running in the same address space to share the same ADARUN and ultimately, the same copy of ADALNK. This is important when it is necessary to have only one Adabas user ID for the different programs, and is also needed if single copies of ADALNK user exits are required.

To create a shareable ADARUN, re-link it with the REUS attribute using the sample job JCLLINRR in the MVSJOBS library.

If both nonreusable and reusable versions of ADARUN are required, they must be located in different load libraries since both must be loadable using the name ADARUN.

## Storage Above 16 MB

Adabas can acquire a number of its required areas, including buffer space, above the 16-MB addressing limit, allowing Adabas to increase the buffer pool size. Storage above 16 MB is available on OS IV/F4 MSP 20 AE and MSP/EX operating systems.

To reverse the space allocation to be below the 16-MB limit, set the SECTION value in the PARM statement in the example below to "(24)":

```
//LINKRUN EXEC PGM=IEWL,
PARM='LET,LIST,NCAL,XREF,RENT,SECTION=(24)'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//ADALIB DD DSN=user.loadlib,DISP=SHR
//SYSLMOD DD DSN=user.loadlib,DISP=SHR <==APF-authorized library
//SYSLIN DD *
INCLUDE ADALIB(ADARUN)
SETCODE AC(1)
NAME ADARUN(R)
```

In addition, Adabas must be run with a sufficient REGION specification, either on the JOB or EXEC statement or as an installation default. For example:

```
//BIG JOB ...,REGION=30M,...
```

The attached buffers and the Adabas command queue can also be allocated in storage above the 16-MB line. These components are allocated in ECSA if the SVC table entry FLAG 4 specifies AMODE=31.

## Adalink Considerations

### Note:

For information about connecting a database that is enabled for data conversion using the universal encoding service (UES), see the section *Connecting UES-Enabled Databases*.

### User Exit B (Pre-Command) and User Exit A (Post-Command)

One or two user exits may be linked with an Adalink routine (SMA job number I088):

- UEXITB receives control before a command is passed to a target with the router 04 call.

### Note:

Special commands emanating from utilities and from Adabas Online System are marked as physical calls. These calls must be bypassed in user exits. These calls have X'04' in the first byte (TYPE field) of the command's Adabas control block (ACB). UEXITB must check this byte and return if it is set to X'04'. Be sure to reset R15 to zero on return.

- UEXITA receives control after a command has been completely processed by a target, the router, or by the Adalink itself.

At entry to the exit(s), the registers contain the following:

Register	Contents
1	Address of the UB.  If the flag bit UBFINUB is reset, the contents of the halfword at Adabas + X'86' have been moved to UBLUINFO. If those contents are greater than zero, the two bytes starting at UBINFO (UB+X'40') have been set to zero.  If UBFINUB is set, no changes can be made to the UB or ACB (except for ACBRSP).
2	Address of a 16-word save area (for ADALNC only)
13	Address of an 18-word save area (for non-CICS Adalink exits)
14	Return address
15	Entry point address: UEXITB or UEXITA

Any registers except register 15 that are modified by the user exits must be saved and restored; the address of a save area for this purpose is in register 13 (or register 2 for ADALNC).

If at return from UEXITB register 15 contains a value other than zero, the command is not sent to the target but is returned to the caller. The user exit should have set ACBRSP to a non-zero value to indicate to the calling program that it has suppressed the command: response code 216 is reserved for this purpose.

The UEXITB exit may set the UB field UBLUINFO to any lesser value, including zero; an abend occurs if the user exit sets UBLUINFO to a greater value. The UBLUINFO length cannot be changed when any other exit is used; for example, Adabas Review or Adabas Fastpath.

The user information received by a UEXITA exit may have been modified; this modification may include decreasing its length, possibly to zero, by any of the Adalink user exits.

An Adalink routine can return the following non-zero response codes in ACBRSP:

Response Code	Description
213	No ID table
216	UEXITB suppressed the command
218	No UB available

At least the following three equates, described at the beginning of the source, can be modified before an Adalink routine is assembled. In some Adalink routines, however, the corresponding information can be zapped:

Equate	Description
LOGID	The default logical ID, ranging in value from 1 to 65535. The default is 1.
LNUINFO	The length of the user information to be passed to Adalink user exits, ranging in value from 0 to 32767. The default is 0.
SVCNR	The Adabas SVC number; its range of values and the default depend on the operating system. This value can be provided as SYSPARM value for assembly of the following Adalink routine:  //EXEC PGM=ass,PARM='.....,SYSPARM( <i>svcnr</i> )'

The first 152 (X'98') bytes of all Adabas Adalinks must maintain the following structure:

Offset	Label	Contents	Meaning
00	ADABAS		Entry code
12		CL6'ADALNx'	Program name
18		XL4'yyyymmdd'	Assembly date
1C		A(ZAPTAB)	Address of zap table
20	PATCH	XL96'00'	Patch area
80	LNKLOGID	AL2(LOGID)	Default logical ID (default: 1)
82		82 XL2'00'	Reserved
84	LNKSVC	SVC SVCNR	Executable SVC instruction for Adabas SVC (default: operating-system-dependent)
86	LUINFO	Y(LNUINFO)	Length of user information (default: 0)
88	VUEXITA	V(UEXITA)	Address of user exit after call (weak)
8C	VUEXITB	V(UEXITB)	Address of user exit before call (weak)
90	ADABAS51	CL8'ADABAS51'	IDT ID



## ADAUSER Considerations

ADAUSER is a program that links the user to Adabas. It is specific to an operating system and is independent of release level and mode. It can be used in batch and in some TP environments (such as TSO).

ADAUSER operates in the following way:

- ADAUSER contains the entry point ADABAS and should be linked with all user programs that call Adabas. No other programs containing the CSECT or entry point name ADABAS can be linked in these load modules/phases.
- On the first call to Adabas, ADAUSER loads the latest version of ADARUN. This makes the calling process release-independent. Subsequent Adabas calls bypass ADARUN.
- ADARUN processes its control statements. If the ADARUN PROGRAM parameter has the (default) value USER, ADARUN loads ADAIOR and, depending on whether the ADARUN MODE parameter specifies MULTI or SINGLE, loads the appropriate TP Adalink or ADANUC, respectively. This makes the calling process mode-independent.