

Installing Adabas for BS2000/OSD

This section describes the preparation for and installation of Adabas on systems running under the Fujitsu Technology Solutions BS2000 operating system.

Information for your specific installation is contained in the *BS2000 Product Installation Package* distributed with the System Installation Aid.

- Installation Checklist
 - Contents of the Release Tape
 - Preparing to Install Adabas
 - Installing the Adabas Release Tape
 - Installing an Adabas Database
 - The Adabas BS2000 Communication Environment
 - Considerations of Device Types
 - Applying Zaps
 - Job Variable (JV) Handling
 - Job Switches
 - Linking User Exits B and A to ADALNK
 - ADALNK Optimizer Exit OPTEXIT
 - Enabling Universal Encoding Support (UES) for Your Adabas Nucleus
 - Linking LNKUES to ADALNK for Data Conversion
 - Formatting New Adabas Libraries
 - Loading Elements into the Time Zone Library (TZ00)
 - Starting Adabas
 - Interpreting BS2000 Error Messages
-

Installation Checklist

The following list provides an overview of the Adabas installation procedure on BS2000 systems.

Step	Description	Additional Information
1	Provide disk space for the Adabas libraries.	The libraries are restored from the installation tape into these libraries. Refer to the section <i>Adabas Library Disk Space Requirements</i> .
2	Allocate disk 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 address space for running the Adabas nucleus.	Refer to the section <i>Adabas Nucleus Address Space Requirements</i> .
4	Install the product license file (job I010, steps 0010 and 0011).	For more information, read <i>Step 1: Check, Prepare, and Install the Product License File (SMA job number I010)</i>
5	Allocate and format the Adabas database with the ADAFRM utility job (job I030, steps 0010 to 0018).	
6	Define the global database characteristics with the ADADEF utility job (job I030, steps 0020 and 0021).	
7	Start the Adabas nucleus and test the Adabas communications with the ADANUC job.	
8	Load the demonstration files ADA _{vrs} .EMPL/MISC/VEHI/PERS/PERL with the ADALOD utility (performed in job I050 steps 0001 to 0003). Note: The Personnel demo file must be installed on a UES-enabled database because it includes wide-character format (W) fields.	
9	If appropriate, test Adabas address space communications by running ADAREP (job I050 step 9999).	
10	If appropriate, load the Adabas Online System (AOS) selectable unit into a Natural system file by running the ADAINPL job. Alternatively, install the AOS demo version delivered with Adabas (job I061, steps 0111 and 0112).	

Step	Description	Additional Information
11	Terminate the Adabas nucleus with an ADAEND operator command.	
12	Back up the database by running the ADASAV utility job.	

Contents of the Release Tape

The following table describes most of the libraries included on the release tape. Once you have unloaded the libraries from the tape, you can change these names as required by your site, but the following lists the names that are delivered when you purchase Adabas for BS2000 environments.

Library Name	Description
ADAvrs.ECnn	The Adabas library containing character encoding members to support various languages and Unicode. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas. The <i>nn</i> letters in the library name represents a number from "00" to "99", assigned by Software AG.
ADAvrs.EMPL	The Employees demo file, containing dummy employee data you can use for testing Adabas. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.
ADAvrs.ERRN	Error messages for the Adabas Triggers and Stored Procedures Facility. These messages can be viewed using the Natural SYSERR utility. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.
ADAvrs.INPL	The code for Adabas Online System, Adabas Caching Facility, Triggers and Stored Procedures Facility, and various add-on demo products. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.
ADAvrs.JOBS	The sample job library for Adabas. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.
ADAvrs.MISC	The Miscellaneous demo file, containing dummy miscellaneous data you can use for testing Adabas. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.
ADAvrs.MOD	The load library for Adabas. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.
ADAvrs.PERL	The LOB demo file storing the LOB data referenced by the new Personnel demo file. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.

Library Name	Description
ADAvrs.PERS	<p>The Personnel demo file, containing dummy personnel data you can use for testing Adabas. This demo file includes fields that make use of the extended and expanded features of Adabas 8, include large object (LOB) fields. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.</p> <p>Note: The Personnel demo file must be installed on a UES-enabled database because it includes wide-character format (W) fields.</p>
ADAvrs.SRC	<p>The source library for Adabas. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.</p> <p>For a complete list of the time zones supported by Adabas in any given release, refer to the TZINFO member in this Adabas library.</p>
ADAvrs.TZ00	<p>The time zone library for Adabas. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas. Adabas bases its time zone library on the time zones defined in the public domain tz database, also known as the <i>zoneinfo</i> or <i>Olson</i> database.</p> <p>For a complete list of the time zones supported by Adabas in any given release, refer to the TZINFO member in the Adabas source library (ADAvrs.SRC).</p>
ADAvrs.VEHI	<p>The Vehicles demo file, containing dummy vehicle data you can use for testing Adabas. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.</p>
APSVrs.L016	<p>A Software AG internal library. The <i>vrs</i> in the library name represents the <i>version</i> of the internal library code, which is not necessarily the same as the version of Adabas.</p>
APSVrs.PAMS	<p>A Software AG internal library. The <i>vrs</i> in the library name represents the <i>version</i> of the internal library code, which is not necessarily the same as the version of Adabas.</p>
MLCVrs.JOBS	<p>The sample job library for Software AG's common mainframe license check software. The <i>vrs</i> in the library name represents the <i>version</i> of the license check software, which is not necessarily the same as the version of Adabas.</p>
MLCVrs.MOD	<p>The load library for Software AG's common mainframe license check software. The <i>vrs</i> in the library name represents the <i>version</i> of the license check software, which is not necessarily the same as the version of Adabas.</p>
WALvrs.MOD	<p>The load library for Adabas components shared by Adabas and other Software AG products, such as Entire Net-Work. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.</p>

Library Name	Description
WALvrs.SRC	The source library for Adabas components shared by Adabas and other Software AG products, such as Entire Net-Work. The <i>vrs</i> in the library name represents the <i>version</i> of Adabas.

Preparing to Install Adabas

This section provides information related to activities required prior to Adabas installation.

- Defining a BS2000 Logon ID
- Libraries Required for UES Support
- Adabas Library Disk Space Requirements
- Disk Space Requirements for the Database
- Adabas Nucleus Address Space Requirements
- Disk Space Requirements for Internal Product Libraries
- Using DAB-Supported Volumes
- Migrating an Existing Database
- Removing Version Numbers from Time Zone Library Elements

Defining a BS2000 Logon ID

Before you install Adabas in a BS2000 system, you must:

- define a BS2000 logon ID for Adabas that permits loading from a release or installation tape;
- enable all privileges for the logon ID to allow optional operations. This includes such capabilities as priority, T/P class, and maximum virtual memory size.

When allocating direct access files, private volumes are preferred to avoid the space fragmentation that can occur with public volumes.

Libraries Required for UES Support

The Software AG internal product libraries (APS - operating system layer) are required if you intend to enable a database for universal encoding service (UES) support. For UES support, the following library must be loaded and included in the BLSLIB concatenation: APSvrs.LIB (internal product application server), updated with the last patch level (APSvrs.Lnnn).

Note:

Newer APS module libraries are updates of the original. We advise you to copy the original, merge the new library into it, and reference the merged library in your JCL. The new merged library must be provided twice, with the link-name APSLIB and with the link-name BLSLIB.

Also for UES support, the following library must be loaded and included with the link name DDEC SOJ in the session execution JCL: ADA vrs .EC00, where vrs is the *version* of Adabas (for example, "812").

Adabas Library Disk Space Requirements

The Adabas files and libraries require the following disk space in PAM pages where vrs is the Adabas *version*:

File Name	Size	Description
ADA vrs .MOD	3468	Module library
ADA vrs .SRC	768	Adabas source, macros, and example jobs
WAL vrs .SRC	192	Adabas limited source, macros, and example jobs
WAL vrs .MOD	384	BS2000-dependent modules for Adabas limited
ADA vrs .EC nn	6432	UES code pages
ADA vrs .TZ00	672	Time zone elements

Disk Space Requirements for the Database

The Adabas database size is based on user requirements. For more information, refer to the *Adabas DBA Tasks* documentation. The following are suggested 2002 device cylinder and PAM page sizes for an initial Adabas database, allowing for limited loading of user files and the installation of Natural:

Database Component	Cylinders	PAM Pages
ASSOR1 (Associator)	50	7700
DATAR1 (Data Storage)	250	38000
WORKR1 (Work space)	50	7700
TEMPR1 (temporary work space)	25	3850
SORTR1 (sort work space)	25	3850

Adabas Nucleus Address Space Requirements

The typical Adabas nucleus requires at least 800-1024 KB of storage to operate. The size of the nucleus partition address space may have to be larger, depending on the ADARUN parameter settings. The address space parameters for the BS2000 user ID's JOIN entry must also be adequate. Parameter settings are determined by the user.

Disk Space Requirements for Internal Product Libraries

The minimum disk space requirements on the disk for the internal product libraries delivered with this version of Adabas are as follows:

Library	PAM Pages
ADAvrs.ECnn	6432
APS272.LIB	2304

Using DAB-Supported Volumes

The following restrictions apply to Adabas components when located on DAB-supported volumes:

Component	Recommendations/Restrictions
ASSO	not recommended for read caching; prohibited for write caching
DATA	not recommended for read caching; prohibited for write caching
WORK	no restrictions for read caching; prohibited for write caching
CLOG/PLOG/RLOG	no restrictions for read caching; prohibited for write caching
SORT/TEMP	no restrictions
Sequential input	no restrictions
Sequential output	no restrictions

Migrating an Existing Database

Use the ADACNV utility to migrate existing databases to new releases of Adabas. See *Adabas Utilities* for more information.

Removing Version Numbers from Time Zone Library Elements

The time zone library for Adabas has been delivered containing elements with version numbers. Attempts to read these will result in the following error message:

```
ADANTZ .... Error Opening DD TZINFO(....)
```

To avoid this, remove the version numbers by issuing the following BS2000 commands:

```
MOD-FILE-ATTR ADAvrs.TZ00,AC=W
STA-LMS
O ADAvrs.TZ00,U
M-E-AT ( , * , * ),N=(V=*U)
END
MOD-FILE-ATTR ADAvrs.TZ00,AC=R
```

Installing the Adabas Release Tape

Adabas release tapes available to Software AG affiliates contain all Adabas product options. The affiliates use these release tapes to create custom installation tapes for customers according to contract agreements.

For specific information about your particular release tape, refer to the *Installation Notes* delivered with the installation tape.

- Installation Using SMA
- Installation Not Using SMA

Installation Using SMA

If you are installing Adabas using the Software AG System Maintenance Aid (SMA) , refer to the *System Maintenance Aid* documentation and to the information provided with the installation tape for specific installation instructions.

Installation Not Using SMA

Note:

If you are using System Maintenance Aid (SMA), refer to the *System Maintenance Aid* documentation. If you are not using SMA, follow the instructions below.

This section explains how to copy all files from tape to disk. You will then need to perform the individual installation procedure for each component to be installed.

- Step 1: Copy the Library SRVvrs.LIB from Tape to Disk
- Step 2: Copy the Procedure COPY.PROC from Tape to Disk
- Step 3: Copy all Product Files from Tape to Disk

Step 1: Copy the Library SRVvrs.LIB from Tape to Disk

This step is not necessary if you have already copied the library SRVvrs.LIB from another Software AG installation tape. For further information, refer to the element #READ-ME in this library. The library SRVvrs.LIB is stored on the tape as a sequential file named SRVvrs.LIBS containing LMS commands. The current version vrs can be obtained from the *Report of Tape Creation*. To convert this sequential file into an LMS library, execute the following commands:

```
/IMPORT-FILE  SUPPORT=*TAPE(FILE-NAME=SRVvrs.LIBS,-
/  VOLUME=volser, DEV-TYPE=tape-device)
/ADD-FILE-LINK LINK-NAME=EDTSAM, FILE-NAME=SRVvrs.LIBS,-
/  SUPPORT=*TAPE(FILE-SEQ=3), ACC-METH=*BY-CAT,-
/  BUF-LEN=*BY-CAT, REC-FORM=*BY-CAT, REC-SIZE=*BY-CAT
/START-EDT
@READ  '/'
@SYSTEM 'REMOVE-FILE-LINK  EDTSAM'
@SYSTEM 'EXPORT-FILE  FILE-NAME=SRVvrs.LIBS'
@WRITE  'SRVvrs.LIBS'
@HALT
/ASS-SYSDTA  SRVvrs.LIBS
/MOD-JOB-SW  ON=1
/START-PROG  $LMS
/MOD-JOB-SW  OFF=1
/ASS-SYSDTA  *PRIMARY
```


where:

tape-device is the device type of the tape, for example: TAPE-C4
volser is the VOLSER of the tape (see *Report of Tape Creation*)

Step 2: Copy the Procedure COPY.PROC from Tape to Disk

To copy the procedure COPY . PROC to disk, call the procedure P . COPYTAPE in the library SRVvrs . LIB:

```
/CALL-PROCEDURE (SRVvrs.LIB,P.COPYTAPE), -  
/ (VSNT=volser, DEVT=tape-device)
```

If you use a TAPE-C4 device, you may omit the parameter DEVT.

Step 3: Copy all Product Files from Tape to Disk

To copy all Software AG product files from tape to disk, enter the procedure COPY . PROC:

```
/ENTER-PROCEDURE COPY.PROC, DEVT=tape-device
```

If you use a TAPE-C4 device, you may omit the parameter DEVT. The result of this procedure is written to the file L . REPORT . SRV.

Installing an Adabas Database

Once you have installed the Adabas installation tape, 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 the *Adabas Utilities*.

- Migrate an Existing Database
- Database Installation Steps

Migrate an Existing Database

Use the ADACNV utility to migrate existing databases to new releases of Adabas (SMA job number I021). See *Adabas Utilities* for more information.

Database Installation Steps

Step 1: Check, Prepare, and Install the Product License File (SMA job number I010)

You must install a valid license file on all mainframe platforms in which your Software AG mainframe product is installed. The license file is provided as an XML document (encoding is US-ASCII) and must remain in that format -- even on the mainframe. It must not be modified. Any modification of the license file will invalidate the digital signature and the license check will fail. In the event of a check failure, please contact your Software AG technical support representative.

Note:

Thirty days before the license expires, license check failure messages are produced. Your software product will still function, but these messages warn you that it is time to obtain a new license.

In this step, you will prepare the license file (obtain it from e-mail or the installation tape and store it on your BS2000 system) and then install it:

- Preparing the Product License File
- Installing the Product License File

Preparing the Product License File

The product license file is supplied on the individual customer installation tape or separately via an e-mail attachment. Before you can install the license, you must transfer it from e-mail or the installation tape and store it on a BS2000 system. This section describes how to do this for a license distributed either by e-mail or on the installation tape.

▶ To prepare the license file from an e-mail attachment, complete the following steps:

1. Transfer the license to BS2000, as described in *Transferring a License File from PC to a BS2000 Host Using FTP*.
2. Verify that the transferred license file is stored in an Adabas source library (with RECFM=F or FB and LRECL=80), taking care to preserve its format as ASCII.

▶ To prepare the license file from the installation tape, complete the following steps:

- Verify that the license file is stored from the tape into an Adabas source library (with RECFM=F or FB and LRECL=80), taking care to preserve its format as ASCII.

Installing the Product License File

Once the license file has been prepared, you can install it:

▶ To install the product license file once it has been transferred:

1. Run the LICMAKE license conversion job to load the license file into a library and assemble it.
2. Store the assembler output module ADALIC(R) in the job library and add this to your BLSLIB concatenation.

Note:

The Adabas 8 nucleus needs a license file to start up. Either the license file is assembled as an object by ADALIC(R) into a library in the DDLIB/BLSLIB chain or the license is read from the link name DDLIC. If neither of these is the case, the nucleus will end with a DMS0A4F error because it cannot find a link name on DDLIC.

Step 2: Allocate and Format the Adabas Database (SMA Job Number I030)

Customize and run the ADAFRM utility job to allocate and format the Adabas database. The following must be customized:

- 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 database ID and database device type(s);
- Sizes of the data sets for each ADAFRM statement.

Step 3: Define the Global Database Characteristics (SMA Job Number I030)

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 database ID and database device type(s);
- ADADEF parameters.

Step 4: Load the Demonstration Files (SMA Job Number I050)

Customize and run the job:

- ADALODE to load the sample demo file EMPL;
- ADALODV to load the sample demo file VEHI;
- ADALODM to load the sample demo file MISC; and
- ADALODP to load the sample demo file PERS and its associated LOB demo file, PERL.

Note:

The Personnel demo file must be installed on a UES-enabled database because it includes wide-character format (W) fields.

For each job, the following items must be customized:

- Data set names for the database and libraries;
- The Adabas database ID and database device type(s);
- ADALOD parameters.

Step 5: Load Adabas Online System Add-On Product, If Appropriate (SMA Job Number I061)

Customize and run the job ADAINPL to load the Adabas Online System (AOS) into a Natural system file using the appropriate batch version of Natural (read the Adabas Online System documentation to determine its Natural requirements). The following items must be customized:

- Data set names of the database and libraries;
- The Adabas database ID and device type(s);
- The Natural INPL parameters and system file number.

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

Step 6. 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 database ID and device type(s);
- ADASAV parameters.

Step 7: Insert the ADARUN Defaults

The member DEFAULTS in the Adabas job 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: 2300)
 - Database ID (default: 1)

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

Step 8: 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.

The Adabas BS2000 Communication Environment

The installation of a supervisor call (SVC) to provide a communication environment for Adabas is not required on BS2000 systems. Adabas uses the BS2000 executive services common memory pool and eventing for interprocess communication.

If the common memory pool should not be the default, set the ADARUN or DDLNKPAR IDTNAME parameter or reassemble SSFB2C after setting the parameter IDTNAME in the macro B2CONFIG.

The rules to determine which IDTNAME is used are as follows:

1. If it is a client program and the link name DDLNKPAR is present, the IDTNAME is taken from this file.

2. Otherwise (if the condition in #1 is not met), if the link name DDCARD is present, the IDTNAME is taken from this file.
3. Otherwise (if the conditions in #1 and 2 are not met), the IDTNAME is taken from module SSFB2C.

Note:

If the application is running in a subtask, for example, the Natural program in triggers and stored procedures, this file must be a permanent (not a temporary) file.

The router functions are implemented in the form of subroutines contained in the module ADARER. ADARER is loaded into common memory pool during nucleus initialization, and is shared by all user tasks that issue Adabas calls.

For those systems running an earlier Adabas/WAL library but that wish to run with the Version 8.2 environment, the Version 8.2 system must be started first in order to create the version 8.2 router (ADARER) required for this version's nucleus.

Considerations of Device Types

The new default device type is 2300. All devices have standard block sizes that are a multiple of 4 and rounded to a 2K boundary. Thus they can be used on NK4 disks and D3475-8F devices.

If you do not wish to migrate the default device used by prior releases of Adabas (device 2000) to the new default (device 2300), then the JCL will need to include `DEVICE=2000` in the ADARUN cards.

Devices and Spanned Records

Spanned record files are those loaded with the ADALOD SPAN option.

Writing a record over a block boundary in such a file requires an extra 50 to 100 bytes in the Work file block. In such cases, the Work file needs to have a block size of 50 to 100 bytes larger than the Data block size.

The default device 2300 has Data and Work block sizes of 8192 bytes. Therefore, there are not enough spare bytes in default device 2301 to span records over blocks. Device 2301, however, has Data and Work block sizes of 16384 bytes, so this device can be used.

If the default device must process spanned records, the Work file block should be formatted, for example, as a 2301 device at block size 16384, and the ADADEF database define should set `WORKDEV=2301`. A zap file will be made available at a later time to address this workaround.

Applying Zaps

Every effort has been made to make Adabas operating system independent. However, Adabas development is done in an IBM environment and corrections and changes are prepared in this environment.

As a result, corrections are applied to Adabas modules with IMASPZAP, which determines the syntax of any zaps. When corrections must be applied to a module library at BS2000 locations, the zaps must be revised to the LMS format. The following example shows how this can be done:

▶ **To revise a zap to LMS format:**

1. Assuming a zap in the following LMS format:

```
/EXEC LMS
LIB ADAvrs.MOD,BOTH
UPDR ADAEXAMP (a)
*COR ADAEXAMP,12BC,X'47B0A123'=X'4720BAEE' (b,c)
*END
END
```

where:

a specifies corrections to module ADAEXAMP.

b verifies the contents at location X'12BC'. The correction is applied only if the content of this location is X'47B0A123'.

c specifies a replacement value X'4720BAEE' for the location X'12BC'.

2. To verify the original data at location X'12BC' on the BS2000 system, perform the following:

```
/LOAD (ADAEXAMP,ADAvrs.MOD) Load module
/DISPLAY L'12BC'.(L=4) Display original content
%D C=ADAEXAMP.H'12BC'%XL4 (AID)
```

If the data displayed is different from the value given on the original VER statement, do not continue with step 3.

3. Run the LMS procedure.
4. To ensure that the last step was performed properly, perform step 2 again. This procedure should also be used for ADARUN and ADALNK zaps, described later in this section.

Job Variable (JV) Handling

The control job variable (JV) is filled automatically if the statement

```
/DEL-JV name
/SET-JOB-STEP
/CRE-JV name
/SET-JV-LINK *ADA,name
```

—is contained in the nucleus startup job control statements. The JV layout is as follows:

Positions	Length	Representation	Contents
1 - 128	128	undefined	unused
129 - 136	8	alphanumeric	program name
137 - 140	4	numeric	user abend code
141 - 145	5	numeric	error number (parameter or utility)
146 - 150	5	numeric	Adabas response code
151 - 157	7	numeric	CPU time, program-related (sec.)
158 - 158	1	alphanumeric	blank
159 - 165	7	numeric	elapsed time, program-related (sec.)
166 - 178	13	alphanumeric	blank, reserved

Job Switches

Adabas uses job switch 10. The job switch is set by ADARUN and reset if the nucleus or utility session terminates normally. It can be used for session control to indicate whether or not a termination is normal. When ADARUN is called with PROGRAM=USER, no switches are set or reset.

Linking User Exits B and A to ADALNK

One or two user exits may be linked to or loaded with ADALNK:

- UEXITB (pre-command) receives control before a command is passed to a target by 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 (post-command) receives control after a command has been completely processed by a target, the router, or by ADALNK itself.

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

Register	Content
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).
13	Address of an 18-word save area
14	Return address
15	Entry point address: UEXITA or UEXITB

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.

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 (ADARSP216) 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.

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 ADANUC user exits.

Important:

If an ADALNK batch link routine has been modified to accommodate the needs of an Adabas product extension (such as Adabas Review), it should not be used for the Adabas nucleus or Adabas utility jobs.

ADALNK Optimizer Exit OPTEXIT

An exit in ADALNK for the Natural Optimizer is provided in BS2000 environments. The exit is called before the UEXITA (user exit A). The source module OPTEXIT(S) is supplied in the ADA_{vr}s.SRC library and is already bound to the supplied ADALNK. If it is to be changed, it will have to be replaced.

Enabling Universal Encoding Support (UES) for Your Adabas Nucleus

Prior to Adabas Version 7, Entire Net-Work converted all data for mainframe Adabas when necessary from ASCII to EBCDIC. Starting with Version 7, Adabas is delivered with its own data conversion capability called universal encoding support (UES). Entire Net-Work detects when it is connected to a target database that converts data and passes the data through to Adabas without converting it.

For Adabas Version 8.2, UES is enabled by default for the link routine ADALNK.

Note:

The use of UES-enabled link routines is transparent to applications, including applications that do not require UES translation support: it is not necessary to disable UES support.

- Load Module and Translation Tables
- Job Steps
- Calling LNKUES

Load Module and Translation Tables

No source module for ADALNK is delivered with this version of Adabas. The module ADALNK already has the UES component, LNKUES, and the translation tables ASC2EBC and EBC2ASC bound to it.

The ADALNK module consists of ADALNK, ADAL2P and LNKUES. You can bind modified ASC2EBC or EBC2ASC translation tables to ADALNK-BASE using the job ALNKUES.

The two standard translation tables are:

- ASC2EBC: ASCII to EBCDIC translation; and
- EBC2ASC: EBCDIC to ASCII translation.

It should only be necessary to modify these translation tables in the rare case when a country-specific character other than A-Z a-z 0-9 must be used in the Additions 1 (user ID) or Additions 3 field of the control block.

The Adabas translation table pair is provided in the section *Translation Tables*.

Job Steps

Job library member ALNKUES is an example job of how to link ADALNK with the UES components.

Calling LNKUES

LNKUES is called only on ADALNK requests (X'08'), (X'0C') with reply (X'10') or (X'1C') and reply (X'20') calls if the first byte of the communication ID contains X'01' and the second byte does not have the EBCDIC (X'04') bit set.

For requests, LNKUES receives control before UEXITB. For replies, LNKUES receives control after UEXITA.

- Required Environment
- Connection Possibilities
- JCL Required for UES Support

Required Environment

The Adabas database must be UES-enabled. See *Adabas DBA Tasks* and the ADACMP and ADADEF utilities in *Adabas Utilities* for more information.

Connection Possibilities

UES-enabled databases are connected to machines with different architectures through Software AG's internal product software (APS) or through Entire Net-Work (WCP).

Adabas SQL Gateway (ACE) clients may not be strictly EBCDIC in an environment where databases are connected through Software AG's internal product software (APS).

JCL Required for UES Support

The nucleus job statements for a UES nucleus require the following items:

- BLSLIB access to the APS module libraries is required, so these extras statements are required:

```
/SET-FILE-LINK APSSLIB,$SAG.APSvrs.LIB
/SET-FILE-LINK BLSLIBnn,$SAG.APSvrs.LIB
```

Note:

Newer APS module libraries are updates of the original. We advise you to copy the original, merge the new library into it, and reference the merged library in the JCL.

- The job needs to contain the procedure call below to access the DDEC SOJ object library:

```
/CALL-PROCEDURE ($SAG.APSvrs.LIB,LMSLINKLIB),(LNK-NAME=BLSLIBn1)
```

---where BLSLIB $n1$ is last in the BLSLIB sequence

- The DDEC SOJ SET-FILE-LINK statement should point to the ECS encoding objects \$SAG.ADAvrs.ECnn library.
- No special Software AG internal product software (APS) parameters need to be set.

The setting of extra options on the START-PROGRAM statement can be done as follows:

```
/START-PROGRAM ($SAG.ADAvrs.MOD,ADARUN),-
/ RUN-MODE=*ADVANCED(ALT-LIB=YES,LOAD-INF=*REF,UNRES=*DELAY,-
/ MESSAGE=*ERROR)
```

- An example of Adabas session job control for UES (BS2000) is supplied in the \$SAG.ADAvrs.SRC(ADANUCU,J) library element.

Note:

The Software AG internal product software (APS) batch system will create a subtask job with the same user attributes as the UES nucleus job. This job will be stopped automatically when the UES nucleus is stopped.

Linking LNKUES to ADALNK for Data Conversion

Adabas Version 7 was delivered with the module LNKUES for Universal Encoding Support (UES). LNKUES converts data in the Adabas buffers and byte-swaps, if necessary, depending on the data architecture of the caller. In Adabas version 7, LNKUES had to be linked to ADALNK; in version 8, LNKUES is fix bound to ADALNK.

Prior to Version 7, Entire Net-work converted all data for mainframe Adabas. When Entire Net-work Version 5.5 and above detects that it is connected to a target database that converts data, it passes the data through without converting it.

LNKUES is called only on ADALNK request (X'1C') and reply (X'20') calls if the first byte of the communication ID contains X'01' and the second byte does not have the EBCDIC (X'04') bit set.

- For requests, LNKUES receives control before UEXITB
- For replies, LNKUES receives control after UEXITA

By default, two translation tables are linked into LNKUES/ADALNK:

- ASC2EBC: ASCII to EBCDIC translation; and
- EBC2ASC: EBCDIC to ASCII translation

Note:

It should only be necessary to modify these translation tables in the rare case that some country-specific character other than "A-Z a-z 0-9" must be used in the Additions 1 (user ID) or Additions 3 field of the control block.

If you prefer to use the same translation tables that are used in Entire Net-Work:

- in ASC2EBC and EBC2ASC, change the COPY statements from UES2ASC and UES2EBC to NW2ASC and NW2EBC, respectively.
- reassemble the translation tables and relink LNKUES/ADALNK.

Both the Adabas and Entire Net-Work translation table pairs are provided in section *Translation Tables*. You may want to modify the translation tables or create your own translation table pair. Be sure to (re)assemble the translation tables and (re)link LNKUES/ADALNK.

```
/ASS-SYSDTA *SYSCMD
/STA-PROG $TSOLNK
MODULE ADALNK,LIB=USER.MOD,ELEM=ADALNK
NCAL
LINK SYMBOLS *KEEP
INCLUDE ADALNK,ADABAS.MOD
INCLUDE ASC2EBC,ADABAS.MOD
INCLUDE EBC2ASC,ADABAS.MOD
BIND
/ASS-SYSDTA *PRIM
```

The (re)linked ADALNK must be made available to Entire Net-Work. If you are calling Adabas Version 7 and you do not have the correct LNKUES/ADALNK module, Adabas produces unexpected results: response code 022 (ADARSP022), 253 (ADARSP253), etc.

Formatting New Adabas Libraries

By formatting a database container, the size of it is physically defined. The following rules apply:

1. The highest PAM Page of the database container must be less than the HIGH-US-PA allocation in the `SHOW-FILE-ATTRIBUTES container,ALL` command output. Otherwise, accessing those pages higher than that physical limit will result in an I/O error DMS0922. For more information, read *Device and File Considerations*.
2. The Associator (ASSO) must have a top RABN that is greater than 30. Data Storage (DATA) must have a top RABN that is greater than 2.
3. The SORT and TEMP containers need not be formatted *if* they are being used with the access mode OUTIN.

Loading Elements into the Time Zone Library (TZ00)

The time zone library is delivered with Adabas as `ADAvrs.TZ00` and contains many elements of type X. If you need to load new elements into this library, modify and run job `TZUPLOAD` provided in `ADAvrs.SRCE`. This job can be adapted to load the new time zone element into the library in the correct format.

The loaded time zone element should contain the default version number (*U). If it does not, the following error occurs:

```
ADANTZ .... Error Opening DD TZINFO(....)
```

To resolve this problem, read *Removing Version Numbers from Time Zone Library Elements*.

Starting Adabas

Adabas User Exit 2 for BS2000 permits two methods for specifying the `/ENTER-JOB` job:

- using a job variable containing the complete `/ENTER-JOB` job command;
- defaulting to an `ENTER-JOB RES.E.xLCO` startup command.

For more information, see the description of *User Exit 2* in the user exit and hyperdescriptor exit documentation.

Interpreting BS2000 Error Messages

When running Adabas, some BS2000 I/O error messages may indicate that a parameter is missing or is incorrect. These errors can be shown online by entering:

```
HELP-MESSAGE DMS <message-id>
```

The following table lists the message IDs that can occur and their explanations in an Adabas environment:

Message	Description
0922	The SAM file (for example, for DDOUT1) primary and secondary allocation must be a multiple of both the standard block size (16) and of the allocation unit (3). The smallest multiple allowed is 48. The standard block size may have been changed using the ADARUN parameter QBLKSIZE. If so, the standard block size is determined by dividing the sum of the QBLKSIZE value and 2047, by 2048.
0BBB	The application is using an illegal record length. The target file may need a larger block size.
0D99	Parallel access to a PAM file (for example, DATA) was attempted, but the PAM file has been opened by another application.
0DB1	Refer to the 0922 description above.
0DC2	SHARUPD=YES was specified for a SAM file (for example, DDSIBA).