Installation Procedure

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

Information for your specific installation is contained in the *BS2000 Product Installation Package* (order number SIA-111-016) distributed with the System Installation Aid.

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. Refer to the section *Adabas Library Disk Space Requirements*.
2	Allocate disk space for the Adabas database.	For better performance, distribute the database files over multiple devices and channels. Refer to the section *Disk Space Requirements for the Database*.
3	Specify the address space for running the Adabas nucleus.	Refer to the section *Adabas Nucleus Address Space Requirements*.
4	Allocate and format the Adabas database with the ADAFRM utility job (job I030, step 1000)	
5	Define the global database characteristics with the ADADEF utility job (job I030, step 1000)	
6	Start the Adabas nucleus and test the Adabas communications with the ADANUC job (job I040, step 1000)	
7	Load the demonstration files ADA*vrs*.EMPL/MISC/VEHI with the ADALOD utility (performed in job I050 steps 0001 to 0003)	
8	If appropriate, test Adabas address space communications by running ADAREP (job I050 step 9990)	
9	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 delivered with Adabas (job I061, step 0112).	
10	Terminate the Adabas nucleus with an ADAEND operator command	
11	Back up the database by running the ADASAV utility job	

Preparing to Install Adabas

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

- Defining a BS2000 Logon ID
- Datasets 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 Datasets
- Using DAB-Supported Volumes
- Migrating an Existing Database

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.

Datasets Required for UES Support

The Software AG internal product libraries (BTE - basic technologies; and APS - operating system layer) are required if you intend to enable a database for universal encoding service (UES) support. These libraries are now delivered separately from the product libraries. For UES support, the following libraries must be loaded and included in the BLSLIB concatenation:

- BTE421.LD*nn*
- APS271.LD*nn*

--- where *nn* is the load library level. The library with the higher level must precede those with lower numbers in the BLSLIB concatenation.

Also for UES support, the following library must be loaded and included in the session execution JCL:

- BTE421.ECSO

Adabas Library Disk Space Requirements

The Adabas files and libraries require the following disk space in PAM pages where *vrs* is the Adabas version/release/system maintenance (SM) level:

File Name	Size	Description
ADA*vrs*.MOD	1728	Module library
ADA*vrs*.SRC	480	Adabas source, macros, and example jobs
WAL*vrs*.SRC	288	Entire Net-Work source, macros, and example jobs
WAL*vrs*.MOD	384	BS2000-dependent modules for Entire Net-Work
AUT*vrs*.LIB	039	ADAUTM library (LMS) (optional)

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 2000 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	4000
DATAR1 (Data Storage)	250	20000
WORKR1 (Work space)	50	4000
TEMPR1 (temporary work space)	25	2000
SORTR1 (sort work space)	25	2000

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 Datasets

The minimum disk space requirements on the disk for the internal product libraries delivered with Adabas Version 7.4 are as follows:

Library	PAM Pages
BTE421.LD01	720
BTE421.ECSO	8802
APS271.LD06	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.

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

If you are not using SMA, copy the datasets from tape to disk using the procedure described below:

- Step 1: Copy the Library SRVnnn.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 SRV*nnn*.LIB from Tape to Disk

Note:
This step is not necessary if you have already copied the library SRV*nnn*.LIB from another Software AG tape. For more information, refer to the element #READ-ME in this library.

The library SRV*nnn*.LIB is stored on the tape as the sequential file SRV*nnn*.LIBS containing LMS commands. The current version *nnn* 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=SRVnnn.LIBS, -
/ VOLUME=<volser>, DEV-TYPE=<tape-device>)
/ADD-FILE-LINK LINK-NAME=EDTSAM, FILE-NAME=SRVnnn.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=SRVnnn.LIBS'
@WRITE 'SRVnnn.LIBS'
@HALT
/ASS-SYSDTA SRVnnn.LIBS
/MOD-JOB-SW ON=1
/START-PROG $LMS
/MOD-JOB-SW OFF=1
/ASS-SYSDTA *PRIMARY

<tape-device> = device type of the tape, for example, TAPE-C4
<volser> = VOLSER of tape (see Report of Tape Creation)
```

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

Call the procedure P.COPYTAPE in the library SRV*nnn*.LIB to copy the procedure COPY.PROC to disk:

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

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

Step 3: Copy all Product Files from Tape to Disk

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

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

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

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.

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.

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 start-up 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 A and B to ADALNK

One or two user exits may be linked 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: 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.

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.

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.

Connecting UES-Enabled Databases

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 7.4, 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
- Default or Customized Translation Tables
- Source Modules
- Job Steps
- Calling LNKUES

Load Module

There is only one load module for ADALNK on BS2000 and this has to be linked with LNKUES and the default translation tables. LNKUES converts data in the Adabas buffers and byte-swaps, if necessary, depending on the data architecture of the caller.

The two standard translation tables are:

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

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

Default or Customized Translation Tables

You may use the load modules with the default translation tables linked in, or you may prepare your own customized translation tables, re-assemble the tables, and link them with the LNKUES module that is delivered.

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.

The LNKUES module is functionally reentrant; however, it is not linked that way in the Adabas load library.

Source Modules

The ADALNK module has been coded to enable UES support and is accessible via weak external references and will be enabled if the LNKUES module is linked or bound to it.

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 Smarts, or through Entire Net-Work.

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

JCL Required for UES Support

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

- BLSLIB access to the BTE and APS module libraries, so an extra BLSLIB statement is required:

  ```
  /SET-FILE-LINK APSLIB,$SAG.APSvrs.LDnn
  /SET-FILE-LINK BLSLIBnn,$SAG.APSvrs.LDnn
  /SET-FILE-LINK BLSLIBn1,$SAG.BTEvrs.LDnn
  ```

- the job needs to contain the procedure call below to access the DDECSOJ object library:

  ```
  /CALL-PROCEDURE ($SAG.APSvrs.LIB,LMSLINKLIB),(LNK-NAME=BLSLIBn2)
  ```

  ---where BLSLIB*n*2 is last in the BLSLIB sequence

- no special Smarts 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)
  ```

- there is no need for the DDECSMS SET-FILE-LINK statement

- the DDECSOJ SET-FILE-LINK statement should point to the ECS encoding objects $SAG.BTE*vrs*.EC*nn* library

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

Note:
The Smarts 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 is delivered with the module LNKUES for Universal Encoding Support (UES). This module must be linked to ADALNK. LNKUES converts data in the Adabas buffers and byte-swaps, if necessary, depending on the data architecture of the caller.

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.
- re-assemble 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 LNKUES,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, 253, etc.

Formatting New Adabas Datasets

The following formatting is required when creating new datasets:

Dataset	Formatting Required
ASSO	the first track plus the first 30 RABN blocks
DATA	the first two tracks
WORK	the whole dataset
PLOG / CLOG / RLOG	the whole dataset
SORT / TEMP	no formatting required

Formatting SORT and TEMP

The SORT and TEMP files can optionally be formatted. If non-formatted files are used, the following two FILE commands must be issued:

```
/CREATE-FILE ADA99.SORT,PUB(SPACE=(4800))
/SET-FILE-LINK DDSORTR1,ADA99.SORT,BUFF-LEN=STD(2),OPEN-MODE=OUTIN
```

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.*x*LCO start-up command.

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

Interpreting BS2000 Error Messages

When running Adabas, some BS2000 messages may indicate that a parameter is missing or is incorrect. The following are the message IDs that can occur, and their explanations in an Adabas environment:

Message	Description
D922	SAM file allocation (for example, for DDOUT1) primary and secondary allocation must be a multiple of the both 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.
DD99	Parallel access to a PAM file (for example, DATA) was attempted, but SHARUPD=YES was omitted
DDB1	Refer to the D922 description above
DDC2	SHARUPD=YES was specified for a SAM file (for example, DDSIBA)