

Installing Adabas Transaction Manager for z/OS

- The Installation Tape
 - Installation Checklist
 - Installation Procedure
-

The Installation Tape

Review the *Report of Tape Creation* that accompanies the release package before restoring the release data to disk.

The installation tape contains the following datasets in the sequence indicated in the report:

Dataset	Contents
ATM vrs .JOBS	Installation jobs, utility jobs
ATM vrs .LOAD	Executable modules
ATM vrs .SYS5	ADALOD input for the ATM PRR file
ATM vrs .SYS6	ADALOD input for the ATM STJ file
ATM vrs .SYS7	ADALOD input for the user proxy log (UPL) file
ATM vrs .SYS8	ADALOD input for the ET data file
ATM vrs .INPL	Online Services application
ATM vrs .ERRN	Online Services messages

where vrs is the version, revision, and system maintenance level of the product. Use the comments in the jobs when modifying them to conform to site requirements.

Installation Checklist

The steps needed for a successful installation are as follows:

Note:

The checklist and Installation Procedure indicate how to install a new ATM environment. If you need to upgrade an environment from an earlier version of ATM, refer to *Upgrading from Earlier Versions of ATM* for important additional information..

Step	Description	Job Name
	Before proceeding with the installation, ensure that all necessary preparation has been completed as described in section Before You Install.	
1	Restore the libraries from the installation tape.	
2	Create the transaction manager's database.	ATMI030
3	Load the transaction manager files to the transaction manager's database.	ATMI050
4	INPL the Online Services application.	ATMI061
5	Define the default job client runtime controls for use by the transaction manager client proxy.	
6	Prepare the ATM transaction manager job.	ATMI200
7	If required, define an Adabas System Coordinator file for migrated transaction records.	
8	Enable databases for two-phase commit.	
9	Prepare the Adabas System Coordinator daemons to support ATM.	
10	Make Adabas Transaction Manager components available as required.	

Installation Procedure

Following is the general Adabas Transaction Manager installation procedure. The actual installation depends on your particular requirements and the specific contents of the release package provided by Software AG for your site. Information in the release package is intended for your system. If that information differs from the information in this section, use the release package information or contact Software AG technical support for assistance.

Step1: Copying the Tape Contents to Disk

If you are using System Maintenance Aid (SMA), refer to the SMA documentation (included on the current edition of the Natural documentation CD). If you are not using SMA, perform steps 1a, 1b and 1c as described in this section:

- Step 1a: Copy Data Set COPY.JOB from Tape to Disk
- Step 1b: Modify COPY.JOB
- Step 1c: Submit COPY.JOB

Note:

If the datasets for more than one product are delivered on the tape, the dataset COPY.JOB contains the JCL to unload the datasets for all delivered products from the tape to your disk. After that, you will have to perform the individual install procedure for each component.

Step 1a: Copy Data Set COPY.JOB from Tape to Disk

The data set COPY.JOB (label 2) contains the JCL to unload all other existing data sets from tape to disk. To unload COPY.JOB, use the following sample JCL:

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X
//* -----
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=<Tnnnnn>),
// LABEL=(2,SL)
//SYSUT2 DD DSN=<hilev>.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=<vvvvvv>,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

where:

<hilev> is a valid high level qualifier
 <Tnnnnn> is the tape number
 <vvvvvv> is the desired volser

Step 1b: Modify COPY.JOB

Modify the COPY.JOB to conform with your local naming conventions and set the disk space parameters before submitting this job:


- set HILEV to a valid high level qualifier
- set LOCATION to a storage location
- set EXPDT to a valid expiration date

Step 1c: Submit COPY.JOB

Submit COPY.JOB to unload all other data sets from the tape to your disk.

Step 2: Create the Transaction Manager Database (Job ATMI030) (SMA Job I030, Steps 4190,4191)

The transaction manager database holds recovery information and, depending on the setting of the TMETDATA parameter, ET data, but no other operational data. The remaining files are for occasional use for special purposes – to store diagnostic log information obtained by the transaction manager client proxy, and to hold details of transactions that were forcibly terminated. The size of this database depends mainly on the number of transactions that might be in progress at the same time, and the total amount of ET data, if any, that the ATM manager will need to store.

 to create the database:

- Use sample job ATMI030, modified to meet your requirements, to establish the database.

Initially, define a database with the following component sizes:

DATA: 250 cylinders
ASSO: 50 cylinders
WORK: 50 cylinders
TEMP: 20 cylinders
SORT: 20 cylinders

Enough Data Storage space in the recovery file must be allocated to accommodate the maximum number of concurrently active transactions. The following can be used as a guide for calculating the required Data Storage space:

x times y is the required number of Data Storage blocks,

where x is the maximum number of concurrent transactions/branches at the local transaction manager, and y is the average number of compressed recovery records (PRRs) per Data Storage block. Assume that a compressed PRR requires about 2300 bytes if the transaction has 2000 bytes (compressed) of ET data, and 300 bytes if there is no ET data. It is recommended to allow a safety margin of about 50 percent extra space.

Standard block sizes are adequate for most sites. However, if transactions which perform changes to many databases are to be used, it may be necessary to increase the Data Storage block size to accommodate large recovery records.

No ET data will be stored in this database's checkpoint file. The following parameters should be adequate for the checkpoint file.

```
MAXISN=10,000  
DSSIZE=100B  
NISIZE=20B  
UISIZE=8B
```

Use the normal reporting utilities to determine whether space shortages are likely to occur.

Step 3: Load the Transaction Manager Files into the Transaction Manager's Database (Job ATMI050) (SMA Job I050, Step 4190)

The recovery record file, suspect transaction file, proxy log file and ET data file have numbers 5, 6, 7 and 8, respectively. These file numbers are fixed. Since no application files should be loaded into this database, there should be no conflict.

If ET data is to be stored in the transaction manager's database (TMETDATA=ATM), define file 8 with MAXISN large enough to allow an ISN for each ETID that will ever be used in this Adabas Transaction Manager's local system. A decompressed ET data record occupies about 2 kilobytes. If you choose to store ET data in target databases (TMETDATA=TARGETS), file 8 will not be used. For more information, see TMETDATA parameter.

 **to load the transaction manager files into the transaction manager's database:**

- Use job ATMI050 to define the recovery record, suspect transaction, proxy log and ET data files in the ATM transaction manager's database.

The suggested sizes are adequate for the purposes of testing. You can use the normal reporting utilities to check whether your system needs larger files.

Step 4: Install the Online Services Application SYSATM (Job ATMI061) (SMA Job I061, Steps 4190,4193)

The Adabas Transaction Manager Online Services application is delivered on the installation tape as a NATUNLD dataset created under Natural. The application programs are stored in libraries SYSATM and SYSMT vrs where vrs is the version, revision, and system maintenance level of the product.

to install the Online Services application SYSATM:

1. Use a Natural utility to install the Online Services application and ERRLODUS to install its messages.
2. If Natural Security is installed, define the libraries SYSATM and SYSMT vrs (where vrs is the version you are installing, for example 812) and protect as required. You may define MENU as the startup transaction for SYSATM. DO NOT define a startup transaction for SYSMT vrs .

Step 5: Define the Default Client Runtime Controls for the Transaction Manager Client Proxy

The default client runtime controls for the transaction manager client proxy must be defined. For further information, see Online Services, Client Runtime Controls. See also the section Adabas Transaction Manager Components.

Note:

If you are upgrading from a previous version of ATM, you should first have installed Version 8.1 of Adabas System Coordinator. As part of this process, you should have defined a new, empty system file, to contain system data such as client runtime controls. When using SYSCOR for the first time with this new file, you would have been given the option of converting data, including ATM's client runtime controls, from a previous version of system file. Refer to the *Adabas System Coordinator* documentation for details. If you did not carry out this conversion, and you wish to use your existing client runtime controls for ATM, log on to SYSATM now, and you will be guided through the process of converting them.

Note:

If you are upgrading from ATM Version 7.4, you must modify each set of client runtime controls by providing a value for the Adabas System Coordinator Group Name.

to define the default client runtime controls for the transaction manager client proxy:

1. Start a Natural session and log on to the newly installed SYSATM application. Invoke the Menu function. If you have not defined LFILE 152 for your Natural session, a message will prompt you to define it now. Select Systems Settings, then option LFILE 152 Maintenance. Enter the Database ID and file number of the system file that you created when you installed the Adabas System Coordinator. Enter values for the remaining input fields, and confirm your definition. Now return to the main menu and select the Client Runtime Controls option. For a detailed description of the following step, refer to Online Services, Client Runtime Controls.

2. On the Client Runtime Controls screen, press PF10 to add a new default definition. Select an appropriate job type and press Enter. On the next screen, enter *DEFAULT as the job name, and press PF5. Now you can specify the default ATM runtime control settings for the selected job type. Make sure that you set the control Adabas Transaction Manager to ON. Set the SVC number that will be used by the Adabas Transaction Manager that you are installing. You must also set the name of the Adabas System Coordinator group in which your client jobs and TP systems will execute. The provided default values for the remaining client runtime controls should be satisfactory for the purpose of completing and verifying the installation. You can review and change them as required once you have completed and verified the installation.

Step 6: Prepare the ATM Transaction Manager Job (SMA Job I200, Step 4190)

Because the ATM transaction manager executes as a special kind of Adabas nucleus, its JCL is similar to that of a normal Adabas nucleus.

to prepare the ATM transaction manager job:

1. Make any necessary site-specific changes to ATMI200 so that it can be used as a procedure for a started task.
2. Ensure that any ADALNK module that can be loaded by the ATM job does not have the Adabas System Coordinator stub linked into it.
3. Specify ADARUN DTP=TM.
4. Specify ADARUN TMLOG=NEVER for the purpose of verifying the installation. Once the verification process has been completed, reconsider this parameter setting.
5. Specify ADARUN NONDES=YES. This is the default setting.

Set other ADARUN parameters as you would for a normal database. For initial testing, the default limits and pool and buffer sizes should be sufficient. You can use the sample parameters in the JOBS library, member TMPARMS, as a basis for the transaction manager's ADARUN parameters.

See the section Parameters for a description of the ADARUN parameters that affect Adabas Transaction Manager operation.

Note:

An Adabas Transaction Manager must be run in each operating system image that will participate in distributed transaction processing (DTP). If more than one Adabas SVC is used in the same system image, an Adabas Transaction Manager must be run under each Adabas SVC that will participate in DTP.

Step 7: Define an Adabas System Coordinator File

If you plan to use ATM in a multi-system dynamic transaction routing (DTR) environment, you might first need to define an Adabas System Coordinator file (SYSCO file) to hold migrated transaction records (MTRs) for the ATM managers that will execute alongside the Adabas System Coordinator daemons in the group. This file is needed if you plan to use a multi-system DTR environment in which a client who is not at global transaction status can be migrated from one system image to another (for example, a CICSplex cluster with the ATM client runtime control `TransactionModel` set to DYNAMIC).

For detailed instructions about creating and defining a SYSCO file, refer to the Adabas System Coordinator documentation. The following paragraphs provide information that you will need when creating the file and making the appropriate definition in SYSCOR.

▶ **to define a SYSCO file to contain migrated transaction records:**

1. Define a standard Adabas file to contain MTRs. For the Adabas Transaction Managers in a COR group, a single MTR will be stored for each global transaction whose owner is migrated from one system image to another while the transaction is open; the record will be deleted when the transaction is finally committed or backed out. The following ADALOD parameters should be sufficient for 1000 MTRs at the same time, assuming block sizes of about 4 kilobytes for Data Storage and about 2 kilobytes for Associator:

```
DSSIZE=50B
NISIZE=30B
UISIZE=5B
```

During normal use, you can use the standard Adabas reporting utility to check that your file is of adequate size.

2. In the online system, SYSCOR, of the Adabas System Coordinator, navigate to the list of defined System Coordinator groups. Locate the group associated with your Adabas Transaction Manager, mark it with 'F', and press Enter.

At the next screen, press PF10 to add a new SYSCO File definition. Enter the MTR file definition in the pop-up window, as follows, specifying a DBID and file number that are suitable for your installation:

07:57:00	Add File Definition	2006-04-19
		C11270M2
Group Name: CORATMGP		
File Name: ATMMTR	DB ID: 135	File Number: 175 SVC: 254
Description: ATM MIGRATED TRAN RECORDS		
Press PF5 to confirm		
Command ==>		
	PF1 Help	PF3 Exit PF5 Add

ATM's MTR file must be defined with SYSCO file name ATMMTR.

Note:

If you do not create a definition for the MTR file using SYSCOR, ATM will assume that you have determined that no MTR file is needed. Even so, an ATM transaction manager might issue a warning message if it finds no definition. If this happens, and you are sure that you do not need an MTR file, you can prevent the message from appearing simply by creating a dummy MTR file definition for the COR group, using SYSCOR.

Step 8: Enable Databases for Two-Phase Commit Processing

to enable databases for two-phase commit processing

1. Ensure that the parameter

```
ADARUN DTP=RM
```

is set in every database that participates in global transactions coordinated by Adabas Transaction Manager.

2. Add the Adabas System Coordinator load library and the Adabas Transaction Manager load library to the STEPLIB DD concatenation.
3. Ensure that the JCL contains a DD statement for DDWORKR4, and that it identifies a suitable WORK dataset.
4. Ensure that the WORK 4 dataset is allocated and formatted.

The WORK 4 dataset should be allocated and formatted in the same way as the main WORK dataset. In the case of a Cluster Services or Parallel Services database, the WORK 4 dataset is shared by all members of the cluster.

Important Note:

For details about the WORK4 dataset, including information about sizing, please refer to the Adabas documentation, under *Defining an Adabas Database*, which is part of *Database Maintenance Tasks* in the section *DBA Tasks*.

Step 9: Prepare the Adabas System Coordinator Daemons

The ATM load library should be made available to all SYSCO daemons which will run under the same Adabas SVC as an ATM transaction manager.

Note:

An application job or TP system that requires ATM services must execute alongside an ATM transaction manager and a SYSCO daemon, executing in the same system and under the same Adabas SVC or ID table.

The SYSCO daemons must be started with the parameter setting `PRODUCT=ATM`.

Step 10: Make Adabas Transaction Manager Components Available

The ATM load library should be made available:

- in the STEPLIB of batch jobs, TSO procedures, and JCL procedures for IMS TM systems that will be used by clients or applications that require Adabas Transaction Manager services;
- in either the DFHRPL or the STEPLIB of CICS systems that will be used by clients or applications that require Adabas Transaction Manager services. Software AG recommends that you use DFHRPL. If you are not using the autoinstall feature, see member DFHDUP in the JOBS library for samples of the required CICS program definitions. Please also see the *Adabas System Coordinator* documentation for details of that component's CICS requirements.

- in the COMPLIB of the JCL procedures for Complete systems that will be used by clients or applications that require Adabas Transaction Manager services.

Ensure that the Adabas System Coordinator stub has been linked into the Adabas link modules that will be executed in each of the above environments.

The ATM load library and the Adabas System Coordinator library should be made available:

- in the STEPLIB of Adabas Transaction Managers;
- in the STEPLIB of multiuser database nucleus jobs that will run with DTP=RM.

Note:

As in previous releases, there is a requirement to use an unmodified ADALNK in certain areas. With Version 8.1 this restriction has been reduced to unmodified ADALNK being needed only when running standalone Adabas utility jobs.