

# Installing And Using The Adabas Migration Tool

This section describes how to install and use the Adabas migration tool.

- Migration Tool Installation
  - Migration Tool Operation
  - Changing Link Module SVCs Dynamically
  - CICS Considerations
  - Migration Tool Defaults
  - Adabas Options
  - Generating a Link Module Migration Table
  - Migration Table DSECT
- 

## Migration Tool Installation

- Installation Data Sets
- Create Migration Table

### Installation Data Sets

This section describes the data sets used for the migration tool.

### Load Library

Migration tool members in the Adabas load library are listed in the following table:

<b>Member</b>	<b>Description</b>
MIGS01	Batch migration tool.  Rename to ADALNK and make available to any batch job that needs to use the migration tool.
MIGS02	Com-plete migration tool.  Rename to ADALCO and place in a COMPLIB library for any Com-plete job that needs to use the migration tool.
MIGS03	CICS command-level migration tool.  Link this with the CICS command-level stubs, name it Adabas (or whatever your CICS link module is called), and place it in the DFHRPL concatenation of any CICS job that needs to use the migration tool.
MIGS09	Batch migration tool for use with reentrant ADALNK.  Rename to ADALNK and make available to any batch job that uses a reentrant ADALNK and needs to use the migration tool.

**Note:**

There is no requirement to relink any of these modules. However, if you do, you must be sure to leave their attributes unchanged (REUS, AMODE31).

**Source Library**

Migration tool members in the Adabas source library are listed in the following table:

<b>Member</b>	<b>Description</b>
MIGCICIN	Sample CICS program to initialize the migration tool.
MIGLINK	Macro used to generate the migration table.
MIGT01	Sample migration table.

The source library may also contain other members such as README files or ZAPs.

**Jobs Library**

Migration tool members in the Adabas jobs library are listed in the following table:

Member	Description
MIGAFP32	Example zap to change the names of the modules loaded by Adabas Fastpath 3.2 (FASTENV, FASTNUC, FASTPRM)
MIGAVI63	Example zap to change the names of the modules loaded by Adabas Vista 6.3
MIGCIASM	Sample job to assemble and link MIGCICIN
MIGCICLK	Sample job to link MIGS03 with the CICS command-level stubs. This example creates a CICS load module called CICMIG.
MIGTASM	Sample assembly and link job for creating the migration table
MIGZAPCI	Zap to change defaults for the CICS migration tool (see the section <i>Migration Tool Defaults</i> ).
MIGZAPDE	Zap to change default database ID and buffer sizes (see the section <i>Migration Tool Defaults</i> ).
MIGZAPRS	Zap to change the handling of initialization contention under Com-plete and CICS (initialization must be single-threaded in those environments).  By default, the migration tool delays contending users for a second at a time until initialization has completed. You can use this zap to give contending users a bad response code instead.
MIGZAPSV	Zap to change the SVC number in standard batch link modules, depending on the SVC number specified in the ADARUN parameters

## Create Migration Table

A migration table defines

- all available link modules and the versions (if any) of Adabas Fastpath, Adabas Transaction Manager, and/or Adabas Vista linked with them (see the section *Adabas Options* for more information).
- the databases that are to be accessed using which link module. Any call to a database not defined in the table is issued using the default link module. You *must* nominate a default link module.

You can use the supplied sample migration table as a base. Ensure that the assembly terminates with condition code 0. Any error in parameter specification is reported in an MNOTE and generates condition code 16.

The linked migration table must be called:

- MIGT01 for batch;

- MIGT02 for Com-plete;
- MIGT03 for CICS; or
- MIGT09 for batch using a reentrant link module.

The migration table must not be linked reentrant.

- Define the Link Modules
- Example

## Define the Link Modules

Each link module named in the migration table

- must be available to the migration tool at runtime; and
- must have the correct SVC number specified in it.

## Example

You access your databases through an ADALNK Version 7.4.1 using SVC 249, but database 1 is still running at Version 6.2.3 accessed through an ADALNK Version 6.2.3 using SVC 248.

### To allow batch jobs to access all databases including database 1

1. Rename your Version 7.4.1 ADALNK to

```
A741LNK
```

2. Rename your Version 6.2.3 ADALNK to

```
A623LNK
```

3. Define both link modules in the migration table, defining A741LNK as the default and specifying that database 1 is to be accessed through A623LNK:

```
MIGLINK TYPE=LINK, LINK=A623LNK
MIGLINK TYPE=LINK, LINK=A741LNK, DEFAULT=YES
MIGLINK DBID=1, LINK=A623LNK
MIGLINK TYPE=END
END
```

## Migration Tool Operation

The loading environment is

- STEPLIB for batch;
- COMPLIB for Com-plete; and
- DFHRPL for CICS.

To use the migration tool, you need to have available in your loading environment:

- the appropriate migration tool:
  - MIGS01 renamed ADALNK for batch;
  - MIGS02 renamed ADALCO for Com-plete;
  - MIGS03 linked as Adabas (or the name of your choice) for CICS; or
  - MIGS09 renamed ADALNK for batch using a reentrant link module.

**Caution:**

Some products such as the Adabas nucleus and Com-plete (when running in ACCESS mode) require an unmodified ADALNK. Do not put the batch migration tool on the loading environment of any Software AG product that signs on to the Adabas SVC.

- the correct migration table:
  - MIGT01 for batch;
  - MIGT02 for Com-plete;
  - MIGT03 for CICS; or
  - MIGT09 for batch using a reentrant link module.
- all link modules named in the migration table.

When the first Adabas call is issued, the migration tool loads the migration table and all link modules named in it. If any load fails, the migration tool abends with a system 0C3 and the PSW pointing to one of the following error strings:

Error Message	Explanation
MIGT0x NOT AVAILABLE	The LOAD for MIGT0x failed
NO LINK MODULES DEFINED	No link modules are defined in the migration table
LINK MODULE NOT FOUND	The LOAD failed for one of the link modules defined in the migration table

When initialized, the migration tool examines every Adabas call to determine the target. If the target is defined in the migration table, the call is routed to the appropriate link module. Otherwise, the call is passed to the default link module.

Under CICS, the migration tool can also be initialized using the supplied MIGCICIN program. This may in fact be necessary if you want to force initialization during PLT processing without specifying PARMTYP=ALL in ADAGSET. MIGCICIN sends a special Adabas command IM to instruct the migration tool to initialize.

## Changing Link Module SVCs Dynamically

### Note:

This section applies to the standard batch migration tool (MIGS01) only.

When using the batch migration tool MIGS01, the SVC number assembled into ADALNK is not overwritten at execution time by the SVC number specified in the ADARUN parameters. Instead, you must assemble the correct SVC number into each copy of ADALNK.

This means, for example, that you cannot easily switch a batch job from a production to a test SVC. You would need to maintain two ADALNKs; one with the production SVC and another with the test SVC.

Depending on the SVC number specified in the ADARUN parameters, you may be able to avoid this situation by zapping translation rules into MIGS01 so that it will overwrite the SVC numbers in the link modules. The supplied job MIGZAPSV provides an example of how to do this.

An SVC translation rule is 16 bytes long. You may zap as many as 10 of these into MIGS01.

The format of each rule is:

Format	Description
00xx	ADARUN SVC to which this rule applies (2-bytes where the first byte must be X'00' and the second byte X'xx' is the relevant ADARUN SVC number in hexadecimal)
yyxx	2-byte pairs of SVC numbers in hexadecimal (up to 7 allowed per rule where the first byte X'yy' specifies the old and the second byte X'xx' specifies the new SVC number)

If the link module contains the first number, it is overwritten with the second number which is usually the same as the SVC to which the rule applies; that is, X'xx'.

The final pair must have X'00' as the first number. If the second number in the final pair is not X'00', this is treated as a default value which is applied to any link module that did not match one of the preceding pairs.

If there is no rule for the ADARUN SVC, no translation is done.

### Examples

Example 1:

You have 2 SVCs: 249 (Production) and 248 (Test), but all your batch ADALNKs contain SVC 249. To zap an SVC translation rule into MIGS01:

```
00F8 This rule will be in effect if ADARUN SVC=248
F9F8 Any ADALNK assembled with SVC 249 will be changed at execution time to
      use SVC 248 instead
0000 Any other ADALNK remains unchanged
```

Example 2:

You have 4 SVCs: 249 (old Production), 248 (old Test), 239 (new Production), 238 (new Test) and you want to ensure that all ADALNKs use the new versions of the SVCs. In addition, you want to use the test version if ADARUN SVC=248 or 238 and the production version if

ADARUN SVC=249 or 239. To zap SVC translation rules into MIGS01:

```
00F9 rule for ADARUN SVC=249
00EF make all ADALNKs use SVC 239
00000000 pad to next rule
00000000
00000000
00F8 rule for ADARUN SVC=248
00EE make all ADALNKs use SVC 238
00000000 pad to next rule
00000000
00000000
00EF rule for ADARUN SVC=239
00EF make all ADALNKs use SVC 239
00000000 pad to next rule
00000000
00000000
00EE rule for ADARUN SVC=238
00EE make all ADALNKs use SVC 238
00000000 pad to next rule
00000000
00000000
```

## CICS Considerations

This section provides information related to the use of the migration tool with CICS.

- PPT Entries
- LNKENAB and LNKTRUE
- Direct Call Interface
- Passing Parameters
- Supported Versions
- Newcopy

### PPT Entries

You must define a PPT entry for the migration tool itself, the migration table, and all link modules named in the migration table.

The entries for the migration tool and table should have the same options as for the link modules. The migration tool and table must both be defined as resident.

### LNKENAB and LNKTRUE

If you use the task-related user exit, you must have a LNKENAB and a LNKTRUE for each Adabas link module that operates in TRUE mode.

#### Note:

You may not define both TRUE and non-TRUE link modules in the same migration table.

In ADAGSET, specify the name of the actual link module and not the name of the migration tool.

### Example

In order to communicate through two SVCs, 248 and 249, you might define:

```
ADAEN48
ADATR48
ADABAS48
```

```
-with ADAGSET parameters
```

```
ENTPT=ADABAS48 , PARMTYP=ALL , SVCNO=248 , TRUENM=ADATR48...
```

```
-and
```

```
ADAEN49
ADATR49
ADABAS49
```

```
-with ADAGSET parameters
```

```
ENTPT=ADABAS49 , PARMTYP=ALL , SVCNO=249 , TRUENM=ADATR49...
```

## Direct Call Interface

MIGS03 supports callers using both the DCI and the EXEC CICS LINK interface to Adabas. MIGS03, however, always uses the DCI interface to call the various link modules.

## Passing Parameters

When MIGS03 is called using

- EXEC CICS LINK, it first looks for the Adabas parameter address list in COMMAREA and then in the transaction work area (TWA).

If neither storage area contains a parameter list, MIGS03 will abend with SOC3 and with the PSW pointing to the string

```
NO ADABAS PARAMETER LIST
```

- the DCI, R1 must contain the address of the Adabas parameter list.

## Supported Versions

The migration tool only works with command-level link modules from Adabas Version 6.2 and above.

## Newcopy

The migration tool records the entry-point addresses of all link modules named in the migration table.

This means that if you want to activate a new copy of an Adabas link module, you must also activate a new copy of the migration tool MIGS03 and the migration table MIGT03.



Similarly, if you wish to activate a new copy of MIGT03, you must also activate a new copy of the migration tool MIGS03.

For most sites, it is probably safer to recycle CICS.

## Migration Tool Defaults

The migration tool modules contain certain default values at fixed offsets, the same as an ordinary link module.

### Changing Non-CICS Module Defaults

For the non-CICS migration tools (MIGS01, MIGS02 and MIGS09), these defaults are shown in the following table:

Offset (Hex)	ADALNK Name	Default Value	Purpose
80	LNKLOGID	0	Default DBID
86	LUINFO	0	Length of user information
9C	RVINFO	0	Length of Adabas Review buffer
A0	XITBLEN	0	Length of buffer for user exit B
A2	XITALEN	0	Length of buffer for user exit A
A4	XITAFPLN	0	Length of Adabas Fastpath buffer

The migration tool uses LNKLOGID (if not 0) to select the link module for calls that have no database ID specified. If LNKLOGID is 0, these calls are given to the default link module.

The various length fields are not used by the migration tool itself. However, it is important to set them to the same values used in your current link modules because they are used by prefetch/multifetch to calculate buffer sizes and by Com-plete to suballocate its Adabas buffers. If you have different values in different link modules, use the highest value.

### Changing CICS Module Defaults

The offsets are different for CICS (MIGS03), as shown in the table below. The recommendations above also apply for CICS.

Offset (Hex)	ADAGSET Name	Default Value	Purpose
9DC	LOGID	0	Default DBID
9E2	LUINFO	0	Length of user information
A38	None	0070	Total length of a UB = (UBPFXL+UBLEN+LUINFO+LRINFO +LUSAVE+7) / 8 * 8
A42	LUSAVE	0	Length of user exit save area
A44	LRINFO	0	Length of Adabas Review buffer
A46	LXITBA	0	Length of buffer for user exit B
A48	LXITAA	0	Length of buffer for user exit A
A4A	LADAFP	0	Length of Adabas Fastpath buffer

## Adabas Options

The Adabas options Adabas Fastpath, Adabas Transaction Manager, Adabas Vista, and Adabas System Coordinator have components operating in the link module and are thus affected by the migration tool.

- Common Considerations
- Option-Specific Considerations

### Common Considerations

For these options:

- each version must have its own system file; and
- when the option's link module component loads other modules (for example, FASTENV, FASTNUC and FASTPRM in the case of Adabas Fastpath), it is necessary to zap older versions to change the names of the modules they load. Similarly, the older versions of the loaded modules must be renamed.

### Option-Specific Considerations

Individual options impose additional restrictions.

#### Adabas Fastpath

Each version of Adabas Fastpath must have a separate Asynchronous Buffer Manager.

#### Adabas Transaction Manager

All databases participating in a transaction must be accessible through the same link module.

## Adabas Vista

All partitions of a partitioned file must be accessible through the same link module.

## Adabas Online Services

Adabas Fastpath, Adabas Transaction Manager, Adabas Vista, and Adabas System Coordinator each have an online services system that are written in Natural.

Certain online functions communicate with the option's link module component by issuing Adabas calls that do not contain a meaningful DBID. The link module component recognizes such calls and satisfies them.

Because the DBID is not significant, the migration tool cannot route these commands to the appropriate link module. However, it can recognize that the commands are internal to an option and it can identify the relevant option and its version.

Having identified the option and its version, the migration tool searches the migration table for a link module that has been defined as containing the matching option version and forwards the call to that link module. If no match is found, the call is passed to the default link module.

### Note:

A link module may contain only one version of an option and an option version may reside in only one link module.

## Generating a Link Module Migration Table

A sample link module migration table is provided in the source library for use in generating a link module migration table.

### to generate a link module migration table:

#### 1. Specify the Link Modules

You may specify up to 10 different link modules to be used in this environment.

For each link module, you must provide the following:

```
MIGLINK TYPE=LINK, LINK=link-module-name
```

One of the link modules must be designated as the default link module by providing the DEFAULT=YES parameter:

```
MIGLINK TYPE=LINK, LINK=link-module-name, DEFAULT=YES
```

Additionally, when a link module has an Adabas option linked with it, you must indicate the option version. For example:

```
MIGLINK TYPE=LINK, LINK=link-module-name, AFP=vrs, AVI=vrs, ATM=vrs, COR=vrs
```

—for a link module containing Adabas Fastpath, Adabas Vista, Adabas Transaction Manager, and Adabas System Coordinator at the version, revision, and system maintenance (SM) levels specified by *vrs*. The same version of an option may not be specified for more than one link module.

**Example:**

There are four link modules: one per Adabas SVC 236, 237, 249, and 254.

The link module for databases on SVC 237 is the default and includes version 712 of Adabas Fastpath and Adabas Vista.

The link module for databases on SVC 254 includes old versions of Adabas Fastpath and Adabas Vista. Note that special zaps are required in order to run more than one version of Adabas Fastpath or Adabas Vista in a single job.

Databases on SVCs 236 and 249 use neither Adabas Fastpath nor Adabas Vista.

```
MIGLINK TYPE=LINK, LINK=ADALNK36
MIGLINK TYPE=LINK, LINK=ADALNK37, DEFAULT=YES, AFP=712, AVI=712
MIGLINK TYPE=LINK, LINK=ADALNK54, AFP=321, AVI=631
MIGLINK TYPE=LINK, LINK=ADALNK49
```

## 2. Specify the Databases and Other Targets

For each database that will use a link module other than the default, you must provide:

```
MIGLINK DBID=dbid [, LINK=link-module-name ]
```

You must specify at least one DBID entry. If you omit the LINK parameter, the database uses the default link module. Otherwise, it should match a previously defined link module.

You may specify up to 1000 DBIDs.

**Example:**

Define databases and other SVC-based targets, for example:

- Database 131 is accessed through ADALNK54, Adabas SVC 254;
- Database 122 is accessed through ADALNK49, Adabas SVC 249;
- Database 153 is accessed through ADALNK36, Adabas SVC 236;
- Calls to any other target are directed to the default link module.
- 

```
MIGLINK DBID=131, LINK=ADALNK54
MIGLINK DBID=122, LINK=ADALNK49
MIGLINK DBID=153, LINK=ADALNK36
```

## 3. Generate the Migration Table

To generate a migration table with optionally up to *nnn* spare DBID entries, specify

```
MIGLINK TYPE=END[ , SPAREDB=nnn ] [ , WTO={YES|NO} ]
```

Specify WTO=NO if the contents of the migration table are not to be written to the operator console at execution time. The default is WTO=YES.

**Example:**

Generate the table with 10 spare entries for zapping in emergency targets that should not use the default link module. Do not write the migration table contents to the operator console:

```
MIGLINK TYPE=END, SPAREDB=10, WTO=NO
END
```

**Migration Table DSECT**

The DSECT for the migration table is shown below:

Field	Description
MLTMLT DSECT	
MLTEYE DS CL8	
MLTALNK DS F	A(TABLE OF LINK MODULES)
MLTNLNK DS F	NUMBER OF LINK MODULES
MLTADBS DS F	A(TABLE OF DBIDS)
MLTNDBS DS F	NUMBER OF DATABASES
MLTAUSR DS F	A(TABLE OF USERIDS) (FUTURE)
MLTNUSR DS F	NUMBER OF USERIDS (FUTURE)
MLTRSVD DS XL2	RESERVED
MLTVRL DS XL2	VERSION
MLTWTO DS CL1	ISSUE WTOS OR NOT
DS XL27	RESERVED FOR FUTURE USE
MLTLNK DSECT	LINK MODULE TABLE
MLTLKNM DS CL8	NAME
MLTLKAD DS F	ADDRESS
MLTAFPV DS XL2	VERSION OF ADABAS FASTPATH
MLTAVIV DS XL2	VERSION OF ADABAS VISTA
MLTATMV DS XL2	VERSION OF ADABAS TRANS. MGR.
MLTAAFV DS XL2	VERSION OF ADASAF
MLTREVV DS XL2	VERSION OF REVIEW
MLTCORV DS XL2	VERSION OF SYSTEM COORDINATOR
MLTLKST DS XL1	LINK MODULE STATUS
MLTLKOK EQU 0	LINK MODULE CAN BE USED
MLTLXX1 DS XL5	SPARE
MLTLKX DS XL2	ENTRY NUMBER

<b>Field</b>	<b>Description</b>
MLTLLEN EQU *- MLTLKNM	LENGTH OF AN ENTRY
MLTDBS DSECT	DBID TABLE
MLTDBID DS XL2	DBID
MLTDLKX DS XL2	INDEX INTO LINK TABLE
MLTDLNK DS CL8	LINK MODULE FOR THIS DBID
MLTDXX1 DS XL20	SPARE
MLTDLEN EQU *- MLTDBID	LENGTH OF AN ENTRY
MLTUIDS DSECT	USERID TABLE (FOR FUTURE USE)
MLTUSER DS CL8	USERID
MLTULKX DS XL2	INDEX INTO LINK TABLE
MLTULNK DS CL8	LINK MODULE FOR THIS USER
MLTUXX1 DS XL14	SPARE
MLTULEN EQU *- MLTUIDS	LENGTH OF AN ENTRY