

Installing Adabas with TP Monitors

This chapter provides information needed to install Adabas in batch mode and with its teleprocessing (TP) monitors. It covers the following topics:

- Preparing Adabas Link Routines for IBM Platforms
 - Installing Adabas with IMS/TM under Adabas 8
 - General Considerations for Installing Adabas with CICS
 - Installing Adabas with CICS under Adabas 8
 - Installing Adabas with Com-plete under Adabas 8
 - General Considerations for Installing Adabas with Batch/TSO
 - Installing Adabas with Batch/TSO under Adabas 8
 - Modifying Source Member Defaults (LGBLSET Macro) in Version 8
-

Preparing Adabas Link Routines for IBM Platforms

This section describes the preparation of Adabas link routines for TP monitors for IBM platforms. The source modules for Adabas 8 link routines are not provided in the Adabas 8 base source library. The Adabas 8 link routines can only be tailored via zap or using a link globals table.

- Addressing Mode Assembly Directives in Adabas Link Routines

Addressing Mode Assembly Directives in Adabas Link Routines

All Adabas 8 link routines include AMODE and RMODE assembly directives. These assembly directives allow the linkage editor to produce warning messages when conflicting AMODE or RMODE linkage-editor control statements are encountered in the link JCL, JCS, or EXECs.

These assembly directives also serve to document the preferred AMODE and RMODE for each link routine. It is important to note that in and of themselves, these directives do not alter the actual addressing mode of the link routine during execution.

Re-linking Adabas 8 Link Routines

When re-linking the Adabas 8 link routines with certain AMODE and RMODE combinations, a warning message may be generated by the linkage editor. This may be safely ignored as long as it pertains to a conflict of AMODE or RMODE in the ESD record of one or more of the load modules that comprise the link routine, and as long as the resulting module has the proper AMODE and RMODE attributes for execution with the intended calling application programs.

Care must be taken to ensure that AMODE (24) applications will operate properly when invoking the link routine with the attributes chosen when it is re-linked. This is particularly important if the RMODE (ANY) attribute is associated with a link routine that will be loaded dynamically but invoked by a program that is AMODE (24). In this case, the link routine should be re-linked AMODE (31), RMODE (24) to avoid

addressing exception ABENDs because the AMODE (24) application cannot correctly invoke the link routine if it resides above the 16-megabyte line.

The Adabas 8 link routines all run AMODE (31) after initialization, but they will return to the caller in the caller's AMODE.

Note:

Under CICS, the version 8 links run AMODE (31), but the Dataloc RDO parameter governs the AMODE and RMODE of the running CICS transaction.

The batch/TSO non-reentrant link routine, ADALNK, has been assembled and linked with AMODE (31), RMODE (24), and that is the recommended configuration to support AMODE (24) or RMODE (24) application programs. It may be re-linked AMODE (31), RMODE (ANY) if desired, but this should only be done if it is certain that all calling programs are AMODE (31).

The ADALNKR batch TSO reentrant link routine has been assembled and link-edited with AMODE (31), RMODE (ANY). If it is loaded by an application that is AMODE (24), it should be re-linked AMODE (31), RMODE (24).

The z/OS Com-plete module ADALCO has been assembled and linked AMODE (31), RMODE (ANY). The Com-plete TP monitor ensures proper AMODE switching between AMODE (24) or RMODE (24) programs that invoke ADALCO through the Com-plete Adabas interface routine, TLOPADAB.

All of the V8 CICS link routine modules - ADACICS, ADACICT, ADACIC0 and ADACIRQ - have been assembled and link-edited AMODE (31), RMODE (ANY). CICS manages the loading of programs and their invocation depending on the DATALOC values associated with their program and transaction definitions.

The Adabas IMS interface link routine ADALNI has been assembled and link-edited AMODE (31), RMODE (ANY). This is the preferred configuration for modern IMS applications, but if there are still AMODE (24) IMS applications executing at your installation, ADALNI may be re-linked AMODE (31), RMODE (24).

ADAUSER AMODE/RMODE Considerations

Software AG recommends that all batch applications invoke Adabas calls through the ADAUSER module. This module is normally link-edited with the application program and it then loads the appropriate link routine as well as ADARUN and ADAIOR/ADAIOS. The source member has the AMODE and RMODE directives coded as AMODE 31, RMODE ANY. This is the most flexible configuration for assembling and linking ADAUSER with the widest variety of application programs. However, if ADAUSER is dynamically loaded, either the RMODE assembler directive should be changed to RMODE 24 before re-assembling it or the ADAUSER module should be re-linked AMODE (31), RMODE (24) to ensure that AMODE 24 application programs may invoke it properly below the 16-megabyte line.

Installing Adabas with IMS/TM under Adabas 8

This section describes installation of the Adabas link routine for the IMS/TM TP monitor with Adabas 8.

IMS requires an Adabas link routine if it is to communicate with Adabas databases. The Adabas Version 8 executable default link routine is delivered in member ADALNI of the AII*vrs*.LOAD library (where *vrs* is the number of the latest Adabas version delivered on the installation medium). If you want to modify this link routine, use member ADALNI8 to do so. ADALNI8 must be linked with a link globals module you prepare and with any link routine exits you require to create the final ADALNI load module that is loaded

by the IMS message processing program (MPP) regions when an application calls them. Members ADALNI and ADALNI8 are provided with some default settings.

This section covers the following topics:

- IMS/TM Link Routines for Adabas 8
- Obtaining the Adabas User ID
- Obtaining the SAF ID
- Installation Procedure under Adabas 8

IMS/TM Link Routines for Adabas 8

These are Adabas 8 link routines for IMS/TM:

- ADALNI is the executable default module for message processing programs (MPPs). If you require no changes to the defaults provided in the link routine, use this module.
- Use ADALNI8 as the base module for message processing programs (MPPs). If you need to tailor ADALNI for your installation, use ADALNI8 to generate an updated ADALNI.
- ADALNK is the batch Adabas link routine for batch message processing (BMP) programs, batch-oriented BMP programs, and batch processing programs (DLIBATCH).

ADALNI and ADALNK use the CSECT name and ENTRY directive ADABAS by default.

The Adabas Version 8 ADALNI and ADALNK are UES-enabled as distributed.

This section describes using ADALNI and ADALNI8 only. For information on using ADALNK, read *General Considerations for Installing Adabas with Batch/TSO*.

Obtaining the Adabas User ID

The Adabas user ID is obtained at execution time by the ADALNI load module from the LTERM field (first eight bytes) of the IOPCB. The user ID is stored in the Adabas user block field UBUID and will be used for the last eight bytes of the Adabas communication ID.

Obtaining the SAF ID

The SAF ID is supported for use by Adabas SAF Security (ADASAF) if an external security package such as IBM's RACF or CA's ACF2 is present. The SAF ID is obtained at execution time by the ADALNI load module from the user ID field (bytes 33-40) in the IOPCB. To get a valid SAF user ID, SAF sign-on must be active in your IMS installation and the user must have performed an IMS /SIGN command to log onto an IMS terminal.

Installation Procedure under Adabas 8

➤ To modify the default settings and prepare the Adabas 8 link routine for IMS

1. Copy the sample member LNIGBL provided in the Adabas 8 AIIvrs.SRCE library to any appropriate user source library where they can be modified. These modules contain LGBLSET parameters that are used to create default settings for link components. A complete description of LGBLSET parameters can be found in *Modifying Source Member Defaults (LGBLSET Macro) in Version 8*.
2. Modify the LNIGBL member in the user source library.

Note:

The OPSYS parameter must be set to ZOS.

3. Modify and run sample job ASMGBLS as described at the top of the job. ASMGBLS can be found in the Adabas 8 ADAvrs.JOBS library. When fully modified, the SET statement in the job should reference the LNIGBL member you prepared in the previous step and the NAME link edit control statement should reference the name specified by the GBLNAME parameter in the LNIGBL member.

Once modified, submit the ASMGBLS job to assemble and link-edit the link globals module.

A new link globals module (with the name specified by the GBLNAME parameter in LNIGBL) will be generated in the user load library identified in the ASMGBLS job.

4. Copy sample job LNKLN18 to a user source library and modify it to link the new link globals module you created in the previous step and any required exits with the ADALN18 base module. Instructions for modifying the sample job are described at the top of the job. Be sure to direct the output from this job to an appropriate user load library. LNKLN18 can be found in the Adabas 8 AIIvrs.SRCE library.

The module resulting from this job is ADALN1.

5. Place the ADALN1 module in a load library available for IMS MPP regions.

The Adabas 8 link routine is prepared.

General Considerations for Installing Adabas with CICS

The macro-level link routine ADALNC is no longer supported for all levels of CICS running under z/OS. These environments must run a current version of Adabas and use the supplied command-level link component.

The Adabas command-level link routine supports the CICS transaction server (CTS) environment.

Notes:

1. The OPID option for the USERID field is not supported under CICS/TS 1.1 and above; therefore, it is not provided with the command-level link routine.
2. The CICS components from Adabas 7.4 or later are required when running with an Adabas 8 SVC.

The following sections describe specific points of Adabas/CICS installation and operation from the CICS perspective:

- Adabas Bridge for VSAM Considerations

- CICS MRO Environment Requirements
- Using CICS Storage Protection
- Sample Resource Definitions

Adabas Bridge for VSAM Considerations

If you are running Adabas Bridge for VSAM 4.2 or 5.1 under CICS, you must run CICS 3.3 or above and the Adabas Version 7.1 or above command-level link routine.

CICS MRO Environment Requirements

If you run the Adabas CICS command-level link routine with the CICS multiple region option (MRO), you must set the MRO parameter to "YES" and use the default for the NETOPT parameter. In an Adabas 8 installation, these parameters are supplied via the LGBLSET macro (read *Modifying Source Member Defaults (LGBLSET Macro) in Version 8*).

You can use the LGBLSET NTGPID parameter to provide a 4-byte literal for the Adabas communication ID to be used by the Adabas SVC when applications that call Adabas span multiple application regions.

Alternatively, you can create a user exit for the link routine that:

- sets UBFLAG1 (byte X'29' in the UB DSECT) to a value of X'08' (UBF1IMSR); and
- places a 4-byte alphanumeric value in the UB field UBIMSID.

This exit is link user exit 1 (LUEXIT1). The exit then allows the Adabas SVC to provide a proper Adabas communication ID in the Adabas command queue element (CQE) even when transactions originate in multiple regions.

Using CICS Storage Protection

The storage protection mechanism (STGPROT) was introduced under CICS/ESA 3.3. Storage protection permits resources to access either CICS or user storage by using the storage protection keys.

- User keys may not overwrite CICS storage, thus affording a degree of protection to CICS.
- CICS keys may read or write either CICS or user key storage, affording the highest degree of access to CICS resources.

Transaction isolation is an extension of the storage protection mechanism. It further protects CICS resources by isolating them in subspaces. This protects user key resources from one another, and protects CICS key resources from the CICS kernel. Transaction isolation can be enabled globally through the CICS TRANISO system initialization (SIT) parameter, and for each CICS transaction with the new resource definition ISOLATE keyword. Transaction isolation places some restrictions on CICS resources that must be available both during the life of the CICS system and to all transactions running in the CICS system.

In Adabas 8 installations, the CICS link routine always uses a task-related user exit, module ADACICT, so storage isolation is supported by default.

Sample Resource Definitions

Under CICS/TS 1.1 and above for z/OS and VSE, the preferred method for defining and installing CICS programs and transactions is RDO (resource definition online). The CICS documentation no longer recommends the assembly of PPT and PCT entries to define resources.

Modify and use the sample DEFINE statements located in member DEFADA8 as input to the IBM DFHCSDUP utility to define the Adabas CICS command-level components. Consult the appropriate IBM CICS documentation for information on the DFHCSDUP utility. The DEFADA8 member can be found in the Adabas 8 CICS command-level source library (ACIvrn.SRCE).

Installing Adabas with CICS under Adabas 8

- Supplied Modules
- Installation Procedure Under Adabas 8
- Preparing DDLINK Input for CICS

Supplied Modules

The following table lists the modules supplied in your Adabas installation to support the installation of Adabas with CICS under Adabas 8.

Module	Description
ADACICS	CICS command-level module.
ADACICT	CICS task-related user exit (TRUE) module.

Installation Procedure Under Adabas 8

➤ To install the Adabas 8 CICS link routine components, complete the following steps

1. Copy the Adabas 8 CICS load modules from the Adabas distribution library to a load library that will be in the CICS DFHRPL concatenation (see sample member CPYCICSM in the Adabas 8 ADAvrn.JOBS library).
2. Modify the sample CICSGBL member found in the Adabas 8 ACIvrn.SRCE library. This member contains sample default installation (LGBLSET) parameter settings. For more information about what to modify in this member, read *Modifying Source Member Defaults (LGBLSET Macro) in Version 8*.

Note:

The OPSYS parameter must be set to ZOS.

3. Save the modified CICSGBL member with a unique name in an appropriate user source library.
4. Modify and run sample job ASMGBLS as described at the top of the job. ASMGBLS can be found in the Adabas 8 ADAvrs.JOBS library. When fully modified, the SET statement in the job should reference the CICSGBL member you prepared in the previous step and the NAME link edit control statement should reference the name specified by the GBLNAME parameter in the CICSGBL

member.

5. Review and run the LNKGCICS member in the ACIvrn.SRCE library to link the newly assembled globals table from the previous step with any user or Software AG product exits. (For information about specific Software AG product exits, read the installation documentation for the product.) The LNKGCICS member provides specific instructions. Be sure to link the globals table into a load library that will be made available to CICS in the DFHRPL library concatenation. Note that any user or Software AG link routine exits should be link-edited with this load module.
6. Modify the DEFADA8 member to provide the correct name of the link routine globals default table created in the previous step (Step 4). The default module name is CICSGBL. Tailor this member for any other CICS installation values as required.

Note:

The Adabas 8 CICS program names, other than the name of the link globals table, are predefined and cannot be changed (for example, ADACICS, ADACICT, ADACIRQ, and ADACIC0).

7. Run the IBM DFHCSDUP utility to update the CICS CSD file for the desired CICS using the modified DEFADA8 member as input.
8. Modify the CICS PLTPI table to add the entries that will enable and start the Adabas CICS task-related user exits (TRUE). Use member ADAPLTXX from the Adabas 8 ACIvrn.SRCE library as a sample for enabling and starting a legacy Adabas TRUE and the new Version 8 TRUE in the second phase of the PLT.
9. Assemble and link the modified PLTPI table into a library that will be available to the desired CICS region.
10. Modify the CICS startup JCL to include the DDLINK DD statement that provides the name of the link globals table prepared in Step 4. For more information on doing this, read *Preparing DDLINK Input for CICS*.
11. Start the CICS and note any messages relating to the installation of the Adabas TRUE modules that appear on the console.

Preparing DDLINK Input for CICS

Operation of the Adabas 8 CICS link routines may be tailored for each CICS address space by assembling and linking a link globals defaults table and making that table available to CICS at execution. The table may have any legal load module name that is acceptable to CICS and that does not conflict with existing load module names used in the CICS region.

The globals table must be defined to CICS as a program. Review the provided sample DEFADA8 member found in the Adabas 8 ACIvrn.SRCE library to see a definition of the sample called LNKGBLS. The DEFADA8 member should be modified as necessary and provided as input to the DFHCSDUP utility to define the Adabas 8 components in the CICS CSD. This process is also described as part of the installation procedure under Adabas 8. Consult the IBM CICS documentation for information on the DFHCSDUP utility.

Each link routine globals table for a CICS region may have a unique name. The Adabas 8 CICS link routines are provided this name through an external transient data queue. The queue-name is ADAI and its definition is also provided in the DEFADA8 member.

When the Adabas 8 task-related user exit (TRUE) is enabled via the ADACIC0 program, either at CICS startup or with the ADA0 transaction, the ADACIRQ module is invoked and reads the ADAI transient data queue. The data read is provided in a file, or a partitioned data set member, or as system input through the DDLINK DD statement. The CICS JCL must be modified to provide this DD statement or the default link globals table name "LNKGBLS" will be used. If no link globals table is located, the Adabas 8 TRUE will not be enabled and started.

The format of the input data to be read is:

Content	Description
ADALNK	This keyword must appear in columns 1 through 6.
space	A blank space must appear in column 7.
LGTNAME= or LGT=	The keyword LGTNAME or LGT, followed by an equals sign (=) must appear after the space in column 7, starting in column 8.
<i>module-name</i>	The module name of the prepared link globals table must appear after the equal sign that follows the LGTNAME or LGT keyword.

For example, the following might be added to the CICS JCL:

```
//DDLINK DD *
ADALNK LGTNAME=LNKGBLS
/*
```

In this example, the link default globals table named "LNKGBLS" must be prepared, assembled, link-edited, and defined to this CICS.

Installing Adabas with Com-plete under Adabas 8

Certain Adabas parameters are required by Com-plete, Software AG's TP monitor, when installing Adabas. For more information, see the *Com-plete System Programmer's* manual.

Software AG's TP monitor, Com-plete requires an Adabas link routine if it is to communicate with Adabas databases, use Software AG's Entire Net-Work product, or use products like Entire System Server running under Com-plete. At this time, Com-plete does not support a mixed Adabas 7 and Adabas 8 link routine environment; thus Com-plete must be run with either an Adabas 7 link routine or an Adabas 8 link routine.

The Adabas Version 8 link routine is delivered in member ADALCO of the Adabas 8 z/OS load library. This member must be linked with a link globals module you prepare and with any link routine exits you require to create the final ADALCO load module that is loaded by Com-plete when Com-plete is initialized. The final ADALCO load module and any exits linked with it must be reentrant.

The following table lists the modules supplied in your Adabas installation to support the installation of Adabas with Com-plete under Adabas 8.

Module	Description
ADALCO8	Base module
ADALCO	Executable default module

➤ **To prepare the Adabas 8 link routine**

1. Copy sample member LCOGBL provided in the Adabas 8 ADA vrs .SRCE library to any appropriate user source library where it can be modified (where vrs is the number of the latest Adabas version delivered on the installation medium). LCOGBL is a module containing LGBLSET parameters that are used to create default settings for command-level link components. A complete description of LGBLSET parameters can be found in *Modifying Source Member Defaults (LGBLSET Macro) in Version 8*.
2. Modify the LCOGBL member in the user source library.

At a minimum supply values for the following LGBLSET parameters in LCOGBL:

Parameter	Specify...
LOGID	<p>The default database or target ID. This should be a numeric value between "1" and "65535". The default value is "1".</p> <p>Note: Specifying default values for LOGID and SVCNO under Com-plete is for documentation purposes only. The ADASVC Com-plete runtime control statements will provide the supported database ID/Adabas SVC combinations to be used in running Com-plete. For more information, read the Com-plete documentation.</p>
SVCNO	<p>The default Adabas SVC number. For z/OS, this number should be between "200" and "255".</p> <p>Note: Specifying default values for LOGID and SVCNO under Com-plete is for documentation purposes only. The ADASVC Com-plete runtime control statements will provide the supported database ID/Adabas SVC combinations to be used in running Com-plete. For more information, read the Com-plete documentation.</p>
OPSYS	<p>The three-character abbreviation for the operating system under which Com-plete executes. Valid values include "ZOS" and "VSE".</p> <p>Note: The OPSYS parameter must be set to ZOS.</p>
TPMON	COM. This keyword specifies the three-character TP monitor abbreviation. For Com-plete, this value should be "COM".
RENT	YES. This keyword indicates whether or not the module is serially reentrant. For Com-plete, this value should be "YES".
GEN	CSECT. This keyword indicates whether a CSECT or DSECT is generated. CSECT must be specified so an object module is generated that can be linked as the link routine globals load module.
UES	Whether Adabas Universal Encoding Support (UES) should be enabled. The default is YES.
exit parameters	Whether any other exits are to be active, and in the case of user exits you provide, specify the user exit module names. Specify this information in other parameters of LGBLCOM, as described in <i>Modifying Source Member Defaults (LGBLSET Macro) in Version 8</i> .

3. Modify and run sample job ASMGBLS as described at the top of the job. ASMGBLS can be found in the Adabas 8 ADAvrs.JOBS library. When fully modified, the SET statement in the job should reference the LCOGBL member you prepared in the previous step and the NAME link edit control statement should reference the name specified by the GBLNAME parameter in the LCOGBL

member.

Once modified, submit the ASMGBLS job to assemble and link-edit the link globals module.

A new link globals module (with the name specified by the GBLNAME parameter in LCOGBL) will be generated in the user load library identified in the ASMGBLS job.

4. Copy sample job LNKLCO8 to a user source library and modify it to link the new link globals module you created in the previous step and any required exits with the ADALCO8 base module. Instructions for modifying the sample job are described at the top of the job. Be sure to direct the output from this job to an appropriate user load library. LNKLCO8 can be found in the Adabas 8 ADA*vrs*.SRCE library.

The module resulting from this job is called ADALCO.

5. Place the ADALCO module in a load library available in the job step that will start Com-plete.

The Adabas 8 link routine is prepared.

General Considerations for Installing Adabas with Batch/TSO

When installing Adabas 8 on TSO systems, Adabas-TSO communication is provided by the batch link routines ADALNK8 (non-reentrant) and ADALNKR8 (reentrant).

The Adabas Version 8.1 ADALNK routines are UES-enabled as distributed.

However, it is important to note that user programs linked with ADAUSER also load ADARUN. ADARUN, in turn, loads other modules.

To start a user program linked with ADAUSER, the following modules must all be available from the defined load libraries for that specific TSO user at execution time:

```
ADAIOR  ADAMLF
ADAIOS  ADAPRF
ADALNK  ADARUN
```

This section covers the following topics:

- Non-reentrant ADALNK Batch Routine Operation
- ADALNKR: Reentrant Batch Link Routine

Non-reentrant ADALNK Batch Routine Operation

The ADALNK module in the Adabas 8 load library operates in an Adabas 7-compatible manner when the following conditions are met:

- The calling application must be linked with ADAUSER. If the calling application is not linked with ADAUSER, the ADALNK will not work.

- The ADARUN module from the most recent Adabas 8 load library must be used.
- The database ID and Adabas SVC number must be provided as input through DD statements. Otherwise, the values in the link globals table will override these values.

If all three of these conditions are met, the default database ID and Adabas SVC number will be overridden by the values provided in the DD statement input and passed to the link routine by ADARUN.

Operating in this fashion requires the fewest changes on the part of your data base administrator (DBA) and application programmer. This is also the recommended mode of operation when executing Adabas utilities.

ADALNKR: Reentrant Batch Link Routine

Several Software AG products require the use of a reentrant batch link routine and the ADALNKR load module is provided in the Adabas load library to support them. The Adabas 8 ADALNKR source module is not provided.

You can change default values for these reentrant batch link routines. For more information, read one of the following sections:

- *Changing Default Values for the Adabas 8 ADALNK or ADALNKR Modules*
- *Zapping the Default Values for the Adabas 8 ADALNK or ADALNKR Modules*

Software AG recommends that batch application programs be linked with the ADAUSER module, not ADALNK or ADALNKR. The ADAUSER load module is not reentrant, but the reentrant link routine module may be linked with it as long as the application program conforms to the calling requirements described in *Adabas 8 Batch/TSO Reentrant Link Routine (ADALNKR) Calling Requirements* and the PROG=RENTUSER ADARUN parameter is provided in DDCARD input instead of the keyword parameter PROG=USER.

When using the latest Adabas 8 ADALNKR module to obtain reentrant operation under batch or TSO, you must prepare the ADALNKR module in advance. It must be linked with a customized link globals table that provides defaults for the database ID, Adabas SVC number, and other requirements. Any reentrant exits should also be linked with it as required.

Installing Adabas with Batch/TSO under Adabas 8

When installing Adabas 8 on TSO systems, the standard Adabas 8 batch link routine (ADALNK) provides Adabas/TSO communication (SMA job number I056).

This section covers the following topics:

- Supplied Modules
- Changing Default Values for the Adabas 8 ADALNK or ADALNKR Modules
- Zapping the Default Values for the Adabas 8 ADALNK or ADALNKR Modules

Supplied Modules

The following table lists the modules supplied in your Adabas installation to support the installation of Adabas with batch/TSO under Adabas 8.

Module	Description
ADALNK8	Base module
ADALNKR8	Base reentrant module
ADALNK	Executable default module
ADALNKR	Executable default reentrant module

Changing Default Values for the Adabas 8 ADALNK or ADALNKR Modules

You can change default values for various link routine parameters used by the Adabas 8 ADALNK and ADALNKR modules.

➤ To change default values, complete the following steps

1. Copy the sample member LNKGBLS (for non-reentrant links) or LNKRGBL (for reentrant links) members provided in the Adabas 8 ADA vrs (where vrs is the number of the latest Adabas version delivered on the installation medium).SRCE library to any appropriate user source library where they can be modified. These modules contain LGBLSET parameters that are used to create default settings for link components. A complete description of LGBLSET parameters can be found in *Modifying Source Member Defaults (LGBLSET Macro) in Version 8*.
2. Modify the LNKGBLS or LNKRGBL member in the user source library. Provide values for the LOGID, SVC, and other keywords to suit your installation requirements.

Note:

The OPSYS parameter must be set to ZOS.

3. Modify and run sample job ASMGBLS as described at the top of the job. ASMGBLS can be found in the Adabas 8 ADA vrs .JOBS library. When fully modified, the SET statement in the job should reference the LNKGBLS or LNKRGBL member you prepared in the previous step and the NAME link edit control statement should reference the name specified by the GBLNAME parameter in the LNKGBLS or LNKRGBL member.

Once modified, submit the ASMGBLS job to assemble and link-edit the link globals module.

A new link globals module (with the name specified by the GBLNAME parameter in the LNKGBLS or LNKRGBL member) will be generated in the user load library identified in the ASMGBLS job.

4. Copy sample job LNKLNK8 or LNKLNKR8 (reentrant) to a user source library and modify it to link the new link globals module you created in the previous step and any required exits with the appropriate ADALNK8 or ADALNKR8 (reentrant) base module. Instructions for modifying the sample job are described at the top of the job. Be sure to direct the output from the job to an appropriate user load library. LNKLNK8 and LNKLNKR8 can be found in the Adabas 8 ADA vrs .SRCE library.

The module resulting from this job is called ADALNK or ADALNKR (as appropriate).

5. Tailor the ADARUN DDCARD input for the job steps that will use the Adabas 8 batch/TSO link routines. The DDCARD input should include the following updates:
 - Specify the ADARUN PROG=USER parameter for a non-reentrant link routine, or specify ADARUN PROG=RENTUSER to use a reentrant link routine in the job step.
6. Make sure the appropriate load libraries are made available to the job step. These may be STEPLIB, TASKLIB, JOBLIB, or, for reentrant modules, the LPA or LINKLIB.

Zapping the Default Values for the Adabas 8 ADALNK or ADALNKR Modules

You can change default values for various link routine parameters used by the Adabas 8 ADALNK and ADALNKR modules.

Changes to some default values for the Adabas 8 batch/TSO link routines, ADALNK and ADALNKR, may occur with a zap to either the ADALNK or ADALNKR module. This includes the default values for the database ID and the Adabas SVC number. All other default values should be set using the link globals table, as described in *Changing Default Values for the Adabas 8 ADALNK or ADALNKR Modules*.

Use the following IMASPZAP control statements to change default values in ADALNK or ADALNKR (as appropriate):

```
NAME ADALNK ADALNK8
VER 0080 0001           Default DBID
REP 0080 #####         Site-specific DBID
VER 0084 0AF9           Default Adabas SVC number
REP 0084 0A##           Site-specific Adabas SVC number
*
NAME ADALNKR ADALNKR8
VER 0080 0001           Default DBID
REP 0080 #####         Site-specific DBID
VER 0084 0AF9           Default Adabas SVC number
REP 0084 0A##           Site-specific Adabas SVC number
```

Modifying Source Member Defaults (LGBLSET Macro) in Version 8

The Adabas 8 LGBLSET macro is used to set default installation values for the Adabas link routines. It is used to prepare an object module which may either be link-edited with the Adabas 8 link routines or provided to the link routines in the job step where they are run. Your Adabas libraries include sample members provided to support the various teleprocessing (TP) monitors in each environment. Each of these sample members may be copied to an appropriate library and modified to provide the necessary customization required for the link routine that is intended to run in a given environment.

The LGBLSET parameter options with their default values (underlined> are described in the rest of this section:

- ADL: Adabas Bridge for DL/I Support

- AVB: Adabas Bridge for VSAM Support
- CITSNM: Adabas CICS TS Queue Name
- COR: SYSCOR Exit Support
- DBSVCTN: DBID/SVC Routing Table
- DYNDBSVC: DBID/SVC Routing Table
- ENTPT: Name of the Adabas CICS Command-Level Link Routine
- GBLNAME: Name of Link Globals Module
- GEN: Generate CSECT or DSECT
- IDTNAME: BS2000/OSD IDT Common Memory Name
- IDTUGRP: BS2000/OSD Memory Pool User Bound
- LOGID: Default Logical Database ID
- LUEXIT1A: Length of LUEXIT1
- LUEXIT2A: Length of LUEXIT2
- LUINFO: Length of User Data passed to Adabas LUEXIT1 and LUEXIT2
- LUSAVE: Size of User Save Area for Adabas LUEXIT1 and LUEXIT2
- LX1NAME: User Exit 1 Module Name
- LX2NAME: User Exit 2 Module Name
- MRO: Multiple Region Option
- NETOPT: Method Used to Create User ID
- NTGPID: Natural Group ID
- NUBS: Number of User Blocks Created By CICS Link Routine
- OPSYS: Operating System
- PARMTYP: Area for Adabas Parameter List
- PRE: DSECT Data Prefix
- PURGE: Purge Transaction
- RENT: Reentrant Module Flag
- RETRYX: Retry Command Exit Flag

- REVIEW: Adabas Review Support
- RMI: Resource Manager Interface
- RTXNAME: Command Retry Exit Name
- SAF: Adabas Security Interface Flag
- SAP: SAP Application Support
- SAPSTR: SAP ID String
- SVCNO: Adabas SVC number
- TPMON: Operating Environment
- TRUENM: CICS TRUE Name
- UBPLOC: User Block Pool Allocation
- UES: Universal Encoding Support
- USERX1: User Exit 1 Flag
- USERX2: User Exit 2 Flag
- XWAIT: XWAIT Setting for CICS

ADL: Adabas Bridge for DL/I Support

Parameter	Description	Syntax
ADL	<p>Indicates whether or not the Consistency Interface of Software AG's Adabas Bridge for DL/I is to be supported by this command-level link routine.</p> <ul style="list-style-type: none"> ● ADL=YES: Adabas Bridge for DL/I Consistency Interface is to be supported. ● ADL=NO: Adabas Bridge for DL/I Consistency Interface is <i>not</i> to be supported. 	ADL={ <u>NO</u> YES }

AVB: Adabas Bridge for VSAM Support

Parameter	Description	Syntax
AVB	<p>Indicates whether or not Software AG's Adabas Bridge for VSAM is to be supported by this command-level link routine.</p> <ul style="list-style-type: none"> ● AVB=YES: Adabas Bridge for VSAM is to be supported. ● AVB=NO: Adabas Bridge for VSAM is <i>not</i> to be supported. 	AVB={ <u>NO</u> YES }

CITSNM: Adabas CICS TS Queue Name

Parameter	Description	Syntax
CITSNM	Specifies the 16-byte string that represents the CICS TS queue name for Adabas. The default is "ADACICS".	CITSNM={ <u>ADACICS</u> <i>qname</i> }

COR: SYSCOR Exit Support

Parameter	Description	Syntax
COR	<p>Indicates whether or not Adabas System Coordinator (SYSCOR), Adabas Transaction Manager, and Adabas Fastpath exits are installed and active.</p> <ul style="list-style-type: none"> ● COR=YES: The exits are installed and active. ● COR=NO: The exits are <i>not</i> installed and active. 	COR={ <u>NO</u> YES }

DBSVCTN: DBID/SVC Routing Table

Parameter	Description	Syntax
DBSVCTN	<p>Provides the name of the DBID/SVC routing table that should be used by the link routine during its execution, if any.</p> <p>The routing table name must conform to names for z/OS standard load modules. It is used by a z/OS LOAD macro/SVC during batch, TSO, or IMS operation or by an EXEC CICS LOAD PROGRAM command during CICS operation.</p> <p>If the load module listed is not found, or if it is found to contain invalid header information, user abend U657 is issued in batch, TSO, or IMS environments.</p> <p>If the load module is not defined to CICS or not found in the CICS DFHRPL concatenation, the Adabas CICS link routine environment is not initialized.</p> <p>Note: If the DYNDBSVC parameter is set to NO, this parameter setting is ignored.</p>	DBSVCTN= { <i>name</i> <u>ADASVCTB</u> }

DYNDBSVC: DBID/SVC Routing Table

Parameter	Description	Syntax
DYNDBSVC	Indicates whether Adabas SVC routing by database ID should be enabled for the link routine. DYNDBSVC=YES enables Adabas SVC routing by database ID; DYNDBSVC disables it. The default is NO.	DYNDBSVC= { YES <u>NO</u> }

ENTPT: Name of the Adabas CICS Command-Level Link Routine

Parameter	Description	Syntax
ENTPT	<p>The name given to the Adabas CICS command-level link routine. This name is used in EXEC CICS LINK commands to invoke Adabas services from CICS application programs.</p> <p>See also notes 1 and 2 in the installation procedure.</p>	ENTPT= { <u>ADACICS</u> <i>name</i> }

GBLNAME: Name of Link Globals Module

Parameter	Description	Syntax
GBLNAME	The name of the link globals module.	GBLNAME= { <u>LNKGBLS</u> <i>name</i> }

GEN: Generate CSECT or DSECT

Parameter	Description	Syntax
GEN	Indicates whether a CSECT or DSECT is generated.	GEN= { <u>CSECT</u> DSECT }

IDTNAME: BS2000/OSD IDT Common Memory Name

Parameter	Description	Syntax
IDTNAME	The common memory pool name of the BS2000/OSD IDT.	IDTNAME= <i>name</i>

IDTUGRP: BS2000/OSD Memory Pool User Bound

Parameter	Description	Syntax
IDTUGRP	Indicates whether the common memory pool is user bound (BS2000)	IDTUGRP= { <u>NO</u> YES }

LOGID: Default Logical Database ID

Parameter	Description	Syntax
LOGID	The value of the default target database ID. Valid ID numbers are 1-65535. The default is "1".	LOGID= { <i>nnn</i> <u>1</u> }

LUEXIT1A: Length of LUEXIT1

Parameter	Description	Syntax
LUEXIT1A	The length of the work area for link user exit 1. Valid values are numbers from zero (0) through 32,767. The default is "0".	LUEXIT1A= { <i>nnn</i> <u>0</u> }

LUEXIT2A: Length of LUEXIT2

Parameter	Description	Syntax
LUEXIT2A	The length of the work area for link user exit 2. Valid values are numbers from zero (0) through 32,767. The default is "0".	LUEXIT2A={ <i>nnn</i> <i>0</i> }

LUINFO: Length of User Data passed to Adabas LUEXIT1 and LUEXIT2

Parameter	Description	Syntax
LUINFO	The length of the user data to be passed to target user exit 4. Valid values are numbers from zero (0) through 32,767. If LUINFO is not specified, the default is zero (no user save area is passed).	LUINFO={ <i>0</i> <i>length</i> }

LUSAVE: Size of User Save Area for Adabas LUEXIT1 and LUEXIT2

Parameter	Description	Syntax
LUSAVE	The size of the user save area to be used by Adabas user exits LUEXIT1 and LUEXIT2. Valid values range from zero (0) through 256. The default is "72". If LUSAVE is not specified, the default is zero (no user data is passed).	LUSAVE={ <i>72</i> <i>size</i> }

LX1NAME: User Exit 1 Module Name

Parameter	Description	Syntax
LX1NAME	The name of the link user exit 1 module	LX1NAME={ <u>LUEXIT1</u> <i>name</i> }

LX2NAME: User Exit 2 Module Name

Parameter	Description	Syntax
LX2NAME	The name of the link user exit 2 module	LX2NAME={ <u>LUEXIT2</u> <i>name</i> }

MRO: Multiple Region Option

Parameter	Description	Syntax
MRO	<p>Indicates whether or not the CICS multiple region option (MRO) support is required.</p> <p>If you run the CICS command-level link with the CICS MRO, set this to MRO=YES; otherwise, use the default value MRO=NO.</p> <p>If MRO=YES, NETOPT must be set to NETOPT=NO (the default) to prevent non-unique LU names from multiple application regions.</p> <p>If NETOPT=YES and MRO=YES are specified, an assembler MNOTE and a return code of 16 are produced from the assembly step.</p>	MRO={ <u>NO</u> YES }

NETOPT: Method Used to Create User ID

Parameter	Description	Syntax
NETOPT	<p>If NETOPT=YES is specified, an 8-byte user ID will be constructed from the VTAM LU name.</p> <p>If NETOPT=NO is specified, the user ID is created from the constant CICS plus the four-byte CICS terminal ID (TCTTETI) for terminal tasks. For non-terminal tasks, the user ID comprises the constant CIC plus the CICS task number.</p> <p>If you run with the CICS multiple region option (MRO), you must use the default value for this option. If NETOPT=YES and MRO=YES are specified, an assembler MNOTE and a return code of 16 are produced from the assembly step.</p>	NETOPT={ <u>NO</u> YES }

NTGPID: Natural Group ID

Parameter	Description	Syntax
NTGPID	<p>Specifies a four-byte Natural group ID as required for unique Adabas user ID generation in the CICSplex environment with Natural Version 2.2.8 and above. The value is associated with all users who call the Adabas command-level link routine assembled with the specified value.</p> <p>There is no default value. If no value is specified, the Adabas internal user ID is built in the conventional manner.</p> <p>Any four-byte alphanumeric value may be specified, but it must be unique for each Adabas command-level link routine running in a CICSplex, or z/OS image. If more than one NTGPID is required (for example, both test and production Natural 2.2.8), more than one Adabas command-level link routine with associated TRUE must be generated.</p> <p>If you run with the CICS multiple region option (MRO), you may use NTGPID to provide a 4-byte literal for the Adabas communication ID to be used by the Adabas SVC when multiple application regions call Adabas.</p>	NTGPID= <i>4-byte-value</i>

NUBS: Number of User Blocks Created By CICS Link Routine

Parameter	Description	Syntax
NUBS	<p>The number of user blocks (UBs) to be created in the user block pool by the CICS link routine. The number of blocks must be large enough to handle the maximum possible number of concurrent Adabas requests.</p> <p>Note: The Adabas 6.2 and above command-level link routine obtains storage for the user blocks (the UB pool) above the 16-megabyte line.</p>	NUBS={ <u>100</u> <i>blocks</i> }

OPSYS: Operating System

Parameter	Description	Syntax
OPSYS	The operating system in use.	OPSYS={ <u>ZOS</u> VSE CMS BS2}

PARMTYP: Area for Adabas Parameter List

Parameter	Description	Syntax
PARMTYP	<p>The CICS area which is to contain the Adabas parameter list. "TWA" picks up the parameter list in the first six fullwords of the transaction work area (TWA).</p> <p>When PARMTYP=COM, the Adabas parameters are supplied in the CICS COMMAREA provided by the calling program with the EXEC CICS LINK command. The COMMAREA list for an ACB call must be at least 32 bytes long and begin with the label "ADABAS52". The COMMAREA list for an ACBX call must be at least 24 bytes long and begin with the label "ADABAS8X". In addition, the last ABD in the COMMAREA list for an ACBX call must be indicated by setting the VL-bit -- in other words, the high bit in the address must be on (X'80').</p> <p>PARMTYP=ALL (the default) uses both the COMMAREA and TWA to pass the Adabas parameters; in this case, the COMMAREA is checked first.</p> <p>We do not recommend that you attempt to map the CICS TWA to the Adabas 8 ACBX direct call. This is because the TWA is of finite size per transaction and because the TWA is not available at CICS startup. We therefore recommend that CICS programs using the Adabas 8 CICS link routines use the COMMAREA only for passing data.</p>	PARMTYP={ <u>ALL</u> COM TWA}

PRE: DSECT Data Prefix

Parameter	Description	Syntax
PRE	The two-byte string to be used as the DSECT data prefix. The default is "LG".	PRE={ <u>LG</u> <i>prefix</i> }

PURGE: Purge Transaction

Parameter	Description	Syntax
PURGE	<p>The PURGE parameter is used when assembling with CICS 3.2 or above. If PURGE=YES is specified, the CICS WAIT EXTERNAL will contain PURGEABLE as one of its parameters, allowing the transaction to be purged by CICS if the DTIMOUT value is exceeded and PURGE is specified.</p> <p>If PURGE=NO (the default) is specified, the NONPURGEABLE option is generated.</p>	PURGE={ <u>NO</u> YES }

RENT: Reentrant Module Flag

Parameter	Description	Syntax
RENT	Indicates whether the globals module is reentrant.	RENT={ <u>NO</u> YES }

RETRYX: Retry Command Exit Flag

Parameter	Description	Syntax
RETRYX	Indicates whether the retry command exit is active.	RETRYX={ <u>NO</u> YES }

REVIEW: Adabas Review Support

Parameter	Description	Syntax
REVIEW	Indicates whether or not Software AG's Review performance monitor is installed and active.	REVIEW={ <u>NO</u> YES }

RMI: Resource Manager Interface

Parameter	Description	Syntax
RMI	<p>The RMI parameter is used to indicate whether or not the CICS Resource Manager Interface is in use.</p> <p>If RMI=YES is specified, the Adabas task-related user exit (TRUE) will be executed as a resource manager (RM) using the CICS Resource Manager Interface (RMI).</p> <p>RMI=YES is valid only when the Adabas Transaction Manager is installed, enabled, and available to users executing in the CICS environment. Consult the Adabas Transaction Manager documentation for additional instructions related to the installation of the Adabas TRUE.</p>	RMI = { <u>NO</u> YES }

RTXNAME: Command Retry Exit Name

Parameter	Description	Syntax
RTXNAME	The name of the command retry exit module.	RTXNAME = { <u>LUEXRTR</u> <i>name</i> }

SAF: Adabas Security Interface Flag

Parameter	Description	Syntax
SAF	Indicates whether Software AG's Adabas SAF Security support is required.	SAF = { <u>NO</u> YES }

SAP: SAP Application Support

Parameter	Description	Syntax
SAP	<p>Indicates whether or not SAP user ID generation is supported.</p> <p>If SAP=YES is specified, the program will detect a SAP initialization call and set the user ID for SAP applications from the constant provided on the initialization call, plus the field ACBADD2.</p> <p>For more information, refer to the supplementary information provided to customers using the SAP application system.</p>	SAP = { <u>NO</u> YES }


SAPSTR: SAP ID String

Parameter	Description	Syntax
SAPSTR	The four-byte SAP ID string to use.	<code>SAPSTR={ '<u>SAP*</u>' <i>string</i> }</code>

SVCNO: Adabas SVC number

Parameter	Description	Syntax
SVCNO	<p>The value of the Adabas SVC number.</p> <p>On z/OS systems, valid values range from 200-255 and the default is "249".</p> <p>On z/VSE systems, valid values range from 32-128 and the default is "45".</p>	<code>SVCNO=<i>nnn</i></code>

TPMON: Operating Environment

Parameter	Description	Syntax
TPMON	<p>The TP monitor operating environment. Valid values should be specified as follows:</p> <ul style="list-style-type: none"> • Specify "BAT" to use batch. • Specify "CICS" to use CICS. • Specify "COM" to use Com-plete. • Specify "IMS" to use IMS. • Specify "TSO" to use TSO. • Specify "UTM" to use UTM. <p> Warning: Be sure to specify a TP monitor operating environment that is supported on the operating system you selected in the OPSYS parameter.</p>	<code>TPMON={ <u>BAT</u> CICS COM IMS }</code>

TRUENM: CICS TRUE Name

Parameter	Description	Syntax
TRUENM	Specifies the module name of the Adabas CICS task-related user exit (TRUE). The default is ADACICT.	TRUENM= { <u>ADACICT</u> <i>name</i> }

UBPLOC: User Block Pool Allocation

Parameter	Description	Syntax
UBPLOC	<p>Specifies whether the user block (UB) pool is to be obtained above (the default) or below the 16-megabyte line in CICS.</p> <p>The ECB used by the EXEC CICS WAIT WAITCICS or the EXEC CICS WAIT EXTERNAL is included in the UB pool.</p> <p>The UBPLOC=BELOW setting supports versions of CICS that do not allow ECBs above the 16-megabyte line; that is, CICS/ESA 3.2 or below.</p> <p>Refer to the IBM manual <i>CICS Application Programming Reference</i> for more information.</p>	UBPLOC= { <u>ABOVE</u> BELOW }

UES: Universal Encoding Support

Parameter	Description	Syntax
UES	Indicates whether or not Universal Encoding Support (UES) is required.	UES= { NO <u>YES</u> }

USERX1: User Exit 1 Flag

Parameter	Description	Syntax
USERX1	Indicates whether or not user exit 1 is active.	USERX1= { <u>NO</u> YES }

USERX2: User Exit 2 Flag

Parameter	Description	Syntax
USERX2	Indicates whether or not user exit 2 is active.	USERX2= { <u>NO</u> YES }

XWAIT: XWAIT Setting for CICS

Parameter	Description	Syntax
XWAIT	<p>Indicates whether a standard EXEC CICS WAITCICS (XWAIT=NO) or a WAIT EVENTS EXTERNAL (XWAIT=YES) will be executed by the Adabas 8 task-related user exit (TRUE). XWAIT=YES is the default.</p> <p>The CICS WAIT EVENTS EXTERNAL (XWAIT=YES) is the recommended interface for CICS/TS 1.1 and above.</p> <p>The CICS WAITCICS statement (XWAIT=NO) is provided for use with CICS/MVS 2.1.2 and for CICS/VSE 2.1 through 2.3. It may also be used for CICS/TS 1.1 and above, but may result in poor CICS transaction performance or unpredictable transaction results in busy CICS environments.</p> <p>Note: If XWAIT=NO is specified for use under CICS/ESA 3.3, IBM APAR PN39579 must be applied to the CICS/ESA 3.3 system. For CICS/TS 1.1 and above, this APAR is not required.</p>	XWAIT={NO YES}

Notes:

1. If XWAIT=NO is specified, the ADACICT (Adabas 8 TRUE) module issues an EXEC CICS WAITCICS command instead of the EXEC CICS WAIT EVENT command. XWAIT=YES conforms with recommended IBM usage of the WAIT and ECB lists in a high-transaction volume CICS system with CICS/TS Version 1.1 and above.
2. All EXEC CICS commands are processed by the CICS preprocessor; the LGBLSET parameters cause the subsequent assembly step to skip some of the statements.

XWAIT Posting Mechanisms

CICS WAITCICS (XWAIT=NO) can support a soft post of the specified ECB. This has the disadvantage of becoming a low priority dispatchable unit of work in a CICS environment, since the hand-postable work is not processed by CICS on every work cycle.

EXEC CICS WAIT EXTERNAL (XWAIT=YES), on the other hand, allows CICS to make use of its special post exit code, and will always be checked and processed (if posted) on every CICS work cycle.

For more details on the differences between the various CICS WAIT commands and their relationship to hard and soft posting mechanisms, consult the IBM *CICS Application Programming Reference Guide* and the texts accompanying IBM APAR PN39579 or "Item RTA000043874" on the IBM InfoLink service.

XWAIT and the Adabas SVC / Router

The Adabas SVC is fully compatible with the XWAIT=YES setting. The SVC performs the necessary hard post for Adabas callers under CICS using the Adabas command-level link routine. The same SVC performs a soft post for batch callers where the hard post is not required.