

Adabas System Coordinator

Adabas System Coordinator Versioning Tool

Version 8.2.2

October 2017

This document applies to Adabas System Coordinator Version 8.2.2 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2017 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: COR-VERSIONING-822-20171008

Table of Contents

1 Adabas System Coordinator Versioning Tool	1
2 Supported Products	3
3 Components	5
Client Versioning	6
Database Versioning	6
Daemon Versioning	6
4 Considerations	7
Client Versioning	8
Database Versioning	8
5 Installation	9
Objects	10
Overview	10
Before You Start	10
Client Versioning (Batch)	11
Client Versioning (CICS)	14
Database Versioning	18
Daemon Versioning	18
Final Steps	19
Installation Controls	19
Control Descriptions	22
Installation Verification	26

1 Adabas System Coordinator Versioning Tool

This document describes the Adabas System Coordinator Versioning Tool which enables multiple versions of Add-on products to co-exist in the Adabas databases and the clients that use them. This capability provides increased levels of flexibility when upgrading the Add-on products.



Note: The Versioning Tool is currently only available for z/OS.

[Supported Products](#)

[Components](#)

[Considerations](#)

[Installation](#)

2 Supported Products

The following Add-on products are currently supported for co-existence:

- Adabas System Coordinator Version 8.2.2 and above.
- Adabas Fastpath Version 8.2.2 and above.
- Adabas Vista Version 8.2.2 and above.
- Adabas SAF Security Version 8.2.2 and above.
- Adabas Transaction Manager Version 8.2.2 and above.
- SAF Security daemon supplied on the Adabas limited library (WAL) version 8.2.5 and above.

3 Components

▪ Client Versioning	6
▪ Database Versioning	6
▪ Daemon Versioning	6

The Versioning Tool has the following components:

Client Versioning

Client Versioning is used when, for example, different transactions in the same CICS job require the use of different versions of System Coordinator and its co-products.

The Client Versioning component enables Adabas commands to be routed to different Adabas link modules based on Job Name (for Batch) or Transaction ID (for CICS) with each Adabas link module using a different version of the supported Add-on products.

Database Versioning

Database Versioning allows multiple clients, each running a single version of the Add-on products, to access the same database.

The Database Versioning component ensures client and daemon messages issued from multiple versions of the Add-on products are serviced correctly by the appropriate version of the Add-on product's database component.

Daemon Versioning

Multiple version support in the daemon is achieved by running a separate daemon for each required version. The daemon does not support multiple versions in a single daemon instance.

The implementation of Client and/or Database Versioning necessitates all version-related Add-on product modules to be suffixed with a particular identifier. Daemon Versioning allows you to use these same re-named product modules in the Adabas System Coordinator and Adabas SAF Security daemons.

This enables the use of a common versioning library, with common module suffixes, for Adabas databases and daemon tasks.

4

Considerations

■ Client Versioning	8
■ Database Versioning	8

Client Versioning

It is possible to implement Client Versioning without implementing database versioning, but care must be taken to ensure that commands cannot be routed to a database that is not running the appropriate version.

In a re-entrant batch environment, Client Versioning is established by the first task to issue an Adabas call. If that task terminates prior to other tasks which also issue Adabas calls, results are unpredictable.

Database Versioning

It is possible to implement Database Versioning without implementing Client Versioning. Commands issued to the database will be automatically routed to the appropriate version of the Add-on product's database component.



Note: Implementing Database Versioning will have an impact on the memory allocation for the database. Memory region sizes may need to be reviewed and increased as necessary.

5

Installation

■ Objects	10
■ Overview	10
■ Before You Start	10
■ Client Versioning (Batch)	11
■ Client Versioning (CICS)	14
■ Database Versioning	18
■ Daemon Versioning	18
■ Final Steps	19
■ Installation Controls	19
■ Control Descriptions	22
■ Installation Verification	26

This section describes the steps needed to install the *Adabas System Coordinator Versioning Tool*.

Objects

The Versioning Tool objects (load modules, macros and sample jobs) are all supplied with Adabas System Coordinator. All objects have the name prefix “VER”.

Overview

The steps needed for a successful installation of the Versioning Tool are as follows:

Section	Description	Required	Job Name
<i>Before You Start</i>	Create Versioning Libraries.	Yes	
<i>Client Versioning (Batch)</i>	Create client versioning tables(s) and link module(s) for Batch clients.	If Batch client versioning is to be used.	VERI065 VERI085 VERI088
<i>Client Versioning (CICS)</i>	Create client versioning tables(s) and link module(s) for CICS clients.	If CICS client versioning is to be used.	VERI065 VERI085
<i>Database Versioning</i>	Assemble the database versioning table.	If database versioning is to be used.	VERI055
<i>Daemon Versioning</i>	Assemble the daemon versioning table.	If daemon versioning is to be used.	VERI060
<i>Final Steps</i>	Copy and rename required product modules into the Versioning Library. Add the Versioning Libraries to job control.	Yes	VERI080*

Before You Start

Before you start, versioning load libraries need to be created.

- [Load library for Versioning](#)

- [Additional load library for Client Versioning](#)

Load library for Versioning

Software AG recommends that versioning is implemented by creating a new library to hold copies of product modules that have been renamed in support of versioning. This will protect the original modules. It will also enable new maintenance to be applied to the original modules, which can then be refreshed in the versioning libraries by re-running the appropriate installation jobs.

This library is referred to as the *ALLVERS Versioning Library* and, in the sample job control supplied, is named SAG.CORVER.ALLVERS.LOAD.

The library is added to the Joblib or Steplib concatenation of all Adabas databases, client application jobs (for example, batch jobs, TSO CLISTs, CICS jobs), batch Adabas server start-up jobs and Co-ordinator daemon tasks that require versioning support.

Additional load library for Client Versioning

If you are implementing Client Versioning, you need to create a second library that will be used for the special link modules (VERLNK01, VERCIC01, ADALNK nn , ADABAS nn), and the client versioning tables. This library is referred to as the Client Versioning Library and, in the sample job control supplied, is named SAG.CORVER.CLIENT.LOAD.



Note: It is imperative that these link modules are only used in the client environment.

Therefore, they must not be placed in the ALLVERS Versioning Library that is used by the Adabas databases and System Coordinator or SAF Security daemons.

Client Versioning (Batch)

- [Create the Batch versioning table \(Job I065\)](#)
- [Copy the client versioning modules to the Client Versioning Library \(Job I085\)](#)
- (Optional) [Zap the Batch versioning link module defaults \(Job I088\)](#)

- Copy required link modules to the Client Versioning Library

Create the Batch versioning table (Job I065)

Use sample job VERI065 to assemble and link the Batch versioning table, VERC01 (standard) or VERC09 (re-entrant).



Note: The versioning table load modules must be linked REUS, and not RENT.

Using the VERCL macro, the client versioning table for Batch clients can be created. Here is an example:

```
* -----
* Default controls for Batch client versioning
* -----
      VERCL JOBTYP= BATCH,  Batch clients
          OVERIDES=YES,
          WTO=NO,
          LNKGBLXX=YES,
          BINDGBLS=IGNORED,
          DBID=DDCARD,
          SVC=DDCARD
* -----
* >>> Versioned Link Module Entry
* -----
      VERCL VRL=823,          Version 8.2.3
          DEFAULT=YES,      Default link module
          SUFFIX=82
* -----
* >>> Versioned Link Module Entry
* -----
      VERCL VRL=814,          Version 8.1.4
          SUFFIX=81,
          DBID=15,
          SVC=ASIS
* -----
* >>>>> Specific Jobname Entry
* -----
      VERCL JOBNAME=jobname, Uses Version 8.1.4
          DBID=15,
          SVC=ASIS
* -----
      VERCL TYPE=END          End of Controls
```

The (mandatory) first element introduces default controls for (a) general Batch client versioning and (b) Versioned Link Module Entries.

Following the mandatory first element are elements which identify the versioned link module(s) to be used – these are referred to as Versioned Link Module Entries. At least one entry must be

defined and one of those entries must be defined to be the default link module. In the example above, the version 8.2.3 link module identified by a suffix of 82 has been defined to be the default link module.

Versioned Link Module Entries which are not defined as the default link module are each followed by one or more Specific Jobname Entries which comprise of a list of job names that will use the versioned link module that precedes them. In the example above *jobname* will use the version 8.1.4 link module identified by a suffix of 81.

If a job name is not matched (at runtime) with an entry in the versioning table then it will use the default link module. In the example above, this would be the version 8.2.3 link module identified by a suffix of 82.

Versioned Link Module Entries can optionally choose to override the default controls defined by the mandatory first element but only if `OVERIDES=YES` has been specified (as in the example above).

Refer to [Client Versioning Controls \(Batch\)](#) for an explanation of the VERCL controls.

Copy the client versioning modules to the Client Versioning Library (Job I085)

Use sample job VERI085 to copy the supplied load module VERLNK01 to your Client Versioning Library (see "[Additional Load library for Client Versioning](#)") as ADALNK and the supplied load module VERLNR01 as ADALNKR (making sure the real ADALNK and ADALNKR are not lost).



Important: This Client Versioning Library must not be made available to any target in the Software AG network which acts as a DBID – such as Adabas, Broker, Com-Plete, Entire System Server, Net-Work, System Coordinator daemon etc.

(Optional) Zap the Batch versioning link module defaults (Job I088)

Certain fixed offsets in the Adabas link module contain information which is required by some callers of ADALNK, for example, the length of the USER and REVIEW extensions.

If you use other than default values, you must zap the correct values into the ADALNK and ADALNKR copies of VERLNK01 and VERLNR01. Sample job VERI088 may be used to do this.

Copy required link modules to the Client Versioning Library

Copy your existing ADALNK and LNKGBLS modules to the Client Versioning Library, renaming them to have the appropriate suffix:

ADALNK_{xx} / LNKGBL_{xx} for standard batch.

ADALNR_{xx} / LNRGBL_{xx} for re-entrant batch.

Client Versioning (CICS)

- Prepare CICS clients
- Create the CICS versioning table (Job I065)
- Link-edit VERCIC01 into the Client Versioning Library (Job I085)
- Copy required link modules to the Client Versioning Library

Prepare CICS clients

There are several prerequisites which must be fulfilled before you can use the CICS client versioning facility:

- Ensure that `PARMTYP=ALL` is set in `LGBLSET` (this is the default).
- PPT entries must be defined and installed for the following:
 - The Adabas client versioning link module (ADABAS, or other name you may choose). This **must** be specified as Resident.
 - The assembled CICS versioning table VERC03. This **must** be specified as Resident.
 - All renamed Adabas CICS link modules, Adabas TRUEs and Adabas PLT programs (LNKENAB and ADACIC0).
 - All renamed Add-on product modules (if they currently have PPT entries defined for them).

Create the CICS versioning table (Job I065)

Use sample job VERI065 to assemble and link the CICS versioning table, VERC03.

Using the VERCL macro, the client versioning table for CICS clients can be created. Here is an example:

```

* -----
* Default controls for CICS client versioning
* -----
      VERCL JOBTYP=ICIS,   CICS clients
      COFF=0,
      WTO=NO
* -----
* >>> Versioned Link Module Entry
* -----
      VERCL VRL=823,      Version 8.2.3
      DEFAULT=YES,      Default link module
      SUFFIX=82
* -----
* >>> Versioned Link Module Entry
* -----
      VERCL VRL=814,      Version 8.1.4
      SUFFIX=81
* -----
* >>>>> Specific Transaction-ID Entry
* -----
      VERCL TRAN=AA82      Uses Version 8.1.4
* -----
      VERCL TYPE=END      End of Controls

```

The (mandatory) first element introduces default controls for general CICS client versioning.

Following the mandatory first element are elements which identify the versioned link module(s) to be used – these are referred to as Versioned Link Module Entries. At least one entry must be defined and one of those entries must be defined to be the default link module. In the example above, the version 8.2.3 link module identified by a suffix of 82 has been defined to be the default link module.

Versioned Link Module Entries which are not defined as the default link module are each followed by one or more Specific Transaction-ID Entries which comprise of a list of transaction-IDs that will use the versioned link module that precedes them. In the example above transaction-ID AA82 will use the version 8.1.4 link module identified by a suffix of 81.

If a Transaction-ID is not matched (at runtime) with an entry in the versioning table then it will use the default link module. In the example above, this would be the version 8.2.3 link module identified by a suffix of 82.

Refer to [Client Versioning Controls \(CICS\)](#) for an explanation of the VERCL controls.

Link-edit VERCIC01 into the Client Versioning Library (Job I085)

Use sample job VERI085 to link the supplied load module VERCIC01 with your CICS stubs, naming the output load module appropriately (for example ADABAS).

Copy required link modules to the Client Versioning Library

For each suffix defined in the CICS versioning table, you will need modules named as follows:

Module	Description
ACIOPT xx	Adabas CICS Installation Options Table.
ACINAM xx	Adabas CICS Names Module.
CICGBL xx	Adabas CICS Globals Table.
ADAENA xx	Adabas CICS PLT TRUE enabler.
ADABAS xx	Adabas CICS Link module.
ADATRU xx	Adabas CICS TRUE.

where xx is the suffix and the first 6 characters of the module name must be as shown above.

Sample steps to create a set of versioning modules with suffix 82:

Step	Description and sample input
Assemble and link ACIOPT82	<p>Use the Adabas CICS sample job ASMCOPT with input:</p> <pre>MACIOPT ENTRY=GLOBAL,MNTRUE=8 MACIOPT ENTRY=GROUP,GTNAME=CICGBL82 MACIOPT ENTRY=FINAL END</pre>
Assemble and link ACINAM82	<p>Use the Adabas CICS sample job ASMCINS with input:</p> <pre>MACINS TRUENAME=ADATRU82,GTNAME=CICGBL82 END</pre>
Assemble and link CICGBL82	<p>Use the Adabas CICS sample jobs ASMGBLS and LNKGCICS and source CICSGBL with:</p> <pre>GBLNAME=CICGBL82, ENTPT=ADABAS, TRUENM=ADATRU82,</pre> <p>ENTPT= must specify the name of the linked VERCIC01 module. Refer to Link-edit VERCIC01 into the Client Versioning Library (Job I085) above.</p>

Step	Description and sample input
Create ADAENA82	<p>Link ADACIC0 with ACIOPT82 created in an earlier step:</p> <pre> MODE AMODE(31),RMODE(ANY) INCLUDE ACILIB(ADACIC0) INCLUDE USERLIB(ACIOPT82) ENTRY ADACIC0 NAME ADAENA82(R) </pre>
Create ADABAS82	<p>Link ADACICS with ACINAM82 created in an earlier step:</p> <pre> MODE AMODE(31),RMODE(ANY) REPLACE ACINAMES INCLUDE ACILIB(ADACICS) INCLUDE USERLIB(ACINAM82) ENTRY ADACICS NAME ADABAS82(R) </pre>
Create ADATRU82	<p>Link or copy ADACICT:</p> <pre> MODE AMODE(31),RMODE(ANY) INCLUDE ACILIB(ADACICT) ENTRY ADACICT NAME ADATRU82(R) </pre>

**Notes:**

1. The CICS versioning tool supports callers using both the Direct Call Interface and the EXEC CICS LINK interface. It always uses the DCI to call the various Adabas link modules.
2. When invoked via EXEC CICS LINK, it first looks for the Adabas parameter list in COM-MAREA and then in TWA.
3. Do not use CICS NEWCOPY to reload the versioning tool, versioning table or any Adabas link modules defined in the versioning table.
4. If you use the Adabas System Coordinator Node Error Program, you must be sure to change the sample source so that it starts the CORNEP transaction for each link module.

Database Versioning

Use sample job VERI055 to assemble the database versioning table, VERDBT.

Using the VERDB macro, the database versioning table will have an entry for each installed version of the Add-on products. Here is an example:

```
VERDB SUFFIX=81,VRL=812
VERDB SUFFIX=82,VRL=822
VERDB SUFFIX=xx,VRL=vrs
VERDB TYPE=END
```

The following rules apply:

- The values for SUFFIX= are unique across all VERDB entries in the table.
- The values for VRL= are unique across all VERDB entries in the table.
- The table must end with VERDB TYPE=END.

Refer to [Database Versioning Controls](#) for an explanation of the VERDB controls.

Daemon Versioning

Use sample job VERI060 to assemble the daemon versioning table, VERDMT.



Note: This is an optional activity. You may still run the Adabas System Coordinator and/or Adabas SAF Security daemon from the original product libraries, if desired.

Using the VERDM macro, the daemon versioning table will have an entry for each daemon and its corresponding suffix and version. Here is an example:

```
VERDM SUFFIX=81,VRL=812,JOBNAME=SYSC0812
VERDM SUFFIX=82,VRL=822,JOBNAME=SYSC0822
VERDM SUFFIX=82,VRL=822,JOBNAME=SAFDAEMN
VERDM SUFFIX=xx,VRL=vrs,JOBNAME=daemon-name
VERDM TYPE=END
```

Refer to [Daemon Versioning Controls](#) for an explanation of the VERDM controls.

Final Steps

- [Copy and rename required product modules into the ALLVERS Versioning Library](#)
- [Add the Versioning Libraries to required Job Control](#)

Copy and rename required product modules into the ALLVERS Versioning Library

Refer to the sample jobs VERI080* to identify and customize the one relevant to the version whose modules you wish to copy.

The sample jobs indicate all of the modules that require copying and renaming. No other product modules should be renamed.

Add the Versioning Libraries to required Job Control

The ALLVERS Versioning Library must be added to the Joblib or Steplib concatenation of all Adabas database start-up jobs that require versioning support. The library should be added at the top of the concatenation.

If you want to run the System Coordinator or SAF Security daemon from the same versioning load library, you should also add the ALLVERS Versioning Library first in the daemon Joblib or Steplib concatenation.

The ALLVERS Versioning Library and the Client Versioning Library must be added to the top of the Steplib concatenation for any batch jobs that require versioning support.

For CICS, the Client Versioning Library and the ALLVERS Versioning Library must be added to the DFHRPL and Steplib concatenations.

Installation Controls

The Versioning Tool operation is controlled by the following types of controls:

- [Client Versioning Controls \(Batch\)](#)
- [Client Versioning Controls \(CICS\)](#)
- [Database Versioning Controls](#)

- [Daemon Versioning Controls](#)

Client Versioning Controls (Batch)

This section lists the VERCL controls which can be used for Default Settings.

Control	Overridable
JOBTYPE	-
OVERRIDES	-
WTO	-
LNKGBLXX	Y
BINDGBLS	Y
DBID	Y
SVC	Y

This section lists the VERCL controls which can be used for Versioned Link Module Entries.

Control	Overridable
VRL	-
SUFFIX	-
DEFAULT	-
LNKGBLXX	Y
BINDGBLS	Y
DBID	Y
SVC	Y

This section lists the VERCL controls which can be used for Specific Jobname Entries.

Control	Overridable
JOBNAME	-
LNKGBLXX	-
BINDGBLS	-
DBID	-
SVC	-

Client Versioning Controls (CICS)

This section lists the VERCL controls which can be used for Default Settings.

Control	Overridable
JOBTYPE	-
COFF	-
WTO	-

This section lists the VERCL controls which can be used for Versioned Link Module Entries.

Control	Overridable
VRL	-
SUFFIX	-
DEFAULT	-

This section lists the VERCL controls which can be used for Specific Transaction-ID Entries.

Control	Overridable
TRAN	-

Database Versioning Controls

This section lists the VERDB controls which can be used for Database Versioning.

Control	Overridable
SUFFIX	-
VRL	-

Daemon Versioning Controls

This section lists the VERDM controls which can be used for Daemon Versioning.

Control	Overridable
JOBNAME	-
SUFFIX	-
VRL	-

Control Descriptions

This section provides a description of each Versioning control.

- BINDGBLS
- COFF
- DBID
- DEFAULT
- JOBNAME
- JOBTYP
- LNKGBLXX
- OVERIDES
- SUFFIX
- SVC
- TRAN
- VRL
- WTO

BINDGBLS

Control Element	Use	Possible Values	Default
Default Settings	IGNORED:	IGNORED REQUIRED	IGNORED
Versioned Link Module Entry	A dynamically loaded link globals table must be used, if one cannot be found an error is raised.		
Specific Jobname Entry	REQUIRED: A dynamically loaded link globals table is ignored in favor of using a linked-in globals table.		

COFF

Control Element	Use	Possible Values	Default
Default Settings	This parameter is no longer used and will be removed in a future version, until then a value of zero should be specified.	0	none

DBID

Control Element	Use	Possible Values	Default
Default Settings Versioned Link Module Entry Specific Jobname Entry	<p>DDCARD:</p> <p>The value from the ADARUN DBID= parameter is to be propagated to the suffixed dynamic link globals table (if used).</p> <p>Note: ADARUN LNKGNAME= must be used to ensure this propagation.</p> <p>ASIS:</p> <p>The value in the link globals table (whichever one is used) is taken “as is”; any other value (such as ADARUN DBID=) is ignored.</p> <p><i>Number:</i></p> <p>A specific default database number.</p>	DDCARD ASIS <i>number</i>	ASIS

DEFAULT

Control Element	Use	Possible Values	Default
Versioned Link Module Entry	<p>YES:</p> <p>Identifies the default link module. Only one default link module is allowed. Do not define any Specific Jobname Entries or Specific Transaction-ID Entries for the default link module.</p> <p>NO:</p> <p>This link module is only used by the Jobnames or Transaction-IDs listed immediately following it (defined by Specific Jobname Entries or Specific Transaction-ID Entries).</p>	YES NO	NO

JOBNAME

Control Element	Use	Possible Values	Default
Specific Jobname Entry	Specific Jobname Entry:	<i>jobname</i>	none
Daemon Versioning	<p>Identifies the Jobnames that are to use the link module version identified prior to this job-name list (defined by a Versioned Link Module Entry).</p> <p>Daemon Versioning:</p>		

Control Element	Use	Possible Values	Default
	Identifies the jobname of a System Coordinator or SAF Security daemon.		

JOBTYPE

Control Element	Use	Possible Values	Default
Default Settings	Identifies the client versioning job type.	BATCH CICS	none

LNKGBLXX

Control Element	Use	Possible Values	Default
Default Settings	YES:	YES NO	YES
Versioned Link Module Entry	A versioned link globals table is to be used, paired with the versioned link module of the same suffix.		
Specific Jobname Entry	NO: No versioned link globals table is to be used even when using a suffixed link module.		

OVERIDES

Control Element	Use	Possible Values	Default
Default Settings	Allow Default controls to be overridden by Versioned Link Module Entries.	YES NO	YES

SUFFIX

Control Element	Use	Possible Values	Default
Versioned Link Module Entry	Identifies the two character suffix to be used when loading all necessary modules.	Any two characters.	none
Database Versioning	Example:		
Daemon Versioning	A JOBTYPE=BATCH Versioned Link Module Entry with SUFFIX=82 would cause ADALNK82 or ADALNR82 (for re-entrant operation) to be loaded and, if LNKGBLXX=YES, LNKGBL82 or LNRGBL82.		

SVC

Control Element	Use	Possible Values	Default
Default Settings Versioned Link Module Entry Specific Jobname Entry	<p>DDCARD:</p> <p>The value from the ADARUN SVC= parameter is to be propagated to the suffixed dynamic link globals table (if used).</p> <p>Note: ADARUN LNKGNAME= must be used to ensure this propagation.</p> <p>ASIS:</p> <p>The value in the link globals table (whichever one is used) is taken “as is”; any other value (such as ADARUN SVC=) is ignored.</p> <p>Number:</p> <p>A specific default SVC number.</p>	DDCARD ASIS <i>number</i>	ASIS

TRAN

Control Element	Use	Possible Values	Default
Specific Transaction-ID Entry	Identifies the Transaction-IDs that are to use the link module version identified prior to this transaction-ID list (defined by a Versioned Link Module Entry).	<i>Transaction-ID</i>	none

VRL

Control Element	Use	Possible Values	Default
Versioned Link Module Entry	VRL needs to be set accurately because it is used internally by the software to ensure correct operation.	Any supported VRL.	none
Database Versioning Daemon Versioning	<p>For Client Versioning (VERCL): Denotes the version, release and maintenance level of the Adabas link module.</p> <p>For Database/Daemon Versioning (VERDB/ VERDM): Denotes the version, release and maintenance level of the suffixed collection of modules.</p>		

WTO

Control Element	Use	Possible Values	Default
Default Settings	Details of the versioning configuration will be written to the console.	YES NO	YES

Installation Verification

At the end of the installation process, you can perform verification checks to ensure successful installation, as follows:

- [Verify Client Versioning \(Batch\)](#)
- [Verify Client Versioning \(CICS\)](#)
- [Verify Database Versioning](#)

Verify Client Versioning (Batch)

Setting the WTO control to YES will result in job-specific details of the configuration being written to the console enabling verification, as follows:

```
+VER101I Jobname  Link Module Globals  SVC    DBID
+VER101I  -----  -
+VER101I ADA81TST ADALNK81    BINDGBLS ASIS    ASIS
```

If client versioning detects an error (for example, the required versioned link module is not available), it issues a message and rejects all Adabas commands with response 101, subcode 87.

Verify Client Versioning (CICS)

During CICS initialisation you will see a set of ADAK messages for each TRUE, each of which will say “in use by ADABAS link routine ADABAS”. This is because the versioning module is invoked by each PLT routine. The versioning module ensures that the correct TRUE and link module are initialized.

Setting the WTO control to YES will result in full details of the configuration being written to the console enabling verification, as follows:

```

+VER101I  Criteria Link Module
+VER101I  -----
+VER101I  Default  ADABASDF
+VER101I  DEMO     ADABASDE
+VER101I  N426     ADABAS81

```

If client versioning detects an error (for example, the required versioned link module is not available), it issues a message and rejects all Adabas commands with response 101, subcode 87.

Verify Database Versioning

If Database Versioning is active then messages will be issued to the console during nucleus initialization. These can be used to verify that the software is installed correctly, and the correct product/version sets of modules have been located. A sample of such messages follows:

Message ID	Description
VER001I	VERPOP initialized.
VER002I	ADAPOP nn loaded for Version vs .
VER003S	No versioning table found, cannot continue.
VER004W	No versioning products found, calls suppressed.
VER005W	ADAPOP nn not found for Version vs .
VER006I	ADApp p loaded for Version vs .
VER005W	ADApp p not found for Version vs .

