



digital
vision
group

MODCON

TRIPsystem
Product Documentation

End User License Agreement

All rights to this software, its documentation and logotypes of the TRIP product family and software (altogether “Software”) supplied by DVG Operations GmbH (DVG) are exclusively owned by DVG.

The transfer of this Software, solutions or parts thereof requires the prior written agreement of DVG. Furthermore, the customer has the right to use licensed Software and / or process solutions supplied by DVG to the extent specified in his contract with DVG.

The free-to-use non-commercial version doesn't require a prior written agreement with DVG but such customers, organizations and/or third parties agree by using the software and / or solution of DVG to be strongly obliged to keep all rights to this software, documentation and logotypes of the TRIP product family absolutely unfringed and protected.



About this document

This document describes how to use the MODCON utility.

What is MODCON?

MODCON is a system utility that TRIP uses during the upgrade installation process to modify CONTROL records to reflect any database or form modifications made specifically for that release. For example, if new fields are added to CONTROL (as, for example, in V3.0 with the queue manager fields) then those fields have to be added to the on-site definition of CONTROL. Also, any entry and output formats which are affected by these new fields must be updated (for example, the database design screens and the SStatus output format for the queue manager fields).

As you will have noticed, this utility does not prompt for username / password validation when it is run, but all databases and formats end up owned by SYSTEM (with the exception of CONTROL). This is because the MODCON utility uses a "bootstrap" version of the TRIPkernel which does not require user login but allows programs which are written to it to set their own user id after the TRIPkernel has started. For obvious reasons, this version of the TRIPkernel is not documented for external users.

How does MODCON work?

You will have seen files called P_CONTROL.BAF and P_CONTROL.BIF in the TDBS\$SYS directory of the TRIP installation. These files contain a minimal TRIP data dictionary - in fact they are what becomes CONTROL.* in the case of a new installation of TRIP. This minimal data dictionary includes the following elements :

| Type | Name | Description |
|------|----------------|--|
| DB | CONTROL | The CONTROL database |
| DB | SHELLERR | The CCL / TRIPclassic error file |
| DB | SHELLHLP | The CCL / TRIPclassic help file |
| DB | SYFORM_* | TRIPclassic psuedo databases |
| DB | TDBS | TRIPkernel psuedo database |
| DB | TDBSERR | The TRIPkernel error file |
| DB | TDBSHLP | The TRIPkernel help file |
| DB | THES | The Thesaurus template |
| Efo | SYFORM_*.* | TRIPclassic administration entry forms |
| Fmt | TDBS.* | TRIPkernel output formats |
| Proc | SYSTEM.MACRO_* | The MACRO help system |
| User | TDBS | A "bootstrap" user and the owner of the CONTROL db |
| User | SYSTEM | The normal SYSTEM user |

as well as the designs / forms / procs for the demonstration databases ALICE, CARROLL, CORR and THESALI.

What MODCON does is simply to export these elements from P_CONTROL to temporary files in the TDBS\$SCRATCH directory, and then import update them to CONTROL. To do this in a soft coded manner, the TRIPkernel uses a number of procedures (which are also members of the minimal data dictionary, although they don't make their way into the end user's CONTROL) :

| Type | Name | Description |
|------|-------------------|--|
| Proc | SYSTEM.TDBS_DB | Updates TRIPkernel databases and formats |
| Proc | SYSTEM.TDBS_FRM | Updates TRIPkernel output formats |
| Proc | SYSTEM.SYFORM_DB | Updates TRIPclassic pseudo databases |
| Proc | SYSTEM.SYFORM_FRM | Updates TRIPclassic administration entry forms |

Each of these procedures follows a standard format such that each line, or command, is a CCL command IMPort of a specific element (or collection of elements in the case of the more modern elements) which need to be updated or created. For example, the following is an extract from the procedure TDBS_DB :

```
IMPORT UPDATE BASE=TDBSERR FILE=TDBS$SCRATCH:TDBSERR.DEF
IMPORT UPDATE BASE=TDBSHLP FILE=TDBS$SCRATCH:TDBSHLP.DEF
IMPORT UPDATE BASE=SHELLERR FILE=TDBS$SCRATCH:SHELLERR.DEF
IMPORT UPDATE BASE=SHELLHLP FILE=TDBS$SCRATCH:SHELLHLP.DEF
IMPORT UPDATE BASE=THES FILE=TDBS$SCRATCH:THES.DEF
```

At runtime, the MODCON program modifies the "IMPORT UPDATE" part of the command to "EXPORT" as required.

How do I use MODCON?

As an example, assume that you have created an application for an end user consisting of :

| Type | Name | Description |
|------|-----------------|---------------------------------------|
| DB | APP_MAIN | Main database for the application |
| DB | APP_VALID | Validation lists for the application |
| Frm | APP_MAIN.FULL | Full output for the main database |
| Frm | APP_MAIN.SHORT | Short output for the main database |
| Efo | APP_MAIN.FULL | Full entry form for the main database |
| Sfo | APP_SEARCH | Search form for the application |
| Proc | PUBLIC.APP_INIT | Initialisation proc for search form |
| Proc | PUBLIC.APP_XXX | Procedure for application |

Now you have made a modification to some of the design elements of this application and you want to send those changes to the client using MODCON. The procedure is as follows :

1. Create a temporary directory and define TDBS\$SCRATCH to point to it.
2. Create a procedure, called EXPORT_APP for example, to export all of the above elements to the temporary directory. Store this procedure in your temporary directory, as well as importing it to your CONTROL file.
3. Run the EXPORT_APP procedure from within TRIP to create all the required files.
4. Copy the P_CONTROL files in your installation to your temporary directory, rename them to CONTROL.* and define the logical name TDBS\$CTL / TDBS_CTL to point to that temporary directory.
5. Modify the EXPORT_APP procedure so that it performs IMPort UPDate commands rather than EXPort commands. Import this changed procedure to your new CONTROL file (perhaps with a different name, such as IMPORT_APP).
6. Run the (now modified) procedure so that your new CONTROL file now contains all of your application elements.
7. You can now delete the files associated with your application from your temporary directory as well as deleting the modified procedure from your new CONTROL file.
8. Export the procedure SYSTEM.TDBS_DB from your new CONTROL file to a file in your temporary directory.
9. Edit this file and append the contents of the (modified) EXPORT_APP procedure to it

- so that after importing the TRIP files, it will import your files. Note: it is very important that you use TDBS\$SCRATCH as the location for all files (your files as well as TRIP's).
10. IMPORT UPDATE the procedure SYSTEM.TDBS_DB to your new CONTROL file from the file in your temporary directory.

Therefore, using the example files listed above, your addition to the TDBS_DB procedure would be:

```
IMPORT UPDATE BASE=APP_VALID.* FILE=TDBS$SCRATCH:APP_VALID.DEF
IMPORT UPDATE BASE=APP_MAIN.* FILE=TDBS$SCRATCH:APP_MAIN.DEF
IMPORT UPDATE SFORM=APP_SEARCH FILE=TDBS$SCRATCH:APP_SEARCH.SFO
IMPORT UPDATE PROC=PUBLIC.APP_INIT FILE=TDBS$SCRATCH:APP_INIT.PRC
IMPORT UPDATE PROC=PUBLIC.APP_XXX FILE=TDBS$SCRATCH:APP_XXX.PRC
```

which MODCON will execute by first exporting the named elements from the P_CONTROL file to the scratch file, and then importing updating them to the live CONTROL file.

To follow the example all the way through, the following is a pseudo log of a complete session to perform the items listed above:

UNIX (ksh) :

1. \$> mkdir \$HOME/scratch
 \$> export TDBS_SCRATCH=\$HOME/scratch
 \$> cd \$HOME/scratch
2. \$> cat > EXPORT_APP.PRC
 EXPORT BASE=APP_VALID.* FILE=TDBS\$SCRATCH:APP_VALID.DEF
 EXPORT BASE=APP_MAIN.* FILE=TDBS\$SCRATCH:APP_MAIN.DEF
 EXPORT SFORM=APP_SEARCH FILE=TDBS\$SCRATCH:APP_SEARCH.SFO
 EXPORT PROC=PUBLIC.APP_INIT FILE=TDBS\$SCRATCH:APP_INIT.PRC
 EXPORT PROC=PUBLIC.APP_XXX FILE=TDBS\$SCRATCH:APP_XXX.PRC
 <CTRL-D>
 \$> trip
 CCL> IMPORT PROC=EXPORT_APP FILE=EXPORT_APP.PRC
3. CCL> RUN EXPORT_APP
 CCL> STOP
4. \$> cp \$TDBS_CTL/P_CONTROL.BAF \$HOME/scratch/CONTROL.BAF
 \$> cp \$TDBS_CTL/P_CONTROL.BIF \$HOME/scratch/CONTROL.BIF
 \$> export TDBS_CTL=\$HOME/scratch
5. vi EXPORT_APP.PRC
 :1,\$s/^EXPORT/IMPORT UPDATE/
 :wq
6. \$> trip
 CCL> IMPORT PROC=IMPORT_APP FILE=EXPORT_APP.PRC
 CCL> RUN IMPORT_APP
7. CCL> DELETE PROC=IMPORT_APP
 CCL> STOP
 \$> rm -f APP_*.;*;
8. \$> trip
 CCL> EXPORT PROC=SYSTEM.TDBS_DB FILE=TDBS\$SCRATCH:TDBS_DB.PRC
 CCL> STOP
9. \$> vi TDBS_DB.PRC

```

<Shift-G>
:r EXPORT_APP.PRC
:wq

10.  $> trip
CCL> IMPORT UPDATE PROC=SYSTEM.TDBS_DB FILE=TDBS_DB.PRC
CCL> STOP

```

Now, you are ready to go to your client with your new CONTROL files (P_CONTROL.*) in order to update their installation.

Running MODCON

Once at your client's site, you will need actually to run the MODCON utility in order to register the changes / creations that you have made. In order to do this, you must perform the following actions:

1. Copy your P_CONTROL files to the client's TDBS\$CTL directory, thus overwriting or creating new versions of their existing files. This should not be a problem as you have simply added your files to the normal TRIP distribution set and have not changed how the base kit works.
2. Define the TDBS\$SCRATCH logical, if it is not already defined. MODCON requires this variable and will not work without it.
3. Run MODCON.

For example :

```

UNIX (ksh):

$> cd $TDBS_CTL
$> tar xvf /dev/rmt0
x ./P_CONTROL.BAF, 354879 bytes, 380 media blocks
x ./P_CONTROL.BIF, 102768 bytes, 105 media blocks
$> env | grep TDBS_SCRATCH
$> cat /.triprc | grep TDBS_SCRATCH
$> export TDBS_SCRATCH=`pwd`
$> $TDBS_EXE/modcon

```

The output from MODCON is the same for both operating systems and should look like the example below if all goes well :

```

**** TRIP System Utility MODCON - Modify CONTROL records ****
      Version 8.3-1    4-Nov-2024 14:27:26.13

Verifying migration path...
Processing system database designs...
Processing English forms/formats...
Processing Swedish forms/formats...
Processing Norwegian forms/formats...
Processing German forms/formats...
Processing Finnish forms/formats...
Processing Chinese forms/formats...

```

The line under which your definitions will be processed is "Processing system database designs..." which is when MODCON is running TDBS_DB and SYFORM_DB. The other lines refer to when MODCON is running the language specific versions of the TDBS_FRM and SYFORM_FRM procedures.

Migration path verification

As previously stated, the MODCON utility is used to migrate the CONTROL database design to a newer version as part of an upgrade installation procedure. Downgrading is not supported. If one still attempts a downgrade, it can be dangerous as it can result in a corrupted or unusable CONTROL database. In order to catch such issues before they arise, the MODCON will from version 8.4 perform a verification of the migration path as its first step. This verifies that the CONTROL database can actually be migrated during install. If this verification fails, MODCON will abort without having committed any changes.

You can also run MODCON in a diagnosis mode to show the version information for your current CONTROL and P_CONTROL files. For example:

```
$> $TDBS_EXE/modcon --check

**** TRIP System Utility MODCON - Modify CONTROL records ****
      Version 8.4-0      7-Nov-2024 16:50:44

Version information for /var/lib/trip/ctl/CONTROL.BAF
  Current version: 8.4-0:0
  Compatible from: 8.4-0:0

Version information for /var/lib/trip/ctl/P_CONTROL.BAF
  Current version: 8.4-0:0
  Compatible from: 8.4-0:0

Migration of current CONTROL is supported.

Elapsed: 00:00:00
```

Or, if you want to check the version of a specific CONTROL.BAF or P_CONTROL.BAF file that currently is not in the TDBS_CTL directory:

```
$> $TDBS_EXE/modcon --check --version -f /var/lib/trip/backup/CONTROL.BAF

**** TRIP System Utility MODCON - Modify CONTROL records ****
      Version 8.4-0      7-Nov-2024 16:50:44

Version information for CONTROL.BAF
  Current version: 8.4-0:0
  Compatible from: 8.4-0:0
```

The displayed version information can in some cases state “unknown (older than 8.4-0)” instead of a specific version number. In this case, the CONTROL file is from a TRIPsystem version older than 8.4-0.