Migrating to the Progress V9 Database White Paper

Document History:
The following identifies all of the significant changes to this document:

Version	Date	Comment
1.0	Feb 1999	First Version
1.1	Jan 2000	V9.1A update. Section 5.9
2.0	Apr 2000	Edits Sections 3, 4.4, 5.3, 5.6, 7.1, 7.2 Added Appendix 2.

Table of contents

1. Introduction **4**

2. Assumptions **4**

3. Overview **4**

4. Planning the migration **5**
4.1. Timed step by step plan 5
4.2. If you have chosen to Dump & Load, include in the plan 5
4.3. If you have chosen to use the Conversion utilities include in the plan 5
4.4. If you have to convert from single-volume to multi-volume, include in the plan 6

5. General Considerations. **7**
5.0. Conversion utilities 7
5.1. Databases with 512-byte blocksize. 7
5.2. Startup parameters when Dumping and Loading the database 7
5.3. Binary Dump & Load 9
5.4. Migrating a DataServer Schema Holder 9
5.5. Two-Phase-Commit 9
5.6. Bulkload utility and Rebuilding the indexes 9
5.7. Dump tables exceeding 2Gb 9
5.8. Database migration tool. 10
lew in V9.1 5.9. CONTAINS can fail if -cpinternal is different than the database codepage 10

6. General steps to migrate the database regardless of the database version **12**

7. General migration paths overview **12**
7.1. Dump & Load 12
7.2. PROUTIL Conversion utilities 13
7.3. Conversion from Single-Volume to Multi-Volume 15
7.4. Progress V9 Database Storage Areas. 15
7.4.1. Overview 15
7.4.2. Recommendations on how to use Progress V9 Database Storage Areas. 16

8. Migrating a Progress V6 database **17**
8.1. Dump and Load. 17
8.2. Use PROUTIL conversion utilities 17
8.2.1. For V6 single-volume databases 17
8.2.2. For V6 multi-volume databases 17

9. Migrating a Progress V7 database **19**
9.1. Dump and Load. 19
9.2. Use PROUTIL conversion utilities 19
9.2.1. For V7 single-volume databases 19
9.2.2. For V7 multi-volume databases 19

10. Migrating a Progress V8 database **21**
10.1. Dump and Load. 21
10.2. Use PROUTIL conversion utilities 21
10.2.1. For V8 single-volume databases 21
10.2.2. For V8 multi-volume databases 21

Appendix. Other references to written information **23**

Appendix 2 - Migration Scenarios and Recommendations **24**
Scenario #1 – Version 8.3 database - Single Volume 750 Mb in size (or less) 24
Scenario #2 – Version 8.3 database - Multi-volume 5 Gb in size 24
Scenario #3 – Version 8.3 database - 5 Gb or larger - 1 hour max downtime allowed 24

1. Introduction

This document provides guidelines for converting an existing Progress database to V9. There are several steps that have to be taken into consideration, depending on the version of the database to convert and its platform. This document describes the recommended paths to follow to convert the database. The appendices contain references to additional written information and documentation.

2. Assumptions

This document assumes that you have successfully installed and configured Progress V9 in your system. If you do not have enough disk space to hold both your current and new V9 Progress versions, please make sure that you have backed up your database and your current Progress version directory tree, and that you have verified that you can restore your environment.

3. Overview

You may choose one of two paths to upgrade your database to V9:

- Dump & Load or
- Conversion utilities

The conversion utility conv89 is designed to work with a multi-volume V8 database. The tools to migrate a V6, V7 or V8 single-volume database to a V8 multi-volume database are included with V9. Note, however, that for V6 databases this may be a lengthy process and requires careful planning, therefore you may wish to consider dump & load.

Any Progress database may be dumped and loaded directly into a V9 database.

4. Planning the migration

Having a detailed migration plan is critical for a smooth migration. Many issues have to be considered, including: readiness of the operating system environment, Operating System (O.S.) upgrades, O.S. set-up and tuning, hardware set-up, disk space, go/no-go checkpoints, and the time needed for every step in the migration process.

The timing is particularly important, to have a very good estimate of how long will it take to migrate the database is critical to allocate enough time while minimizing the downtime, as well as to know how long would it take to back up each step in the migration process in case of failure.

After reading this document you should be able to decide what approach best meets your migration requirements.

4.1. Timed step by step plan

Write a step by step plan, including times. The plan should include:

- Review Operating Environment
- Do you have to change your platform ?
- Steps to follow to have the new platform ready to run Progress V9.
- Is it necessary to upgrade the O.S. ?
- Is it necessary to add additional hardware ?
- O.S. Backup and Restore.
- Complete application backup and restore.
- Current Progress version backup and restore (or installation).
- Progress V9 installation and configuration.
- Configuration and environment set-up for the application migrated to V9.
- Step by step action plan.
- Establish GO / NO-GO checkpoints.

Have the media, the serial numbers, control codes and Installation Notes for::

- your current Progress version (V6, V7 or V8)
- the new Progress V9

4.2. If you have chosen to Dump & Load, include in the plan

Step by step actions and time to:

- dump the database
- create the new V9 void (empty) database
- configure the new V9 database (collation tables, code-pages, etc.)
- load the schema information
- load the data
- rebuild indices

4.3. If you have chosen to use the Conversion utilities include in the plan

Step by step actions and time to:

- Perform each conversion (eg. conv68 plus conv89)

4.4. If you have to convert from single-volume to multi-volume, include in the plan

Step by step actions and time to:

- Create the multi-volume structure
- Configure the new multi-volume database (collation tables, code-pages, etc.)
- Load the schema information, Load the data and Rebuild the indices **or**
- Procopy from the old to the new version **or**
- Restore (prorest) a Progress Backup (probkup) of the database to multi-volume

The most straight forward path is to backup (probkup) the single-volume database, create a new void multi-volume structure and restore the backup into the multi-volume structure.

5. General Considerations.

There are a number of utilities, product capabilities and features you need to consider as you plan to convert your database to Version 9, regardless of what version are you currently using.

5.0. Conversion utilities

In order to make the migration to a V9 database as easy as possible, the following database conversion utilities are shipped with Progress V9:

- To convert a V6 to V8 database: proutil conv68
- To convert a V7 to V8 database: proutil conv78
- To convert a V8 to V9 database: proutil conv89
- To truncate V6, V7 or V8 Before Image file: proutil truncate bi for V6, V7 and V8
- To manage multi-volume V6, V7 or V8 databases: prostrct for V6, V7 and V8

The listed utilities are in the following V9 subdirectories:

Unix	Windows
$DLC/bin/82dbutils	%DLC%\bin\82dbutils
$DLC/bin/83dbutils	%DLC%\bin\83dbutils

The README files in these directories are a detailed explanation about the purpose and usage of all the utilities.

5.1. Databases with 512-byte blocksize.

The products-platform combinations with 512-byte database blocksize are:

- V8.1 or lower PROGRESS for Windows (16-bit)
- V7 or lower PROGRESS for DOS
- V6 PROGRESS for OS/2
- V6 and V7 PROGRESS NLM releases

V9 is not available for any of the listed platforms. If you want to migrate to V9 you will have to port your database to UNIX, WinNT, Win95 or Win98 using Dump and Load. *Refer to the MarketWatch online and the Product Life Cycle online for the list of platforms where V9 is available. (documents available in* http://partners.progress.com).

If your are unsure about the blocksize of your database and you have a multi-volume database, you can find the blocksize using the utility *prostrct statistics db-name*.

5.2. Startup parameters when Dumping and Loading the database

For detailed information regarding Dumping and Loading databases, refer to the manuals System Administration Reference and System Administration Guide.

It is very important to make sure that the startup parameters used when dumping and loading the database are the same in order to avoid problems due to inconsistent numeric formats, date format, date century and code-pages.

-E European Numeric Format (-E) tells PROGRESS to interpret commas as decimal points. Use decimal points as commas when displaying or prompting for numeric values.

-d The format in which to display dates in an application. The argument is a three-character string, comprised of the letters d, m, and y in any order.

Note that -d sets the display format, not the storage format, which is fixed. Furthermore, date constants entered in procedures, or as initial values in the Data Dictionary, are always specified in month/day/year format.

-yy Century parameter

PROGRESS applications may use two–digit date formats for the year. The -yy default of 1950 determines the start of the 100–year period in which the two–digit DATE value is defined. When -yy is set at 1950, PROGRESS determines if the two–digit value in DATE is greater or less than 50. If the DATE value is greater than 50, PROGRESS writes the date for the twentieth century. If the DATE value is less than 50, PROGRESS writes the date for the twenty–first century. For example. if you start PROGRESS with -yy 1950, years 50-99 are treated as 1950-1999, and years 00-49 are treated as 2000-2049.

The default is 1900 for V7 or previous versions.
The default is 1950 for V8 or later versions.

Also, the -yy parameter is used with calendars not of western origin, such as the Taiwanese calendar.

-cpstream

Stream code page parameter
Name of a code page for your stream I/O.

The Stream Code Page (-cpstream) parameter determines what codepage PROGRESS uses for stream I/O. Character terminals use the code page you specify for -cpstream unless you also specify a value for the Terminal Code Page (-cpterm) parameter. Stream I/O consists of the following elements:

- Terminals (includes TTY terminals, but does not include graphical interfaces or Windows character mode)
- **Data (.d) files**
- READ-FILE/SAVE-FILE/INSERT-FILE methods for the EDITOR widget
- INPUT FROM and OUTPUT TO statements
- All compilable files (.p, .w, .i, etc.)
- Compiler-generated LISTING, XREF, and PREPROCESS files Code page

Please check any internationalization setting (ie. Collation tables, Case tables, etc.) you may use in your database prior the Dump and Load. Refer to the *Internationalization Guide* for further information regarding internationalization settings and their usage.

Finally check the contents and the usage of any of the previous startup parameters in your startup parameter file(s) (.pf) of your application as well as the contents of the default startup parameter file *startup.pf* in the directory DLC.

5.3. Binary Dump & Load

The binary dump and load utilities were introduced with Progress V8.2. They allow one or more tables to be dumped at high speed while the database is online. On today's average type of systems the performance of the load operation can be 1 Gb per hour, and the index build (idxbuild) 1.2 Gb per hour.

If you have Progress V8.2 or V8.3 you can migrate the database to V9 using the Binary Dump and Binary Load methods. To learn more about Binary Dump & Load please refer to the *System Administration Guide* and *System Administration Reference* manuals.

5.4. Migrating a DataServer Schema Holder

To migrate a DataServer database Schema Holder you can either use the conversion utilities (proutil) or the Dump & Load as described above.

Check the document *DataServer Product Migration Guide* for additional information regarding DataServer migration to V9.

5.5. Two-Phase-Commit

You must create and maintain a transaction log (TL) Storage Area for your database in order to use two-phase commit. For more information, see "*The Transaction Log Area*" section in the *System Administration Guide*.

5.6. Bulkload utility and Rebuilding the indexes

For information regarding the Bulkload utility and parameters to use to accelerate the process of Rebuilding indexes, refer to the Progress Knowledgebase:

Solution ID: 15479
Title: How to dump & load a database the fastest way.

To access the Progress Knowledge Base Solutions online go to http://techweb.progress.com

Additional information about Bulkload can be found in the System Administration Guide and System Administration Reference.

Important: For V6 and V7 the Bulkload utility is the fastest way to load the database. For V8.2+ and V9 the fastest method to dump & load a database is using Binary Dump & Load. Refer to Section 5.3 and 7.1.

5.7. Dump tables exceeding 2Gb

For information regarding the procedures to follow to dump a table which dump file exceeds 2Gb, refer to the Progress Knowledgebase:

Solution ID: 17528
Title: 8.2 Binary Dump with Files larger than 2 Gig

To access the Progress Knowledge Base Solutions online go to http://techweb.progress.com

Additional information about Bulkload can be found in the System Administration Guide and System Administration Reference.

5.8. Database migration tool.

The *'8.3 Database Migration Utility from Datacor'* can be downloaded from our web site, with documentation explaining how it works. This utility helps in the process of migrating from the Progress database from V8.3 to V9, spreading the tables into V9 storage areas. This tool is not supported by PSC.

lew in V9.1

5.9. CONTAINS can fail if -cpinternal is different than the database codepage

Version 9.1A supports a new word-break table format. This format corrects problems with clients using different code pages, accessing a *Type = 2* word index. The new word-break table uses the same format originally introduced in V9.0, but the TYPE is now *Type = 3*. The conversion is performed from the word-break table (.wbt) codepage to the client's -cpinternal codepage. All of the word-break tables provided in the V9.1A %DLC%\prolang\convmap or $DLC/prolang/convmap directory, use the new format.

You do not need to change to this format when upgrading existing databases to V9.1A, but it is recommended. Changing to the new format will require rebuilding the word indexes (to rebuild the indexes use the proutil idxbuild utility). Older word-break table formats are still supported, and their operational behavior is preserved in 9.1A.

For additional information regarding word-break tables and word indexes, refer to: *Internationalization Guide* (section Character Processing Tables, Using Custom Word-break Tables), *System Administration Reference* (section Proutil Utility), and *Programming Hanbook* (section Word Indexes)

Example:

WBT file

```
/*
 *
 * NAME: iso01bas.wbt
 * Progress Word Break Source File for ISO Latin-1 -- ISO-8859-1.
 *
 */

version = 9
codepage = iso8859-1
wordrules-name = basic
type =  3

/* Special word break rules table */
word_attr =
{
    '.',    BEFORE_DIGIT,       /* part of a word only if followed by a digit */
    ',',    BEFORE_DIGIT,
    '-',    BEFORE_DIGIT,
    '',     IGNORE,             /* completely ignore it */
    '$',    USE_IT,             /* always part of a word */
    '%',    USE_IT,
    '#',    USE_IT,
```

```
    '@',  USE_IT,
    '_',   USE_IT,

/* international characters */
/* .../... */
};
```

Compile the WBT

```
proutil -C wbreak-compiler <your-word-break-table-file>.wbt 1
PROWD1=proword.1
export PROWD1
```

Load the word-rules into the database

```
proutil <your-db> -C word-rules 1 [-cpstream <your-cpstream>
                  -cpcoll <your-cpcoll> -cpinternal <your-cpinternal>]
```

Use proutil idxbuild to rebuild the word-indexes:

```
proutil <your-db> -C idxbuild
PROGRESS version 9.1A as of <Date>

Index Rebuild Utility
=====================

Select one of the following:
All  - Rebuild all the indexes
Some - Rebuild only some of the indexes
Quit - Quit, do not rebuild

Enter your selection:
some

Enter the name of table containing the index.

  or enter ! if you do not want to rebuild any more indexes.

Table name:
<table-name-with-word-index>

Index name:
<word-index-to-rebuild>
```

6. General steps to migrate the database regardless of the database version

Before starting the migration of the database and regardless of the version you're migrating from you will have to complete the following steps:

- ?? Shutdown your database and make sure that there's no process accessing it.
- ?? Disable the After-Imaging if you have enabled it.
- ?? Truncate the Before-Image.
- ?? Backup your database.

  Progress recommends that you use the Progress Backup (probkup utility), however if you do not use it, remember to copy all your database files (database, before-image, log-file, after-image and all the database extents if your database is multi-volume).

  It's recommended that you do 2 backups of your database using different media.
  Test the backup copies before proceeding to make sure they are usable should you need to abort the migration process and restore your environment.

7. General migration paths overview

7.1. Dump & Load

Benefits

1. Improved Database Management and Performance. Taking advantage of the V9 database storage areas (refer to Section 11 in this document and the System Administration Guide). Distribute your tables into storage areas prior to load the data in the database.

2. Economize disk space: A database can inefficiently occupy system disk space over time, especially during development phases when frequent deletions of database elements occur. One benefit of dumping and reloading a database is better allocation of disk space.

   Dumping and reloading a database involves essentially creating a new starting version of the database and loading the table contents into this new database. During the loading stage PROGRESS repartitions the disk, removing any gaps created when database elements were deleted, thereby more efficiently allocating disk space.

3. Optimize access: If users repeatedly request data from the same tables in a determined order (for example, to create a certain report), dump and reload the data to optimize it for sequential access. Loading tables in order of average record size results in the most efficient organization of records for sequential access.

4. Optimize data for random access: If your application accesses data in a random order, for example an on-line transaction-processing application, you can dump and reload the data to optimize it for random access. Use a database, with small, fixed-length extents spread across the volumes to balance the I/O across the disks. Use several clients to load data simultaneously, thereby spreading the data across all disks.

Drawbacks

1. The time consumed to perform the operations (dump & load) !

2. You have to have enough disk space to dump the database contents, at least the same size of the database. You can get information about the database internal space utilization using the proutil tabanalys utility.

3. To load the database you have to have enough disk space to hold the new V9 database as well as the files with the dumped schema information and data.

PROGRESS provides three methods of loading table contents: you can use the PROUTIL command to load the data in binary format (only for V8.2+), the Data Administration tool's user interface to load the data in text format, or the PROUTIL command with the Bulk Loader (BULKLOAD) qualifier. Loading table contents in binary format (available in V8.2+) with the PROUTIL command is fastest; however, you can perform a binary load only on database contents that were created with a binary dump.

When you dump a database, you must first dump the database or table definitions, and then dump the table contents. The definitions and contents must be in separate files and cannot be dumped in one step. You perform both procedures with the Data Administration tool if you are using a graphical interface, or the Data Dictionary if you are using a character interface.

You can also dump and load the table contents with the PROUTIL command. This option dumps and loads the data in binary format. The binary dump and load method offers better performance.

First dump the schema definitions including all its tables, fields, indexes, and auto-connect records using the Dump Data Definitions tool in the Data Administration (GUI) or the Data Dictionary (character). If you have hidden tables, unhide them prior to starting the dump, because hidden tables are ignored by the utilities.

Second, dump the data including table contents, sequence values (if you have a V7 or V8 database and you are using sequences in your database), _user table contents (if you have enabled database security) and SQL views.

When the dump is complete create a new empty V9 database, then load the data definitions and finally load the data. If you load the database using the Data Administration or the Data Dictionary load utilitites and Progress encounters an error during the load operations of the schema definitions or the data, the error is reported, the operation aborted and the transaction backed out. Check out and fix the cause of the error and repeat the operation (you do not need to re-create the database).

It's recommended to use the proutil bulkload utility to load the data. The bulkload utility loads the data at higher speed than the Data Administration or Data Dictionary load utilities. When using this utility you have to rebuild the indexes after the load is complete. To rebuild the indexes use the proutil idxbuild utility.

Again, if you are working with V8, you can choose the Binary Dump and Binary Load. These utilities dump and load the data at very high speed.

The speed of the dump and load operations depends of the hardware, it's recommended that you time these operations in your environment to have an accurate estimate of the length of time it will take to the dump and load your database.

For a complete description of the dump & load techniques refer to the System Administration Guide, chapter 13 Dumping & Loading.

7.2. PROUTIL Conversion utilities

The proutil conversion utilities convert databases between versions (V6 to V8, V7 to V8 or V8 to V9).

Benefits

1. Very fast.! For V8 databases the conversion is done under 5 minutes !
2. The disk space requirements are lower than for the dump & load
3. Using these utilities require less manual intervention than the dump & load.

Drawbacks

1. There is always a chance that your schema could become corrupt during conversion. The conversion process consists of three phases and once the conversion starts, you cannot recover your database. If the conversion fails after Phase 1 begins, your database cannot be recovered. PROGRESS issues error messages that your database is corrupt. When this happens, you must revert to the backup copy of your database and begin the conversion again.

2. If the database to convert to V9 is single-volume, it has to be converted to multi-volume at least before performing conv89.

3. The conversion utilities do not repartition the disk, nor remove any gaps created when database elements were deleted, thereby the disk space allocation cannot be expected to be more efficient for the converted database than it was before the conversion is performed.

By default, the conv89 conversion utility places all user tables and indexes in the Schema area. Progress recommends that you create at least one User data area and move your tables and indexes into it. For instructions on how to move your user tables and indexes out of the Schema area into a User data area see the "Adding Storage Areas and Extents" section, then see the "New PROUTIL TABLEMOVE Qualifier," and "New PROUTIL INDEXMOVE Qualifier sections in Chapter 6 "Maintaining Databases".

Exclusively for V6 or V7:

1. The conversion utility from 6 to 8 (proutil conv68) may be long, because the on-disk structure of the V6 databases has to be converted to V8 format.

2. The conversion will require several steps for V6 or V7 databases. They have to be converted to V8 to use the conv89 utility.

For detailed information about the PROUTIL conversion utilities, refer to the System Administration Reference.

- proutil conv68: The CONV68 qualifier converts a V6 database to V8.
- proutil conv78: The CONV78 qualifier converts a V7 database to V8.
- proutil conv89: The CONV89 qualifier converts a multi-volume Version 8 database to a Version 9.0 database.

You can only use the CONV89 qualifier to convert a multi-volume Version 8 database to version 9.0. If you want to convert a single-volume Version 8 database to Version 9.0, you must first convert it to a multi-volume Version 8 database. For more information, see Chapter 6, "Maintaining Databases," in the Version 9 PROGRESS System Administration Guide.9.0 Databases.

Also, refer to the System Administration Guide, Section 4.6 Migrating Version 8 Databases to Version 9.

7.3. Conversion from Single-Volume to Multi-Volume

If the database to convert to V9 is single-volume, it has to be converted to multi-volume at least before performing conv89.

Refer to the System Administration Guide of your current version for information about paths to convert a single-volume database to multi-volume.

Refer to the V9 System Administration Guide for information about the V9 multi-volume database structure and storage areas.

The advantages of using V9 Storage Areas are discussed in the next section.

7.4. Progress V9 Database Storage Areas.

Storage areas are the largest physical unit of a database. Storage areas consist of one or more extents that are either operating system files, operating system device raw partitions, or some other operating system level device that is addressed randomly.

Storage areas give you physical control over the location of specific database objects. You can place each database object in its own storage area or place many database objects in a single storage area.

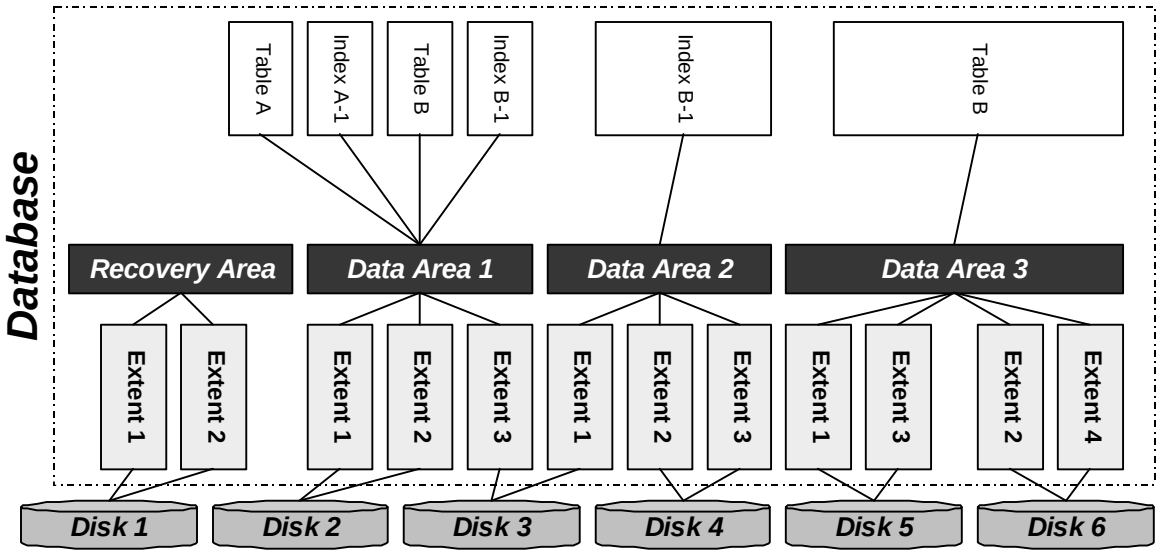
7.4.1. Overview

A feature of the V9 Database is the ability to allocate space for related database objects from a named storage area. CREATE/DROP AREA and ALTER AREA statements are provided to allow new storage areas to be defined and existing storage areas to be removed. A storage area can have as many extents as needed as well as a single variable length extend to ensure that an area never fills up.

By providing this extra level of organization dramatically larger databases can be built as well as effectively managed. In V9 many of the utilities provided use areas to break down work into more manageable chunks and to provide a higher degree of reliability and serviceability.

The CREATE TABLE and CREATE INDEX statements have been augmented to allow a storage area name to be specified so that tables and indexes can easily be placed into an area at create time. Additionally the ability to move a table or an index from one area to another is possible while the system is up and running.

Bellow is a graphical depiction of how Storage Areas are part of a database.



The management of the database Storage Areas : create and delete storage areas, add, delete, and move storage extents (files or raw partitions) is done using the Database Schema Maintenance Tool.

Find more information in the manuals System Administration Guide and System Administration Reference.

7.4.2. Recommendations on how to use Progress V9 Database Storage Areas.

Avoid very small areas, specially if going with self-service or state-reset / state-aware AppServer, as each extra area has an additional file handle to be kept for each local process connected to the database.

For small databases spread across a few disks use less than a dozen storage areas, separating indexes from tables, in the smallest cases, three data areas, one for schema, one for user data and one from user indexes.

Spread tables that get heavy record creation activity across areas. It is a good idea to separate things like order from order-line if order entry is heavy, same thing for other parent-child relationships.

Spread indexes from tables:

1. RECIDs will be packed tighter, which improves non-unique index compression

1. An eventual dump/reload and idxbuild will run faster.

2. Usually table I/O activity will be higher than index I/O activity, so a lot of indexes can be packed together in less areas while more areas could be dedicated to tables. This should be confirmed with promon on your current database. (promon -> R&D-> Activity -> I/O operations by type will give you DB Index Reads, DB Index Writes,DB Data Reads and DB Data Writes separated).

All tables that are larger than 1 Gb should get it's own area.

8. Migrating a Progress V6 database

To migrate your V6 database select the approach that best meets your requirements.

8.1. Dump and Load.

Dump your V6 database using the V6 dump utilities. Then create an empty V9 database and load the schema definitions and the data.

See Section 7, General migration paths, 7.1 Dump & Load

8.2. Use PROUTIL conversion utilities

See Section 7, General migration paths, 7.3 PROUTIL conversion utilities and 7.3 Conversion from single-volume to multi-volume

8.2.1. For V6 single-volume databases

Step 1: proutil conv68
Step 2: Convert the single-volume V8 database to V8 multi-volume
Step 3: proutil conv89

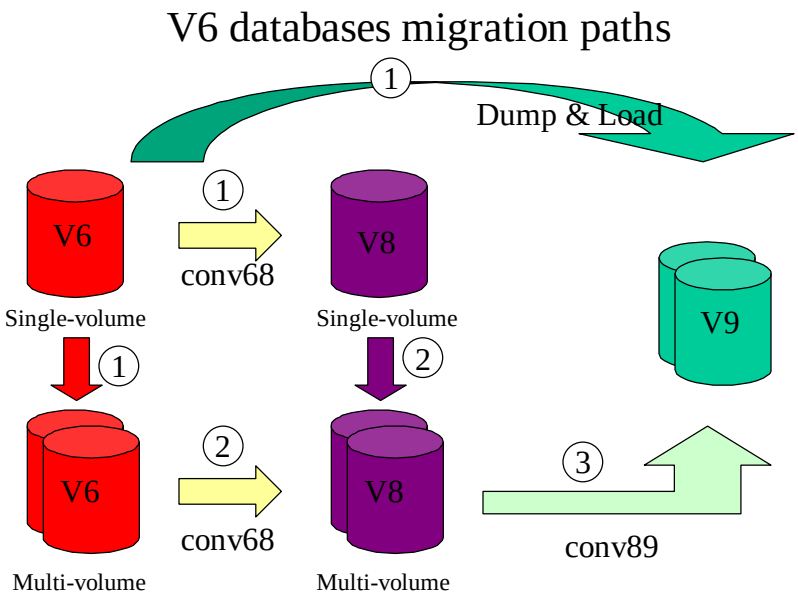
OR

Step 1: Convert the single-volume V6 database to V6 multi-volume
Step 2: proutil conv68
Step 3: proutil conv89

8.2.2. For V6 multi-volume databases

Step 1: proutil conv68
Step 2: proutil conv89

Refer to the Section General Consideration for information regarding databases with 512-byte blocksize.



9. Migrating a Progress V7 database

To migrate your V7 database select the approach that best meets your requirements.

9.1. Dump and Load.

Dump your V7 database using the V7 dump utilities. Then create an empty V9 database and load the schema definitions and the data.

See Section 7, General migration paths, 7.1 Dump & Load

9.2. Use PROUTIL conversion utilities

See Section 7, General migration paths, 7.3 PROUTIL conversion utilities and 7.3 Conversion from single-volume to multi-volume

9.2.1. For V7 single-volume databases

Step 1: proutil conv78
Step 2: Convert the single-volume V8 database to V8 multi-volume
Step 3: proutil conv89

OR

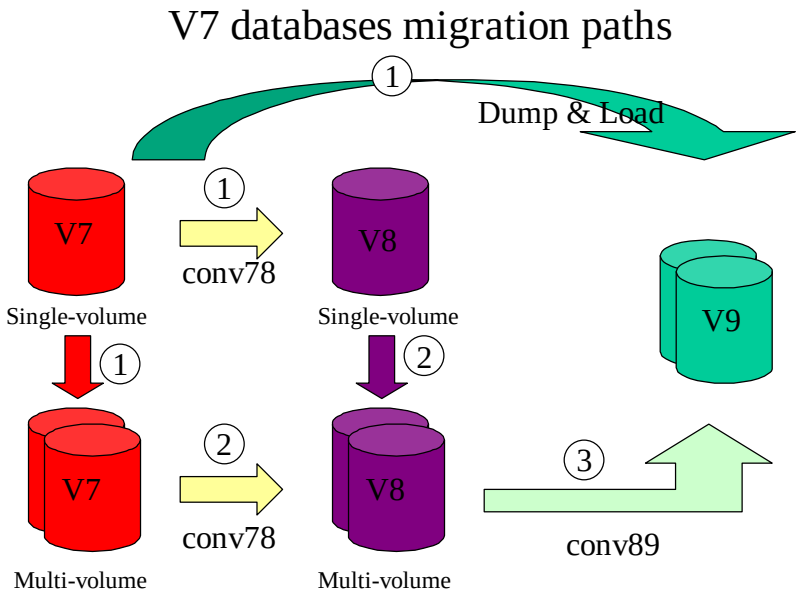
Step 1: Convert the single-volume V7 database to V7 multi-volume
Step 2: proutil conv78
Step 3: proutil conv89

9.2.2. For V7 multi-volume databases

Step 1: proutil conv78
Step 2: proutil conv89

Refer to the Section General Consideration for information regarding databases with blocksize different than 1Kb.

If you are on Sequent on V.4 platforms you must dump and load to convert your database. from version 7 to version 8. This is due to the difference in the blocksizes between V7 Sequent (2Kb) and V8 V.4 (1Kb).



10. Migrating a Progress V8 database

To migrate your V8 database select the approach that best meets your requirements.

10.1. Dump and Load.

Dump your V8 database using the V8 dump utilities. Then create an empty V9 database and load the schema definitions and the data.

See Section 7, General migration paths, 7.1 Dump & Load
If you are using Progress V8.2+ remember that you can use Binary dump & Load as well.

10.2. Use PROUTIL conversion utilities

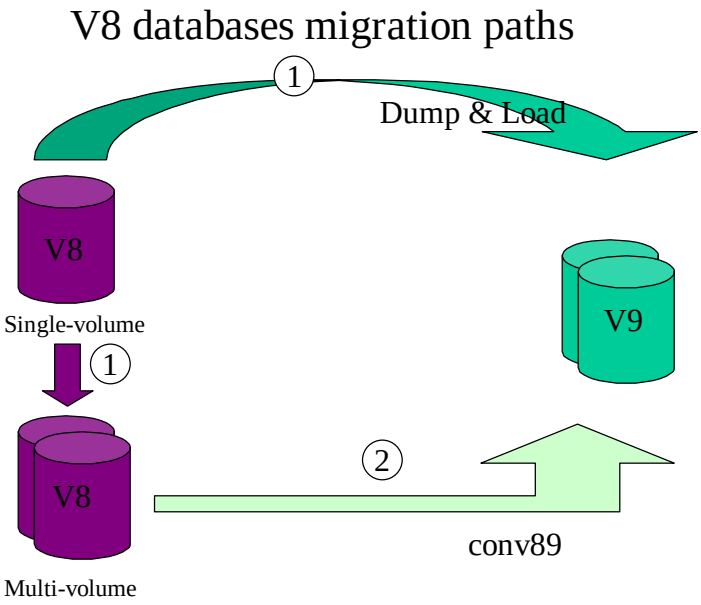
See Section 7, General migration paths, 7.3 PROUTIL conversion utilities and 7.3 Conversion from single-volume to multi-volume

10.2.1. For V8 single-volume databases

Step 1: Convert the single-volume V8 database to V8 multi-volume
Step 2: proutil conv89

10.2.2. For V8 multi-volume databases

Step 1: proutil conv89



Appendix. Other references to written information

To access the Progress Knowledge Base Solutions online go to http://techweb.progress.com

Progress Knowledgebases:

Solution ID	Title
15479	How to dump & load a database the fastest way.
17528	8.2 Binary Dump with Files larger than 2 Gig
18168	What are Storage Areas ?
16673	Steps to convert v6 database to v8 (proutil -C conv68)
17201	PROUTIL CONV68 for version 6.2 - doc bug in Migration Guide
16610	How to use conv78
18006	Conversion not support for blocksize error when conv78
17530	How to probkup/prorest a db to disk > 2GB (Gig)
14061	Dump and Load .df and .d outside of Data Dictionary
17590	SAMPLE CODE to Dump Load .df, .d w/ Data Dictionary routines
12564	How to dump & load data of a larga database with no diskspace
16198	SAMPLE CODE TO CREATE DUMP LOAD PROCEDURES run-time runtime

Documentation:

System Administration Reference
System Administration Guide

and Release Notes

Contact your local Progress representative for details regarding Progress consulting packages available to help you in the Migration of your database and to arrange Stand-By Support Service outside the normal support hours with our Technical Services Center (North America, EMEA or Asia/Pac)

Appendix 2 - Migration Scenarios and Recommendations

Scenario #1 – Version 8.3 database - Single Volume 750 Mb in size (or less)

(Note: Databases of 500Mb or larger should be multi-volume.)

Migration Path
- Convert database to Multi-volume
- Convert the database to Progress V9

Steps to Migrate

1. Backup the database
2. Create a multi-volume database structure
3. Restore the database
4. Convert to Progress V9 using the conv89 utility
5. Backup the database

Benefits:
- Fast – The total time to convert is determined by the time it takes to backup, restore, and backup once you are converted.

Drawbacks:
- No use of Storage Areas – All the date resides in a single data storage area. This is not a major issue since the database is not very large to begin with.

Scenario #2 – Version 8.3 database - Multi-volume 5 Gb in size

Migration Path
- Dump the database in version 8
- Load the database into different storage areas in V9

Steps to Migrate
1. Backup the database
2. Dump the database using the appropriate method (Dictionary, Binary, etc)
3. Create a V9 database with storage areas, and assign tables and indexes to the storages areas as needed
4. Load the data into the database
5. Backup the database

Benefits:
- V9 database utilizes storage areas for ease of maintenance
- Database is freshly reorganized

Drawbacks:
- Slower than Scenario #1 – Dumping and loading is a time intensive operation, however, with storage areas dumping and loading data becomes less needed and much easier to perform in the future.

Scenario #3 – Version 8.3 database - 5 Gb or larger - 1 hour max downtime allowed

Migration Path

- Convert the database using the conv89 utility

Steps to Migrate

1. Backup the database
2. Run the conv89 utility
3. Backup the database
4. Over time, migrate data to new storage areas by either dump and load or table-move.

Benefits:

- Fast – Database is converted in the amount of time it takes to perform 2 backups

Drawbacks:

- No use of Storage Areas – The use of storage areas will help with future maintenance. You will need to migrate to storage areas over time.