DEC Talk

The VAXorcist

Intro to the DECdoc Database

Table of Contents

1. Introduction

I'm a collector of ***DEC*** hardware, software & manuals.

Sometimes collections grow so big that you run the risk to loose track of your collections' items. Then it's time to look for a tool to manage the "chaos"!

Example (my collection as of yet):

701 hardware items

510 software media items

5.810 manuals (paper & pdf)

2. An Example

Imagine for example the ***DSV11 Dual Synchronous Serial QBus Interface*** option.

The following questions might arise:

- When was the ***DSV11*** introduced?
- Which variants of the ***DSV11*** do exist?
- Which module(s) make the ***DSV11***?
- Which other parts belong to the ***DSV11*** option?
- Which cabinet kits are there for the ***DSV11***?
- Which values (e.g. supply voltage, maximum supply current, Qbus loads, baud rates, ...) are specific to the ***DSV11*** option?
- How many ***DSV11s*** are in my collection, where are they from and in what condition?
- Where are my unused ***DSV11s*** stored?
- Into which of my computer systems are ***DSV11s*** configured?
- Which manuals (and other documents) describe the ***DSV11***?

- Which of the ***DSV11*** manuals do I own?
- Which of my ***DSV11*** manuals are printed and which are scanned to pdf or other file types?
- Where are my ***DSV11*** manuals stored?
- Which versions of ***VMS*** (or other ***DEC*** operating systems) support the ***DSV11***?
- Which driver software is needed for the ***DSV11***?
- Are there License PAKs for the ***DSV11*** driver software?
- Which of the ***DSV11*** driver License PAKs have I got?
- What are the diagnostic software modules for the ***DSV11***?
- Which version(s) of the diagnostic software modules for the ***DSV11*** have I got?
- Which media contain the ***DSV11*** diagnostics and where are they stored?

Of course you can find some of this kind of information by reading the relevant manuals, but what and where are they? Naturally you can go through every shelf and every box, look into all of you computers, but that would take ages!

The ***DECdoc*** database is designed to cover all possible aspects of collecting ***DEC*** items, especially hardware, software and manuals of all kinds. The "***doc***" in ***DECdoc*** means "documenting", not "documents" alone.

DECdoc is a (hopefully growing) ***DEC*** knowledge base as well as a means to manage collections of ***DEC*** items (and related "third party" products).

Special care was taken to include the interrelations between two or more of the main categories (hardware, software, manuals) and/or subcategories.

Data now contained in ***DECdoc*** was shamelessly taken from several sources (e.g. Manx for manuals, Modules field guide for hardware items) as well as extracted from ***DEC*** manuals and other sources e.g. by reading and typing or OCR. There was (and will still be) a considerable amount of manual labor especially with filling the "connecting" tables.

Most of the used data sources lack a strict formatting as well as the important cross connections between the three main categories. Finding special information in conventional sources is normally limited to an unspecific text search.

Up to now ***DECdoc*** is mainly focused on ***PDP-11*** and ***VAX*** hardware, software and manuals. Of course ***DECdoc*** is open for older (e.g. ***PDP-8***, ***PDP-10***) as well as newer ***DEC*** computer architectures (for example ***Alpha***). Even so-called "Third-Party" products are integrated into ***DECdoc***. Famous names in this category are ***EMULEX***, ***DILOG*** and ***PLESSEY*** for example. To be honest, ***DECdoc*** could even be used to handle collections of computers originating from ***DEC***s major competitors: ***IBM***, ***HP***, ***SUN*** and others ;-)

Currently the ***DECdoc*** database runs on a Linux ***PostgreSQL*** server. The front end is built with ***LibreOffice Base*** and connected to the database server via ODBC. ***DECdoc*** is not yet accessible via Internet. It would be nice to have a web-based front end as well, but I'm afraid I'm not really good at that. Maybe out there is another ***DEC*** collector who wants to build the web interface (Apache + PHP + PostgreSQL)?

3. ***DEC*doc Database Design**

Sorry, this paragraph may be understandable for database designers only - so just skip it if you don't understand. Lots of efforts were made designing the ***DECdoc*** database structure. Normalization and Referential Integrity were applied wherever possible, and lots of comments added to make clear how it all works. Well, there are of course a few compromises (e.g. which information is property of versions or variants of items and which is property of items directly?). I'd like other database designers to have a closer look at the ***DECdoc*** structure - improvement suggestions are very welcome! See the whole **`SQL CREATE`** script for that.

There are even thoughts to port ***DECdoc*** to a genuine ***DEC*** database: ***RDB***. And I know a ***VAX*** owner who would probably host such a database. But let's develop ***DECdoc*** a little further and port it later when its more stable. Anyone with experience in porting from ***PostgreSQL*** to ***RDB***?

DECdoc is designed as a multi user database with common data (***DEC*** products) and areas for user-specific collection data. A full database rights management is available to manage concurrent multi-user access. You can use ***DECdoc*** to handle your collection making part of it visible to the public and keeping the rest of it concealed. You can use it to explore other members collections (as far as they're public) and look e.g. for parts, software or manuals you want.

Feel free to download the experimental ***LibreOffice Base*** interface to ***DECdoc*** and see yourself - nothing beats personal experience! You need to set up a PostgreSQL database server, sorry ...

3.1 Copyright

For obvious reasons I do not claim the Copyright for the contents of ***DECdoc***, but I do claim the Copyright for its database design. Everyone can use, copy or modify ***DECdoc*** as desired as long as my initial authorship is passed on. Furthermore (this is a wish only) I'd like to hear about improvements, ports, add-ons and the like.

Now that the formal aspects are finished let's go into ***DECdoc*** details!

3.2 Hardware

Looking at systems, options, modules and parts for example makes it obvious that there are hardware hierarchies. Starting at the top and going downwards the "computer hierarchy" yields:

Architectures (***PDP-11***, ***VAX***, ***ALPHA***, ***ITANIUM***, ...)

Families of computers (***VAX-11/7xx***, ***VAX 8xxx***, ***MicroVAX II***, ***MicroVAX 3xxx***, ***VAX 6xxx***, ...)

Computer models (***VAX-11/780***, ***VAX 8200***, ***MicroVAX 3600***, ***VAX 6420***, ...)

Computer systems (= computer models with equipment e.g. as offered per ***DEC*** price lists)

Options (complete functional sub units, often consisting of e.g. modules, cables and cabinet kits)

Modules / Assemblies

Individual parts

Traversing the "devices hierarchy" produces:

Families of devices and related controllers (***RAxx*** disks, ***RDxx*** disks, ...)

Device models (***RA60***, ***RD54***, ...)

Device "systems" (e.g. ***RA60-FA*** = ***RA60-AA*** W/4-HI CAB 120V/60HZ)

Device options

Modules / Assemblies

Individual parts

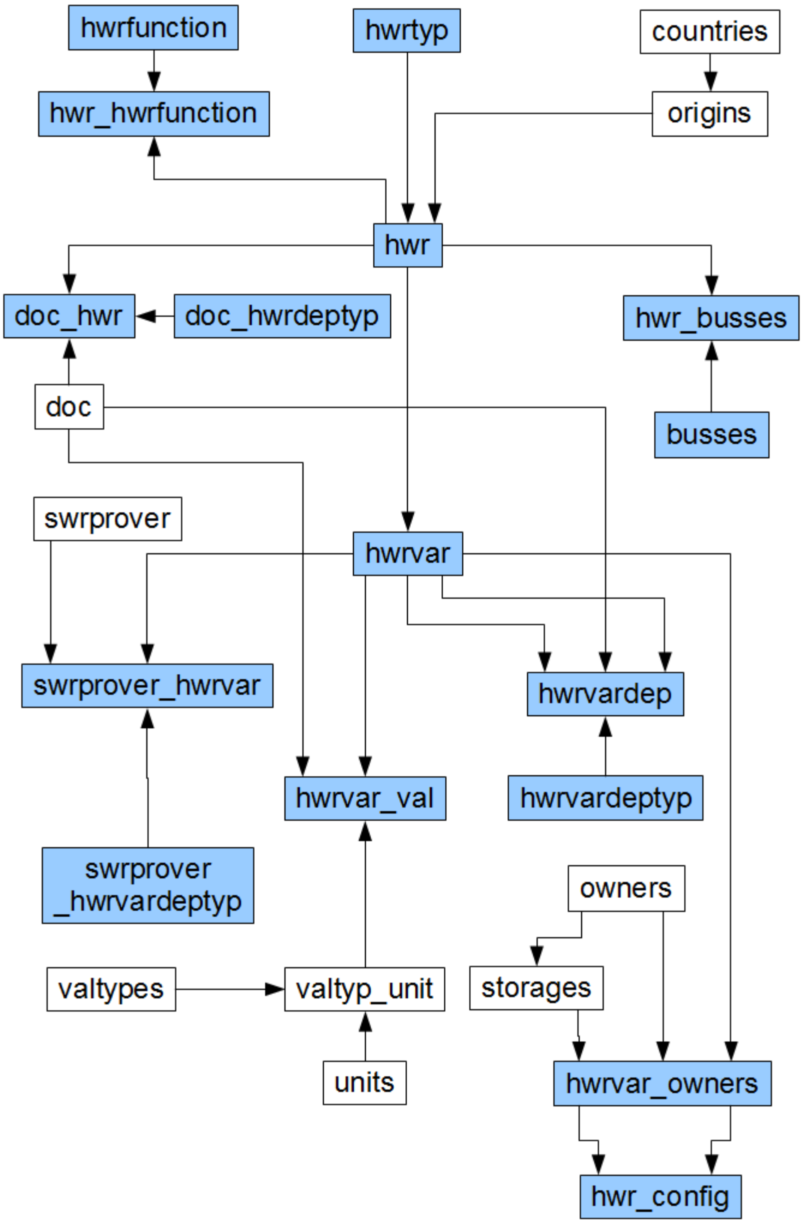
Hardware items do have variants, identified by one or two digits following the ***DEC*** number separated by a hyphen, e.g. ***DELQA-M*** or ***DELQA-SA***. Every hardware item has got the special variant "***-00***" which tags the "standard-variant". Even if it seems that there are only other variants than the standard one, the ***-00*** is still used to store properties common to all variants.

Dependencies between hardware item variants (mostly from different hierarchy levels) are stored in the table "**`hwrvardep`**" (= hardware variant dependencies) together with the type of dependency (taken from the key table "**`hwrvardeptyp`**" (= hardware variant dependency types), for example: "***DELQA-00*** is made of ***M7516-00***" (= "option XXXX is made of module YYYY").

For now the tables covering hardware aspects are:

Table	Usage	# of Records
hwr	hardware items	37.435
hwrvar	hardware item variants	112.742
hwrvardep	hardware item variant dependencies	11.355
doc_hwr	documents for hardware items	12.582
hwrvar_val	hardware item variant values	789
swrprover_hwrvar	software versions for hardware item variants	4.477

The following figure shows the database tables for the hardware section of ***DECdoc*** version 01.4 (light blue background) as well as the accompanying tables from other sections (white background). Remark for database designers: The arrow-head side of the connecting arrows marks the "many" side.



Database tables for the hardware section

3.3 Software

There are a lot of ***DEC*** Software Products (-> table "**`swrpro`**") to deal with, e.g. "***DECNET/E***". Diagnostic Modules are treated as if they were independent Software Products.

Each Software Product (or Diagnostic Module) has possibly got several versions (-> table "**`swrprover`**"), e.g. "***DECNET/E*** V1.0, V1.1, V2.0, V2.1".

The version "0.0" is used to store characteristics common to all versions of a Software Product or Diagnostic Module.

Most versions of a Software Product were packaged on different media, e.g. 9-track tape, disk packs or ***TK50*** tape cassette (-> table "**`swrprover_swrpac`**"):

Sample Software Product Version Packages			
Software Product	**Version**	**Medium**	**Order-#**
DECNET/E	V2.0	***RK05***	AN-M479A-BC
		RL01	MX-M482A-BC
		RK07	MY-M484A-BC
		800 bpi Magtape	AP-M478A-BC
		RK06	AM-M483A-BC
		1600 bpi Magtape	BB-M481A-BC
		RL02	BC-M480A-BC

A version of a Software Product eventually supports or diagnoses certain hardware item variants (-> table "**`swrprover_hwrvar`**").

Examples:

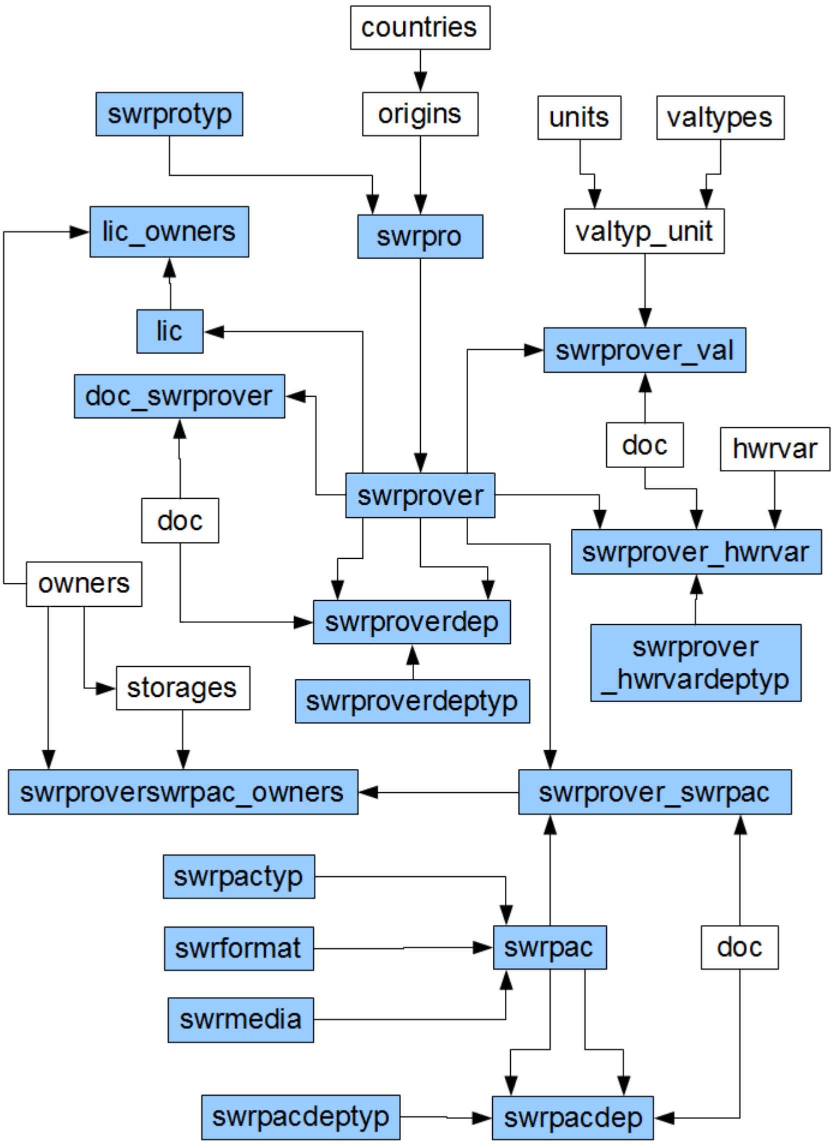
> ***DECNET-11M-PLUS V4.3*** supports the ***DHQ11-00*** as an asynchronous serial network device
>
> ***VHQA V0.0*** is diagnostic software for the ***DHQ11-00***

A software version "***0.0***" designates unknown versions or characteristics common to all versions.

Each version of a Software Product (or Diagnostic Module) usually has got several associated manuals (-> table "**`doc_swrprover`**"):

Sample Software Product Version Manuals			
Software Product	**Version**	**Order-#**	**Title**
DECNET/E	V2.0	AA-H501B-TC	***DECnet/E*** Network Programming in BASIC-PLUS and BASIC-PLUS-2
		AA-H503B-TC	***DECnet/E*** Network Programming in COBOL
		AA-H504B-TC	***DECnet/E*** Guide to User Utilities
		AA-H505B-TC	***DECnet/E*** System Manager's Guide
		AA-J055B-TK	Introduction to DECnet
		AA-K714A-TC	***DECnet/E*** Network Installation Guide
		AA-L265A-TC	***DECnet/E*** Network Programming in MACRO-11
		AA-L266A-TC	***DECnet/E*** Network Programming in FORTRAN
		AA-M269A-TC	***DECnet/E*** Release Notes

The following figure shows the database tables for the software section of ***DECdoc*** version 01.4.



Database tables for the software section

3.4 Documents

DEC made a huge number of documents of all kinds (Manuals, Handbooks, Maintenance Print Sets, ...) for their hard- and software. All documents, whether for hard- or software, are stored in one database table "**doc**". More than twenty-six thousand documents are already covered by the ***DECdoc*** database.

Although there are a lot of documents with different versions, every version of a document is treated as a different document. Some hardware manual versions can be distinguished by the last part of their order numbers, e.g.:

EK-RL012-UG-**004** RL01/RL02 Disk Subsystem User's Guide

EK-RL012-UG-**005** RL01/RL02 Disk Subsystem User's Guide

Some software manuals use letters for version control:

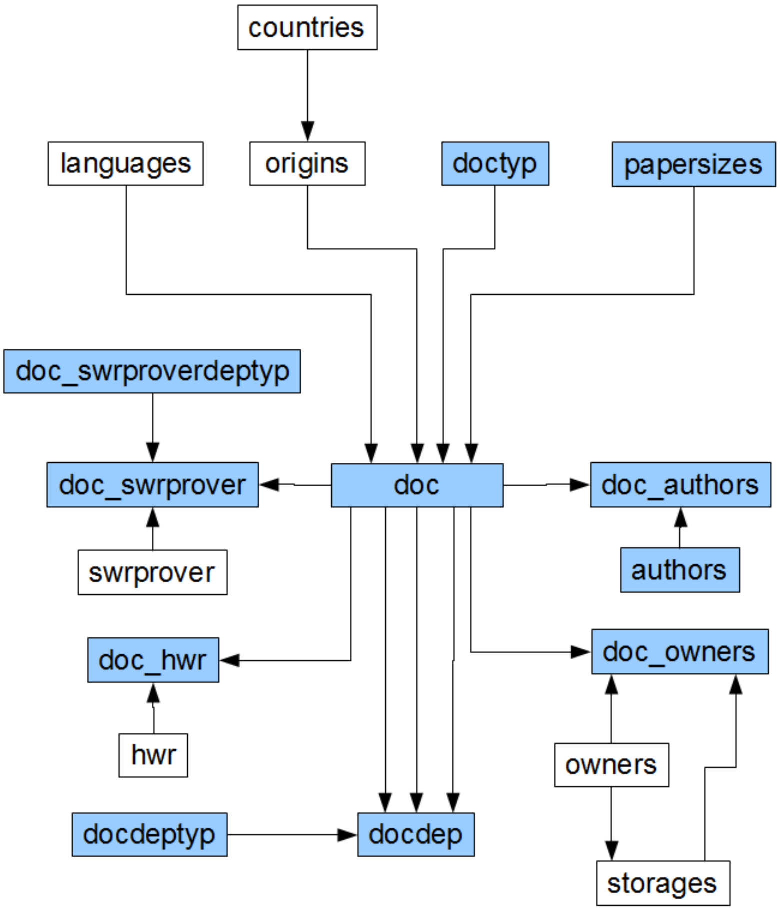
AA-D022**B**-TE ***VAX***/***VMS*** Summary Description and Glossary

AA-D022**C**-TE ***VAX***/***VMS*** Summary Description and Glossary

But these systems are not used consistently and there are documents without any obvious version numbering. Especially with the software documents the structure of the manuals often changes with the software version and completely different order numbers apply. Employing document version numbering in the ***DECdoc*** database however would require a consistent document and version key. While there is one readily available for the hardware items (***DEC*** part/option number), there is apparently no coherent one for the documents.

In terms of database design the sacrifice of distinction between documents and their versions causes some undesirable data redundancy (version independent information about documents has to be stored and managed multiply), but the only alternative would be to create and use a self-made document and version numbering system independent from the ***DEC*** ones. As things are now this is considered the major disadvantage compared to the data redundancy and thus overruled.

The following figure shows the database tables for the document section of ***DECdoc*** version 01.4.



Database tables for the document section

3.5 Summary of DECdoc Database Tables

Table Usages (in *alpha*betical order)		
#	**Table**	**Usage**
1	**`authors`**	Document authors
2	**`busses`**	Types of hardware busses
3	**`countries`**	Countries of origins (meaning firms, institutions or persons making hardware, software or publishing documents)
4	**`diaglevels`**	Types of diagnostic software, e.g.: ***VMS*** only, ***VMS*** or Standalone, Standalone only, ...
5	**`doc`**	All kinds of documents and document sets, e.g binders, books, manuals, handbooks, price lists, etc.
6	**`doc_authors`**	Connection between authors and their writings (who wrote what)
7	**`doc_hwr`**	Dependencies between documents and hardware items
8	**`doc_hwrdeptyp`**	Type of dependency between documents and hardware items (not used yet)
9	**`doc_owners`**	Who owns which document
10	**`doc_swrpac`**	Which document describes which software package or software package set
10	**`doc_swrprover`**	Which document describes which software product version
11	**`doc_swrproverdeptyp`**	Type of dependency between documents and software product versions (not used yet)
12	**`docdep`**	Dependencies between hierarchically superior and inferior documents
13	**`docdeptyp`**	Type of dependency between hierarchically superior and inferior documents, e.g. "is binder of"
14	**`docown_work`**	Work to be done with documents
15	**`doctyp`**	Document types, e.g. binder sets, binders, manuals, books, handbooks, ...
16	**`drawingcode`**	First two digits of ***DEC*** drawing numbers and their meanings

Table Usages (in *alpha*betical order)		
#	**Table**	**Usage**
17	**`hwr`**	Information common to all versions of a hardware item
18	**`hwr_busses`**	Which hardware items have got which busses
19	**`hwr_code`**	First two digits of ***DEC*** part numbers
20	**`hwr_config`**	Owners sub hardware item variants configured into owners super hardware item variants, e.g. system configurations
21	**`hwr_hwrfunction`**	Functions of a hardware item (modules sometimes have more than one function)
22	**`hwr_tests`**	Types of tests performed on specific hardware item variants
23	**`hwr_worktyp`**	Types of work to be done with hardware, e.g. cleaning, testing, ...
24	**`hwrfunction`**	Describes what a hardware item does, e.g. Disk Array, Disk Cartridge, Disk Pack, Disk, ...
25	**`hwrown_work`**	Work to be done with hardware
26	**`hwrtyp`**	Type of hardware, e.g. architecture/series/system/model/device/option/assembly/part
27	**`hwrvar`**	Hardware item variants
28	**`hwrvar_owners`**	Who owns which hardware item variant
29	**`hwrvar_rev`**	Hardware variants and associated revisions (usually found in MP drawing directories DD)
30	**`hwrvar_val`**	Hardware variants and associated values (made from value numbers and value types/units)
31	**`hwrvarown_tests`**	Tests and results performed on specific hardware item variants
32	**`hwrvardep`**	Assemblies consisting of hierarchically superior and inferior hardware item variants
33	**`hwrvardeptyp`**	Type of dependency between hierarchically superior and inferior hardware variant items
34	**`languages`**	Languages used in documents

Table Usages (in *alpha*betical order)		
#	**Table**	**Usage**
35	**`lic`**	***DEC*** PAK licenses (***VMS*** >= V5.0); older licenses see table **`swrpro`** (**`swrprotyp`** = "**`l`**" = license key kit)
36	**`lic_owners`**	***DEC*** PAK licenses and their owners; older licenses & owners see table "**`swrproverswrpac_owners`**"
37	**`origins`**	Firms, institutions or persons making hardware, software or publishing documents - NO authors
38	**`owners`**	Owners of hardware variants, software versions, documents, ...
39	**`papersizes`**	Paper sizes used in documents and drawings
40	**`storages`**	Owner specific storage locations
41	**`swr_worktyp`**	Types of work to be done with software, e.g. imaging, ...
42	**`swrformat`**	File formats, e.g. rt11, ods1, ods2, saveset, dump, tar, ...
43	**`swrmedia`**	Media types like e.g. MT1600, ***RK07***, ***RL02***, ***RX50***, ***RA60***, ...
44	**`swrpac`**	Software products stored on media, either media sets, single media (one or more packages), or single packages
45	**`swrpac_owners`**	Who owns which software packages
46	**`swrpac_worktyp`**	Work to be done with software packages, e.g. imaging, backing up, ...
47	**`swrpacdep`**	Software package assemblies consisting of hierarchically superior and inferior software packages
48	**`swrpacdeptyp`**	Type of dependency between superior and inferior software package
49	**`swrpactyp`**	Software package type; currently one of: media set, (single) medium, (single) software, file group, (single) file
50	**`swrpro`**	Software products like ***VMS***, ***RT-11***, ***VAX-BASIC***, ***DECnet-RSX***, ...
51	**`swrpro_swrlic`**	License types for software products

Table Usages (in *alpha*betical order)		
#	**Table**	**Usage**
52	**`swrprotyp`**	Software product types like diagnostic, operating system, programming language, ...
53	**`swrprover`**	Versions of software products like: ***VMS*** V2.0, ***VMS*** V2.5, ***VMS*** V4.3, ...
54	**`swrprover_hwrvar`**	Which software product version supports which hardware variant
55	**`swrprover_hwrvardeptyp`**	Dependency between software product version and hardware variant
56	**`swrprover_swrpac`**	Software product versions available in software packages
57	**`swrprover_val`**	Software product versions and associated values (made from value numbers and value types/units)
58	**`swrproverdep`**	Dependencies between hierarchically superior and inferior software product versions
59	**`swrproverdeptyp`**	Type of dependency between hierarchically superior and inferior software product versions
60	**`swrproverswrpac_owners` `VIEW???`**	Who owns software product versions (in software packages)
61	**`units`**	Units for values (but independent from value types), e.g. unit A (= Ampere) for current
62	**`valtyp_unit`**	Combined value types and units, e.g. supply voltage (unit volt), max. memory (unit KB) or max. memory (unit MB)
63	**`valtypes`**	Value types (independent from units), e.g. voltage, current or maximum memory supported
64	**`workstatustyp`**	Status of work to be done, e.g. to be done, started, finished, ...