OpenVMS Shareable Libraries: An Implementor's Guide

Session 460

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

Wednesday, October 4, 2000
8:00 am – 9:15 am
Room 504

Compaq Enterprise Symposium 2000
Los Angeles Convention Center
Los Angeles, California

Introduction

Robert Gezelter
Software Consultant

When and Why?

It is well known that shareable libraries make sense in heavily used applications. For example, the OpenVMS Run-Time library is implemented as a series of Shareable Libraries.

Robert Gezelter
Software Consultant

When and Why? (cont'd)

Not as well known are the benefits realized in program development and applications implementation. These benefits are completely user realizeable, and are separate from the traditional, well–known system–wide benefits of using shareable libraries.

Robert Gezelter
Software Consultant

Maintenance

No need to re-link entire program for a change in one routine.

Ability to quickly switch between new and old versions of routines.

Robert Gezelter
Software Consultant

Speed/Efficiency

- ***INSTALLed shareable image***
- ***Read-only pages shared by many processes***
- ***Significant reduction in memory requirements***
- ***Significant reduction in disk storage requirements***

Robert Gezelter
Software Consultant

“Leave No Stone Unturned”

Changes in object libraries require relinking to take effect

Relinking is a major task in a medium/large facility (tens or hundreds of programs)

Robert Gezelter
Software Consultant

Dynamic Code Generation

Permits execution time customization

Highly efficient

Simplifies code

Old tactic; but not well known

Robert Gezelter
Software Consultant

Why use Shareable Libraries?

- ***efficiency/performance***
- ***maintenance/change control***
- ***eliminate regression***
- ***leave "No Stone Unturned" (or program un-relinked!)***

Robert Gezelter
Software Consultant

Why use Shareable Libraries? (cont'd)

- ***different programmers can work on different parts of the project at the same time without interfering with each other.***

Robert Gezelter
Software Consultant

What is an OpenVMS Shareable Library?

A Shareable Library is a section of code and/or data which is dynamically linked to your program at image activation.

Normal usage does not require any privileges not available to a Student user.

Robert Gezelter
Software Consultant

What kinds of code can be included in a Shareable Library?

Almost any code can be placed in a shareable library. The main requirement is that the code be referenced by one or more programs or developers.

Robert Gezelter
Software Consultant

Can data be included in a Shareable Library?

Yes, data can be included in a shareable library.

However, to ensure safety, you should make sure that the data is

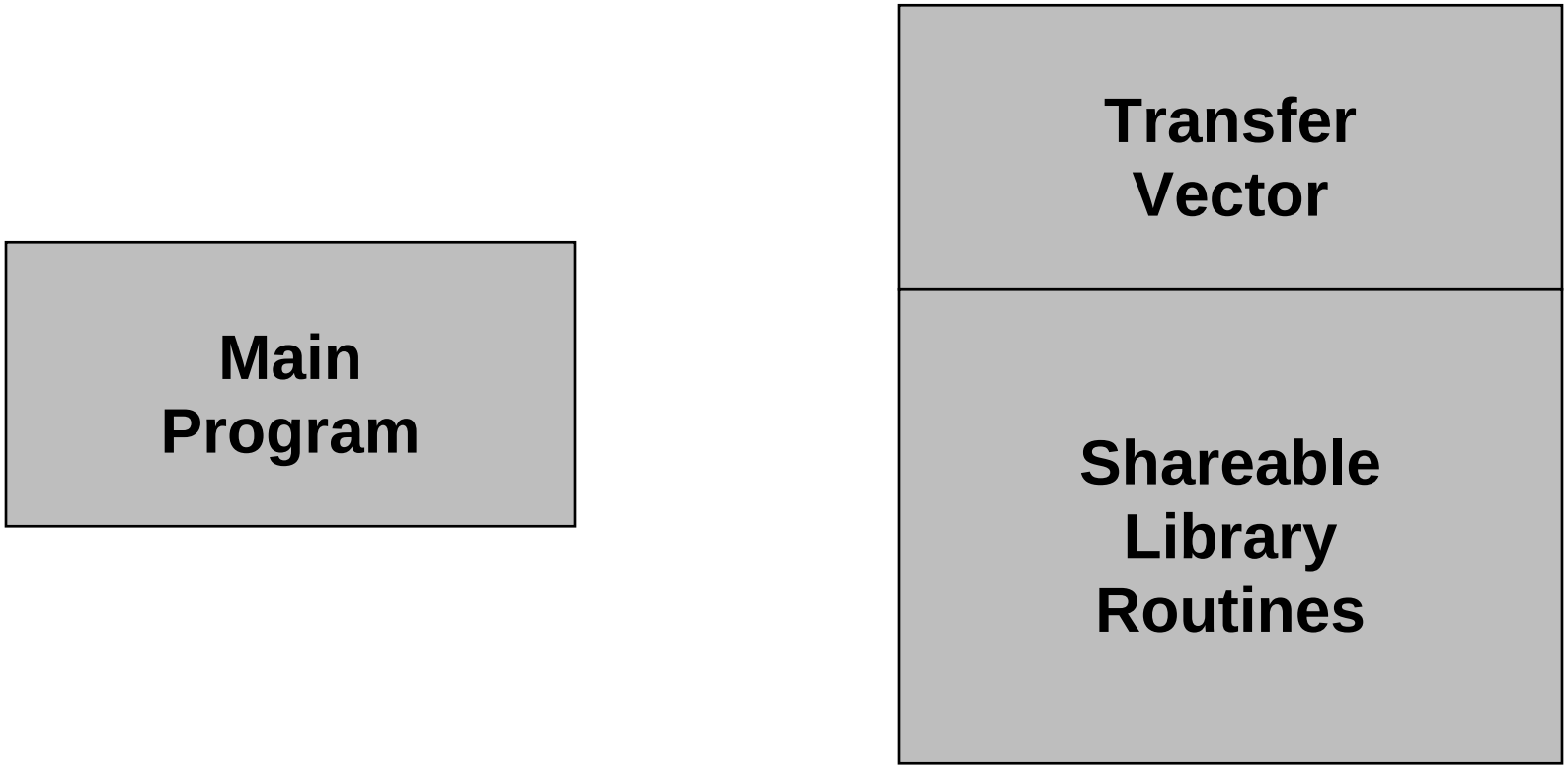
Read only (NOWRT)

OR

Copy on Reference.

Robert Gezelter
Software Consultant

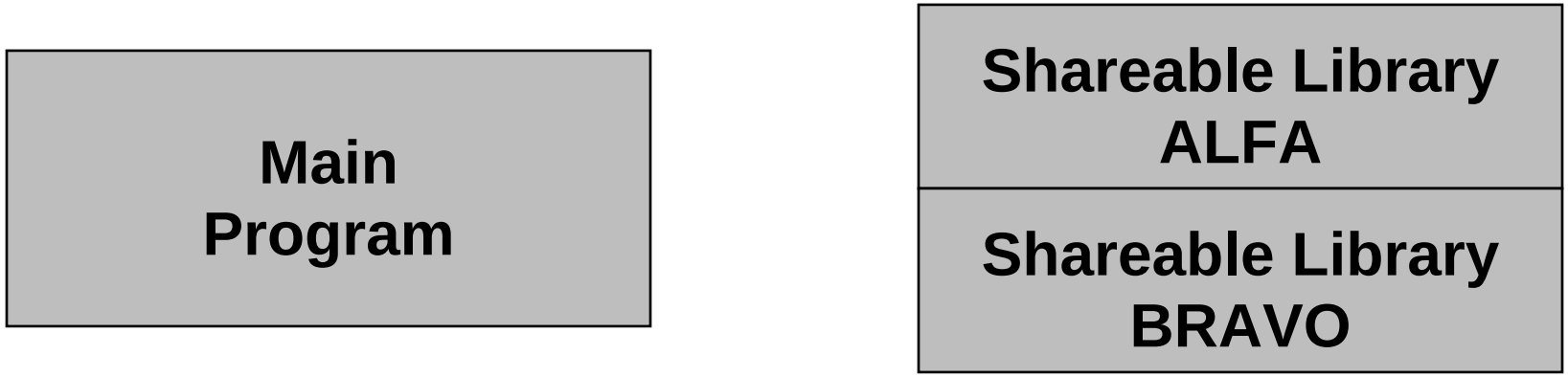
What happens when I call a routine in a shareable library?



Robert Gezelter
Software Consultant

Can I use multiple shareable libraries at the same time?

YES!!!



Robert Gezelter
Software Consultant

How do I specify a shareable library at execution time?

Use logical names.

No privileges required!

```
$ ASSIGN $1$DUA2:[GEZELTER]TEKPLT.EXE TEKPLT
$ RUN program
```


Robert Gezelter
Software Consultant

How do I create an OpenVMS–VAX transfer vector?

Its easy! (Even if you are not a MACRO programmer!)

Define Transfer Vector:

```
.TRANSFER       TEKPLT
.MASK           TEKPLT
JMP             L^TEKPLT+2
.END
```


Robert Gezelter
Software Consultant

How do I create an OpenVMS–VAX transfer vector? (cont'd)

Assemble transfer vector.

LINK the image.

Robert Gezelter
Software Consultant

On OpenVMS–ALPHA Transfer Vectors

In the LINKER Options File:

```
SYMBOL_VECTOR=(name1=PROCEDURE,-
              name2=PROCEDURE,-
              SPARE,-
              SPARE,-
              SPARE,-
              SPARE)
```

Just LINK the image!

Robert Gezelter
Software Consultant

What do I save by using Shareable Libraries?

- ***link time (huge savings possible)***
- ***disk space***
- ***maintenance effort***
- ***regression errors***

Robert Gezelter
Software Consultant

Guidelines:

- ***provide ID entry points***
- ***have main system produce optional revision listing of libraries used***
- ***be careful of multiple versions***
- ***be extremely careful of shareable, writeable data!!!! (JUST SAY NO!)***
- ***enforce use of libraries***

Robert Gezelter
Software Consultant

Shareable Libraries – Concepts

All calls to entry points in shareable libraries are routed through transfer vectors.

Most data areas are allocated as non-shareable space or are located on the stack.

Normal use requires no privileges. Actual sharing of code/data requires the privileges to INSTALL the image.

Robert Gezelter
Software Consultant

Source Program Concerns

Avoid impure references; address constants, use MOVA type instructions instead

Watch out for:
COMMONs (FORTRAN);
external variables (C); and
similar structures

Robert Gezelter
Software Consultant

Compiler related issues

Watch out for PSECT attributes!

In particular, the combination of SHR and WRT is generally a bad idea (when the image is installed, different processes will share Read/Write data).

Robert Gezelter
Software Consultant

Linker related issues

/SHARE switch on command

GSMATCH=LEQUAL,1,0 (in OPT file)

Fix PSECT attributes (if needed)

Be sure to check MAP file

Robert Gezelter
Software Consultant

Debugging Concerns

Try to debug before releasing shareable image to the world.

Local logical names override more global names, thus you can switch between production and test versions from minute to minute.

Robert Gezelter
Software Consultant

Cases from our Files:

We will present two case studies:

Development advantages

Applications tool for dynamic code generation

Robert Gezelter
Software Consultant

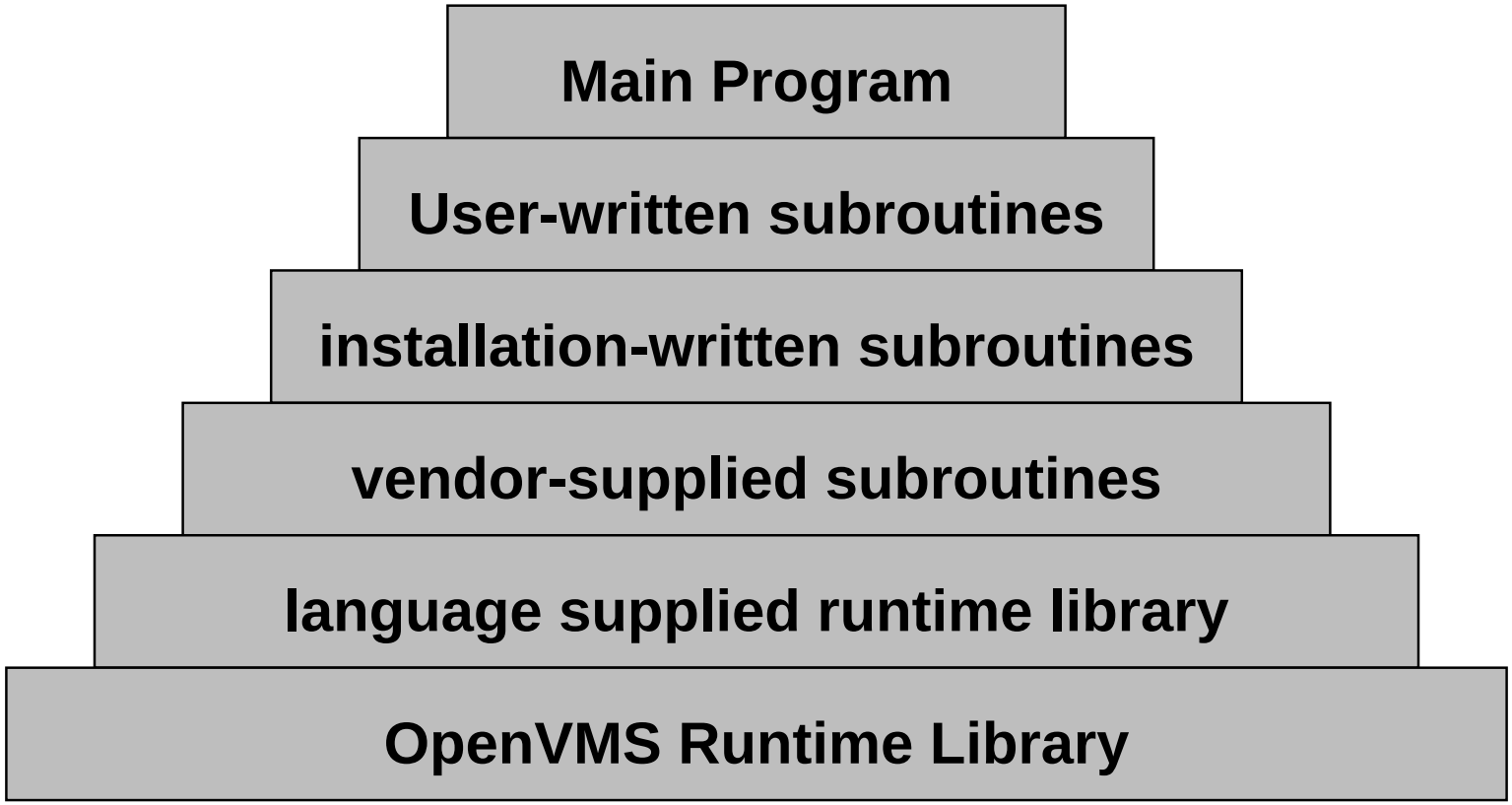
Case 1 – Development

Symptom:
Large Program – Slow Links

Linking this program takes up to 20 minutes on a VAX–11/780

Robert Gezelter
Software Consultant

Problem: Programs are like pyramids – very large foundation



Robert Gezelter
Software Consultant

Solution:

Create one or more user shareable images containing most of the foundation elements.

Result:
Link time reduced to 15 seconds!

Robert Gezelter
Software Consultant

Mechanics of Shareable Libraries

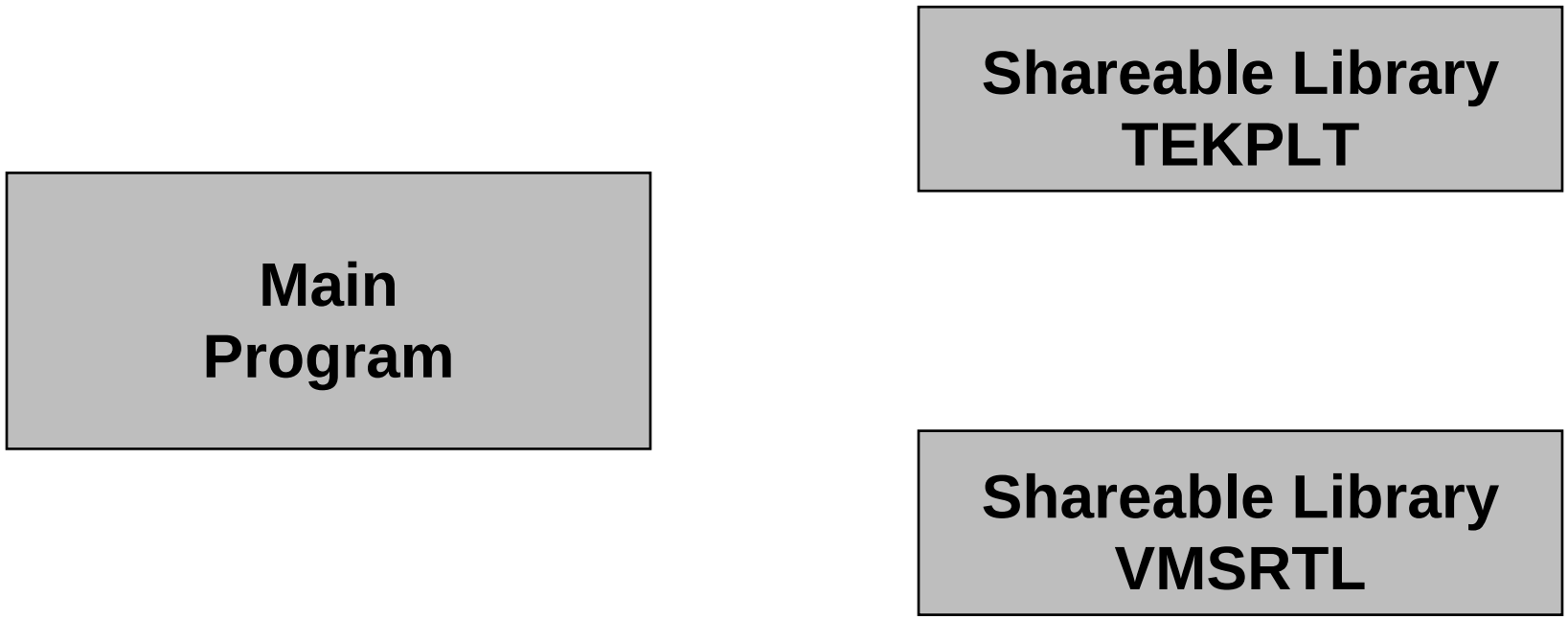
Define Transfer Vector:

```
.TRANSFER       TEKPLT
.MASK           TEKPLT
JMP             L^TEKPLT+2
.END
```

Assemble transfer vector.

Robert Gezelter
Software Consultant

The support code, which is the bulk of the image, is in the shareable libraries!



Robert Gezelter
Software Consultant

Specify the shareable library at execution time

Use logical names.

No privileges required!

```
$ ASSIGN $1$DUA2:[GEZELTER]TEKPLT.EXE TEKPLT
$ RUN program
```


Robert Gezelter
Software Consultant

Case 2 – Dynamic Linking a.k.a. Power T
Interchangeable Heads/Bits

Most programs are written to do a particular job.

How does one write a program to do many different jobs?

With Shareable Libraries, of course!

Robert Gezelter
Software Consultant

Case Subject: Mailing List System

Must generate:

Labels
Envelopes
Form letters
Invitations
Listings
Attendee Lists
...

Robert Gezelter
Software Consultant

Problem: Complexity

Program complexity grows as an exponential (n**m) of the number of different options AND the number of different values of the options

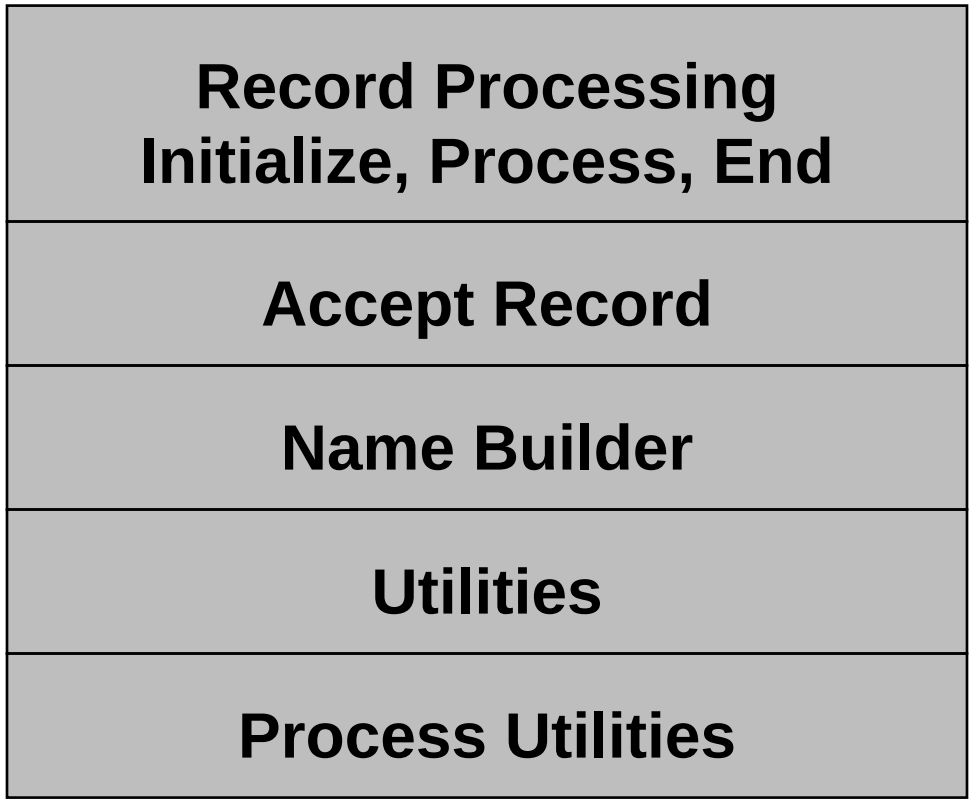
Robert Gezelter
Software Consultant

Complexity

Research has shown that correctness of code is endangered by large numbers of nested IF statemets

Robert Gezelter
Software Consultant

Progamming by Components



Robert Gezelter
Software Consultant

Programming By Chinese Menu

Pick:

1 from Column A

1 from Column B

3 from Column C

Robert Gezelter
Software Consultant

Conventional Programming

Column A: 5 possible choices

Column B: 7 possible choices

Column C: 30 possible choices

TOTAL: 1050 programs
(5 * 7 * 30)

Robert Gezelter
Software Consultant

Goal: Develop a large family of related programs with minimal effort

Maintain separation between different applications

Robert Gezelter
Software Consultant

Programming By Chinese Menu

5 Group A subroutine packages

7 Group B subroutine packages

30 Group C subroutine packages

1 Main Program

TOTAL: 43 programs / packages
(5 + 7 + 30 + 1)

Robert Gezelter
Software Consultant

Conventional Programming vs. Chinese Menu – The Difference

Conventional:

1050 programs

Chinese Menu:

43 modules/packages

3 interfaces

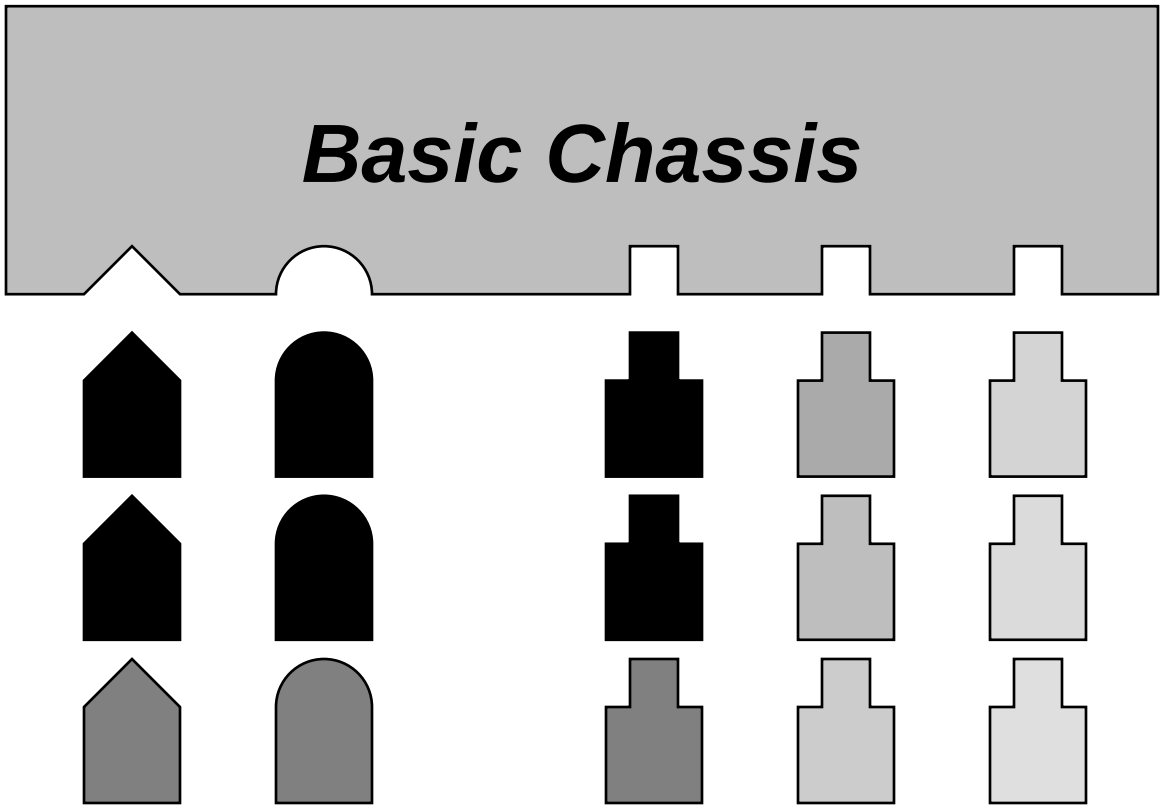
The Difference:

1007 programs!

(or combinations of options)

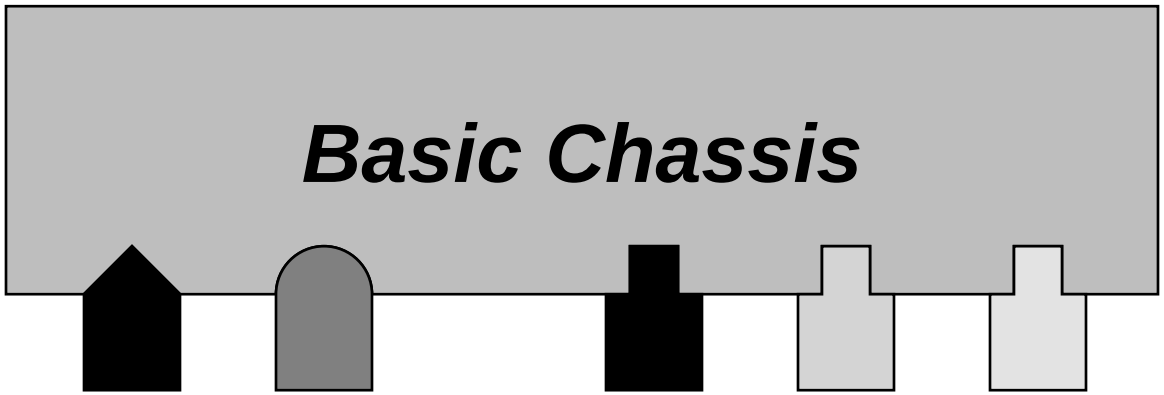
Robert Gezelter
Software Consultant

Key Concept: Programming by Chassis



Robert Gezelter
Software Consultant

Programming by Chassis: Operation



Robert Gezelter
Software Consultant

Result:

Shareable libraries permit us to achieve the effect of multiple levels of nested IF statements without increasing program complexity.

Robert Gezelter
Software Consultant

Production Environment

The selection of components is driven by the menu system. There is little need for multiple levels of IF statements.

Robert Gezelter
Software Consultant

Complexity Solution

By hanging different applications components on the same chassis, we are able to achieve a wide variety of options WITH NO INCREASE APPLICATIONS COMPLEXITY

Robert Gezelter
Software Consultant

Another view:

This way of building applications is conceptually similar to genetics. You build applications (organisms) out of simple building blocks.

Robert Gezelter
Software Consultant

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

Session Notes & Materials:
http://www.rlgsc.com/cets/2000/index.html