Session
AD029

Introduction to AST Programming

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

Wednesday, December 6, 1995
7:00 pm – 7:50 pm
Room 120

Fall 1995 US DECUS Symposium
Moscone Convention Center
San Francisco, California

Agenda –

- ***This session will teach you how to use OpenVMS VAX ASTs***
- ***The rules presented here ARE stricter than many of the rules presented in DIGITAL manuals.***
- ***These rules are designed to ensure correct, efficient applications.***

Robert Gezelter
Software Consultant

Introduction

Robert Gezelter
Software Consultant

NOTES

Basics

- ***Synchronization roughly equivalent to IPL level synchronization in the VMS Executive.***
- ***High Efficiency***
- ***Fewer limits than Event Flags***

Robert Gezelter
Software Consultant

When Should You Use ASTs?

- ***Realtime Applications***
- ***Control***
- ***Transaction Processing***
- ***Monitoring***
- ***Terminal Applications***
- ***Network Applications***
- ***Time related applications***

Robert Gezelter
Software Consultant

NOTES

General AST Concepts

- ***Non-interruptable by other ASTs at same or lesser Access Modes.***
- ***FIFO Execution.***
- ***AST Entry is via an asynchronous(!), simulated, CALLS instruction.***

Robert Gezelter
Software Consultant

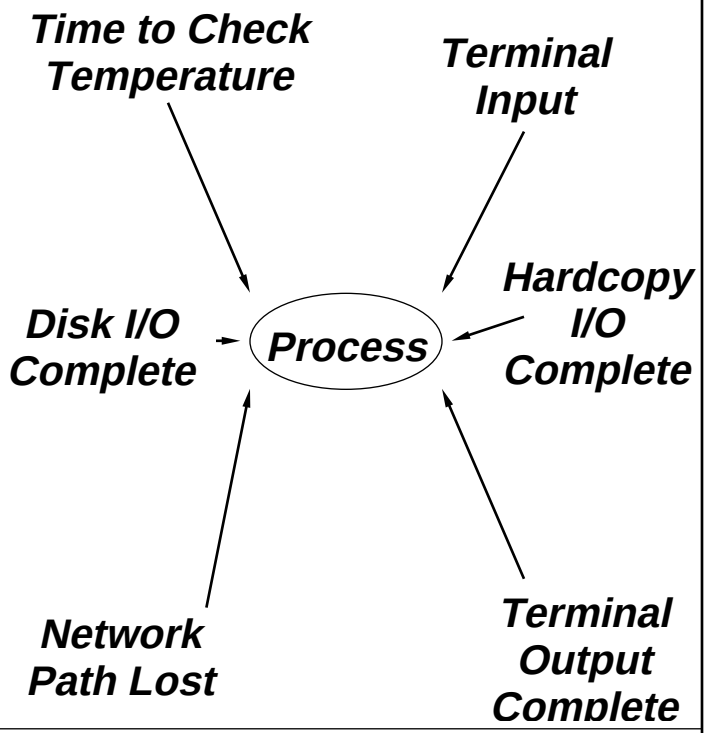
Typical Event Driven Computer Applications

- ***Printing***
- ***Terminal Management***
- ***Process Control***

Robert Gezelter
Software Consultant

NOTES

Typical Event Driven Computer Applicatio



Robert Gezelter
Software Consultant

Common Root —

- ***External events control program***
- ***Programs need to be efficient***
- ***External event sequence is not under program control***
- ***No Dispatch Routine***

Robert Gezelter
Software Consultant

NOTES

Generating and Processing ASTs

- ***Asynchronous System Services***
  - ***$QIO***
  - ***$ENQ***
- ***Record Management Services***
- ***Timer Services ($TIMER)***
- ***Declare AST Service ($DCLAST)***
- ***Mailboxes***
- ***Unsolicited I/O Events***

Robert Gezelter
Software Consultant

Programming Benefits

- ***Event Flags are limited***
  - ***64 Local Event Flags***
  - ***64 Common Event Flags (Remappable)***
- ***No limit on ASTs. AST limits enforced by***
  - ***ASTLIM (from SYSUAF)***
  - ***System Resources***
- ***Capable of supporting multiple, alternative sequences without polling or increases in complexity***

Robert Gezelter
Software Consultant

NOTES

Keep Programs Simple

Best main program for AST based application is extremely simple.

```
PARAMETER NO = 0
CALL INIT
EXIT_FLAG = NO
DO WHILE EXIT_FLAG .EQ. NO
    CALL SYS$HIBER()
    END DO

CALL SYS$EXIT()
```


Robert Gezelter
Software Consultant

Keep Programs Simple

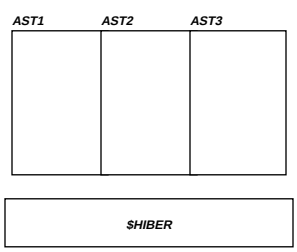
- ***Get in — GET OUT!***
- ***Never use System Service WAIT forms***
- ***Keep Logic simple***

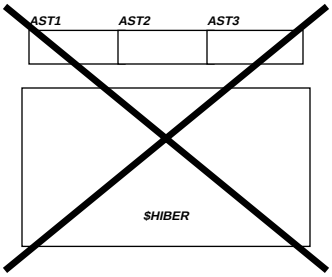
Robert Gezelter
Software Consultant

NOTES

Tricks to Getting It Right

Do ALL Processing in ASTs
Avoid Performing Processing at AST level and normal Process level.





Good ***Bad***

Robert Gezelter
Software Consultant

Tricks to Getting It Right

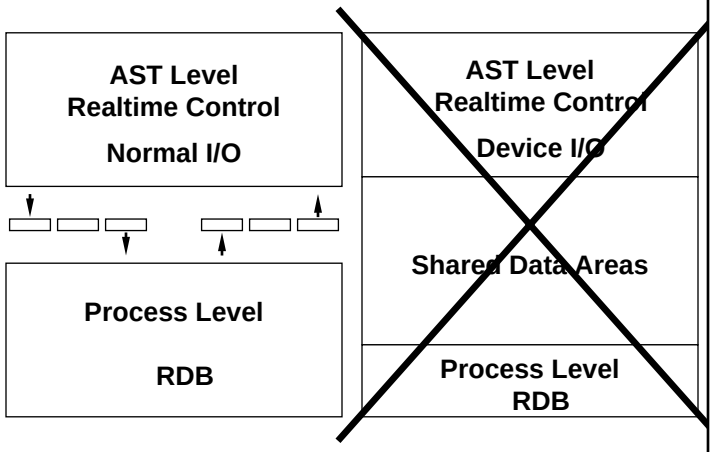
Some packages (e.g. RDB) expect to be used only from normal level, NOT AST level.

Robert Gezelter
Software Consultant

NOTES

Tricks to Getting It Right

Use Work, Answer, and Free Queu to communicate.



Good ***Bad***

Robert Gezelter
Software Consultant

Communications Between AST Level and Process Level

- ***Use Queues, Insert/Remove Queue or LIB$ routines (for HLLs)***
- ***Be careful of queue overflows, handle overflows gracefully***
- ***Remember to ALWAYS issue $WAKE call!***

Robert Gezelter
Software Consultant

NOTES

Communications Between Process Level and AST Level

- ***Use queues, Insert/Remove Queue or LIB$ routines (HLLs)***
- ***Use $DCLAST service to switch to AST level***
- ***Allow ASTs to be processed in the order they are generated, DO NOT process multiple items at a time!***

Robert Gezelter
Software Consultant

Initialization

Do as much initialization as possible from AST level to reduce risk of race conditions.

```
SUBROUTINE INIT
X = SYS$DCLAST(INITAST, PARM)
END

SUBROUTINE INITAST(PARM)
     .
     .
END
```

Good

```
SUBROUTINE INIT
     .
     .
END
```

Bad

Robert Gezelter
Software Consultant

NOTES

Avoid Problems

- ***Kill bugs before they occur***
- ***DO NOT inhibit ASTs. Use $DCLAST to avoid interruptions.***

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

+1 718 463 1079
35 – 20 167th Street, Suite 215, Flushing, New York 11358 – 1731 USA

NOTES