presented by

From Runtime to Compile Time

Improving ASL Through Enhanced Namespace Resolution

Spring 2019 UEFI Plugfest
April 9-11, 2019
Presented by Erik Schmauss (Intel corp.)

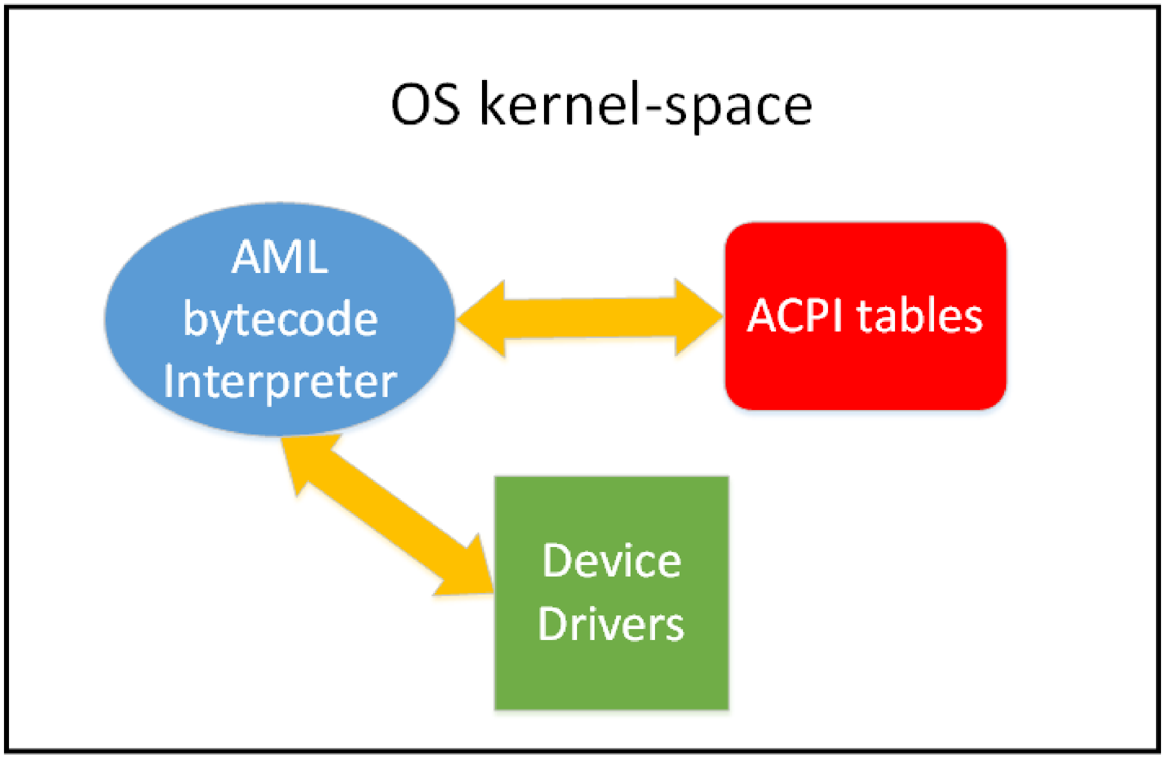
Agenda

- ACPI Overview
- Namespace resolution errors
- Solutions
- Takeaway
- Questions

What is ACPI?

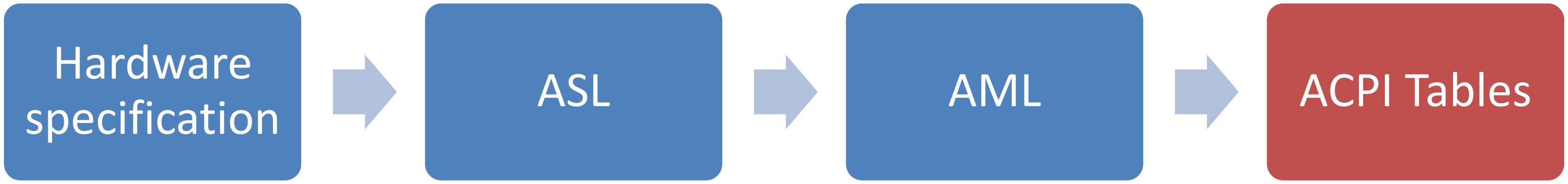
- Firmware interface used by OS
  - Enables device discovery and configuration
  - Enables OS power management
- Specifies firmware data tables as well as executable bytecode called AML.

ACPI firmware interaction with OS



ASL & ACPI firmware development

- AML has a human-readable form called ACPI source language (ASL)



Hardware specification

ASL

AML

ACPI Tables

Sample ASL

```
DefinitionBlock ("", "SSDT", 2, "", "", 0x01)
{
 Device (DEV1){ // Named object DEV1
  Name (ADDR, 0x1234) // Named object ADDR

  Method (MTH1,1){ // Named object MTH1
   Local0 = 0xABCD + Arg0;
   INT1 = 2 + ADDR + Local0;
   return (INT1)
  }

 }
}
```

AML interpreter runtime issues

- ACPI firmware contains platform-specific information encoded in AML bytecode
  - AML is executed by an interpreter
- Like many languages that run on interpreters, there may be runtime errors.
- These runtime errors may be **serious!**

Common Runtime Errors

- Type errors

  ```
  Local0 = 0x54 + DEV1
  ```

  Note: DEV1 is a reference to a device

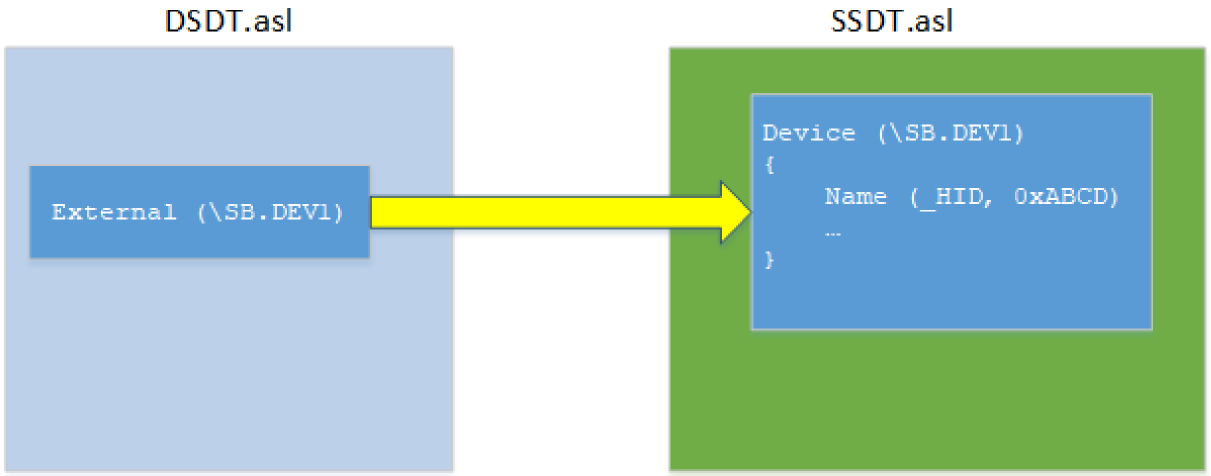
- Out-of-bounds errors

  ```
  Name (BUF1, Buffer(0x3){}) // 3-byte array
  Local0 = BUF1[99] // out of bounds!
  ```

- **Namespace resolution errors**
  - Objects referenced in ASL/AML are undefined
  - Objects are re-defined

Namespace Resolution

- ASL contains a language construct called External (similar to C)
  - Tells the ASL compiler that certain objects are declared in separate tables.



DSDT.asl

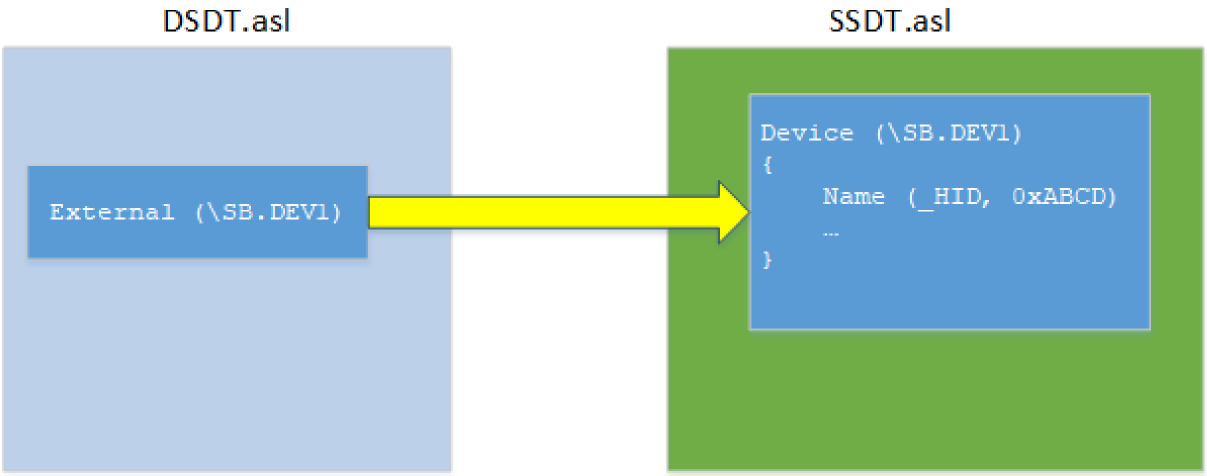
```
External (\SB.DEV1)
```

SSDT.asl

```
Device (\SB.DEV1)
{
    Name (_HID, 0xABCD)
    …
}
```

Namespace Resolution

- ASL files are compiled one file at a time
  - There is no guarantee that these symbols are resolved after compilation of a single file.

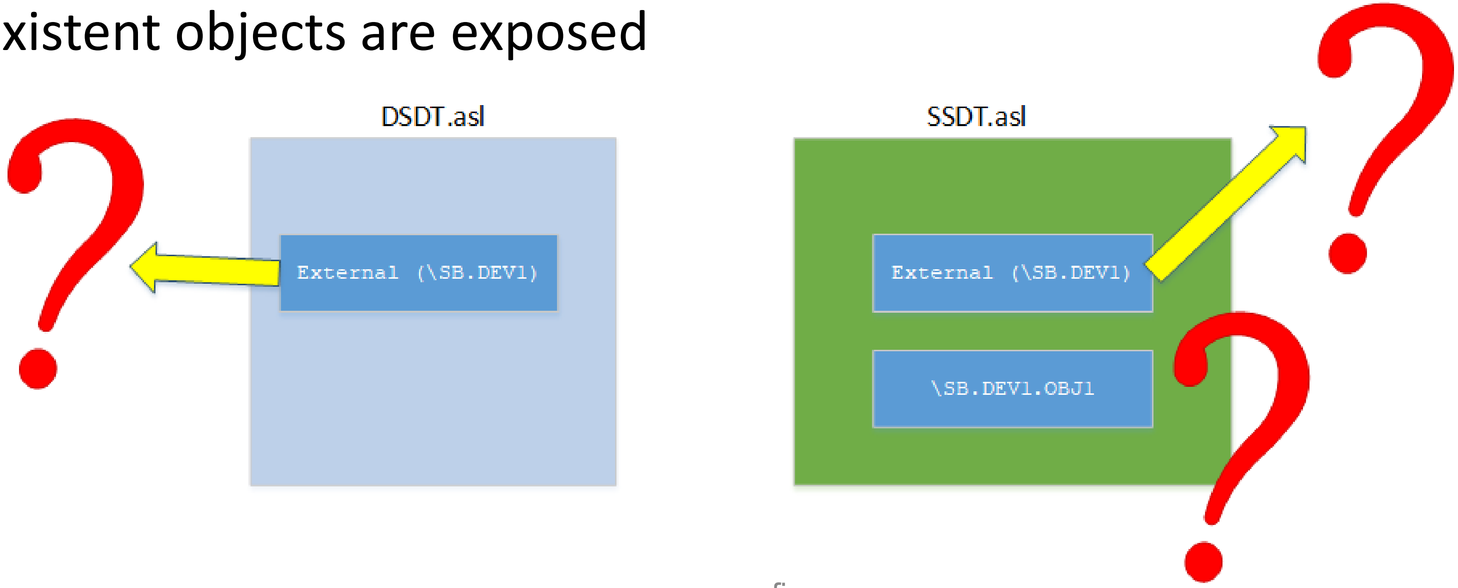


```
External (\SB.DEV1)
```

```
Device (\SB.DEV1)
{
    Name (_HID, 0xABCD)
    …
}
```

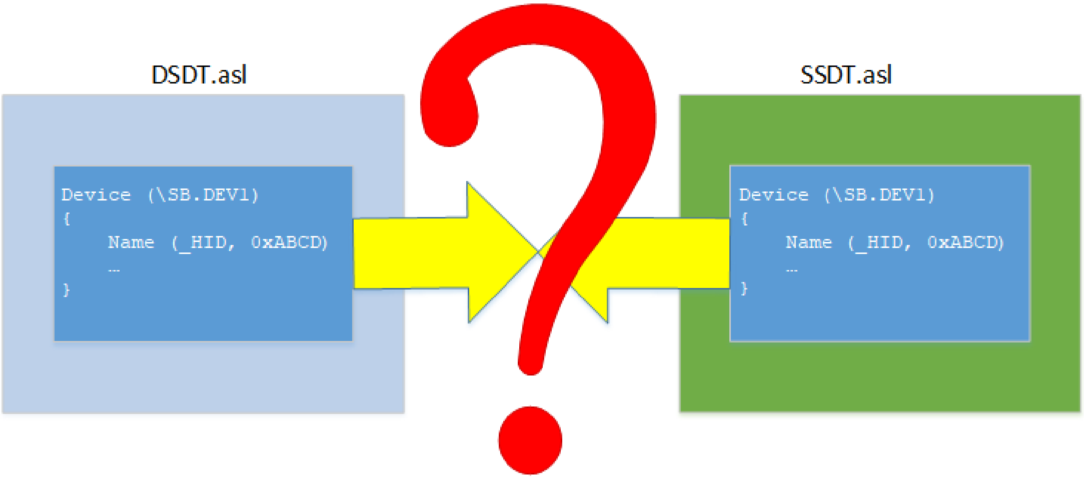
Namespace Resolution

- At OS boot time
  - all ACPI tables are loaded
  - namespace resolution errors appear as references to non-existent objects are exposed



Namespace Resolution

- At OS boot time
  - redundant named objects declarations are also discovered.



```
Device (\SB.DEV1)
{
    Name (_HID, 0xABCD)
    …
}
```

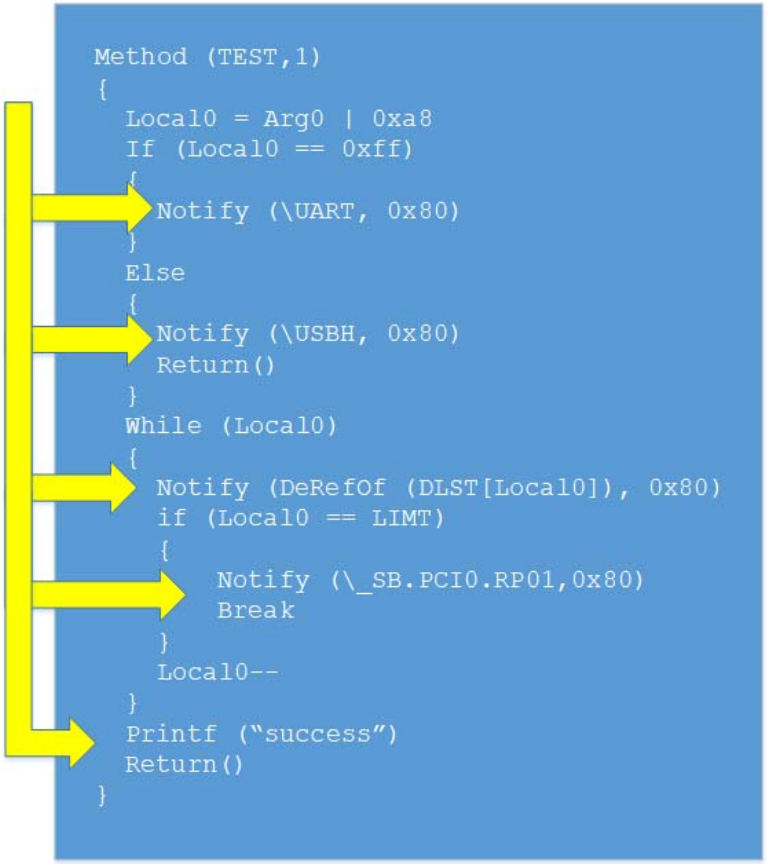
```
Device (\SB.DEV1)
{
    Name (_HID, 0xABCD)
    …
}
```

Solution #1: Use a linker

- These errors should be caught during compilation or linking as a part of development
- Not all ASL compiler emit AML external Opcode for named objects that are declared external

Solution #2: Use an interpreter

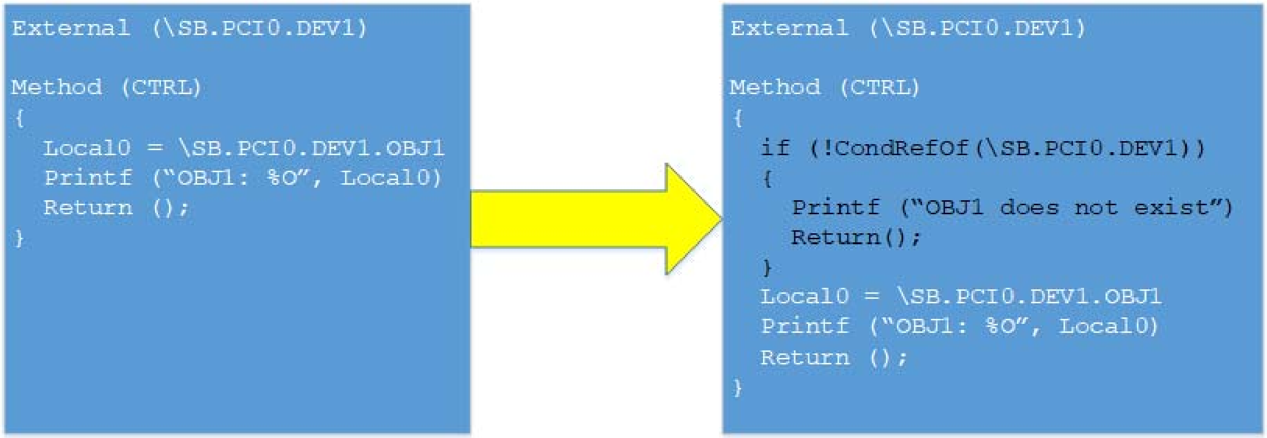
- Use a userspace AML interpreter (ACPIExec) to resolve namespace objects by evaluating named objects
  - Con: it must “uncover” runtime errors by executing all possible code paths in a given ACPI method



```
Method (TEST,1)
{
  Local0 = Arg0 | 0xa8
  If (Local0 == 0xff)
  {
   Notify (\UART, 0x80)
  }
  Else
  {
   Notify (\USBH, 0x80)
   Return()
  }
  While (Local0)
  {
   Notify (DeRefOf (DLST[Local0]), 0x80)
   if (Local0 == LIMT)
   {
     Notify (\_SB.PCI0.RP01,0x80)
     Break
   }
   Local0--
  }
  Printf (“success”)
  Return()
}
```

Solution #3: Add guards in ASL

- Adding If (CondRefOf (…)) to names declared external to avoid referencing undeclared objects
  - Con: may add additional complexity to ASL code



```
External (\SB.PCI0.DEV1)

Method (CTRL)
{
  Local0 = \SB.PCI0.DEV1.OBJ1
  Printf (“OBJ1: %O”, Local0)
  Return ();
}
```

```
External (\SB.PCI0.DEV1)

Method (CTRL)
{
  if (!CondRefOf(\SB.PCI0.DEV1))
  {
    Printf (“OBJ1 does not exist”)
    Return();
  }
  Local0 = \SB.PCI0.DEV1.OBJ1
  Printf (“OBJ1: %O”, Local0)
  Return ();
}
```

Enhanced Namespace Resolution

- Just about everyone who writes ASL uses iASL compiler as a part of their build.
  - Modifying iASL is a solution that does not require an overhaul of firmware build system.
- A new iASL feature allows compilation of multiple ASL tables **in the same namespace**

Enhanced Namespace Resolution

- Previously, the following command compiled each file in separate namespaces

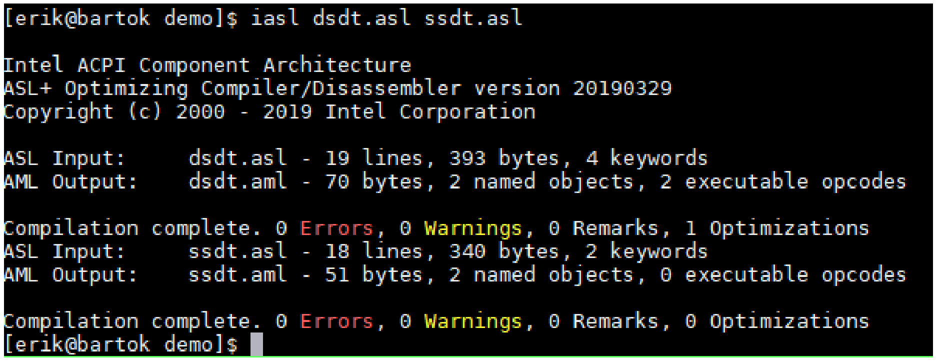
```
iasl dsdt.asl ssdt.asl
```

- Now, the above command assumes that all files compiled together are meant to be packaged together as a set of ACPI tables
  - All named objects in use must be defined
  - No duplicate named object definitions are not allowed

Enhanced Namespace Resolution

```
DefinitionBlock ("", "DSDT", 2, "",
"", 0x01)
{
    External (INT1, IntObj)
    Method (DS01)
    {
       return  (INT1 + 1)
    }
    Name (OBJ1, 0x1234)
}
```

```
DefinitionBlock ("", "SSDT", 2, "",
"", 0x01)
{
    Method (SS01)
    {
    }

    Name (OBJ1, 0x1234)
}
```

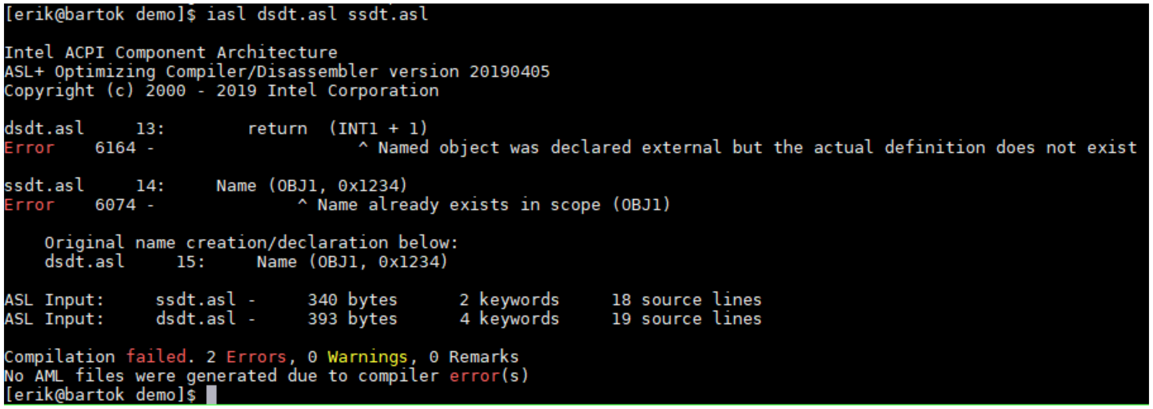


```
[erik@bartok demo]$ iasl dsdt.asl ssdt.asl

Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190329
Copyright (c) 2000 - 2019 Intel Corporation

ASL Input:     dsdt.asl - 19 lines, 393 bytes, 4 keywords
AML Output:    dsdt.aml - 70 bytes, 2 named objects, 2 executable opcodes

Compilation complete. 0 Errors, 0 Warnings, 0 Remarks, 1 Optimizations
ASL Input:     ssdt.asl - 18 lines, 340 bytes, 2 keywords
AML Output:    ssdt.aml - 51 bytes, 2 named objects, 0 executable opcodes

Compilation complete. 0 Errors, 0 Warnings, 0 Remarks, 0 Optimizations
[erik@bartok demo]$
```

Enhanced Namespace Resolution

```
DefinitionBlock ("", "DSDT", 2, "",
"", 0x01)
{
    External (INT1, IntObj)
    Method (DS01)
    {
       return  (INT1 + 1)
    }
    Name (OBJ1, 0x1234)
}
```

```
DefinitionBlock ("", "SSDT", 2, "",
"", 0x01)
{
    Method (SS01)
    {
    }

    Name (OBJ1, 0x1234)
}
```



```
[erik@bartok demo]$ iasl dsdt.asl ssdt.asl

Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190405
Copyright (c) 2000 - 2019 Intel Corporation

dsdt.asl     13:         return  (INT1 + 1)
Error    6164 -                     ^ Named object was declared external but the actual definition does not exist

ssdt.asl     14:      Name (OBJ1, 0x1234)
Error    6074 -               ^ Name already exists in scope (OBJ1)

    Original name creation/declaration below:
    dsdt.asl     15:      Name (OBJ1, 0x1234)

ASL Input:     ssdt.asl -     340 bytes      2 keywords     18 source lines
ASL Input:     dsdt.asl -     393 bytes      4 keywords     19 source lines

Compilation failed. 2 Errors, 0 Warnings, 0 Remarks
No AML files were generated due to compiler error(s)
[erik@bartok demo]$ 
```

Enhanced Namespace Resolution

- Enables iASL to determine unresolved symbols as well as duplicate symbols during compilation
- Eliminates two of the most serious runtime errors found in modern ACPI firmware.

Takeaways

- We need to create more features in iASL or ASL to detect more runtime errors during compilation
- May require firmware developers to improve existing ACPI firmware
- We need feedback on what can make ASL programming easier

Questions?

Links

- ACPICA project website: https://acpica.org/
- Latest ACPICA release: https://acpica.org/downloads
- ACPICA mailing list: https://lists.acpica.org/mailman/listinfo
- ACPI 6.3 specification: https://uefi.org/sites/default/files/resources/ACPI_6_3_final_Jan30.pdf
- ACPI Specification Working Group: https://uefi.org/workinggroups

Thanks for attending the 2019 Spring UEFI Plugfest

For more information on UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by