Win64 (Microsoft)

- Port to NT5 (Same APIs as Win64)
- Use clean code check list on Intel web site to IA-64 clean code your applications
- Visit the MS developer site for detailed information on preparing your app for Win64 and to download tools as they become available

For more detail see:
- Designing 64-bit-Compatible Interfaces
  http://msdn.microsoft.com/developer/news/feature/win64/port64.htm
- Getting Ready for 64-bit Windows
Monterey (IBM/SCO)

- Port to UnixWare 7 (use UNIX 98 APIs)
- Use 64-bit version of Lint available from SCO, for code clean-up
- Use clean code check list on Intel web site to IA-64 clean code your application
- Use PreSi SDK and SoftSDV for preliminary testing (available Summer 99).

Visit SCO and IBM sites for more updates on Linux for IA-64 development tools:
- 64-bit UnixWare Porting Guide
  http://www.sco.com/developer/64bit.htm
  http://www.ibm.com/servers/monterey/
Linux

- Port to IA-32 Linux
- Perform 64 bit code cleanup using the 64-bit Lint code clean-up tool that is available now

Join the VA Research’s news group for up-dates on Linux for IA-64 development tools:
- http://www.linuxia64.com
HP-UX

• Port to HP-UX ver 11.0: Use UNIX 98 APIs
• Use clean code check list on Intel web site to IA-64 clean code your application
• Use PreSi SDK and SoftSDV for preliminary testing (available 2H’99).
• Download the Software Transition Kit (STK) on HP’s web site, which provides useful documents and tools to help you identify and resolve any IA-64 transition impacts in your C, C++, Fortran, or COBOL software and scripts (http:www.software.hp.com/STK)
Solaris (Sun)

- Port to Solaris 7 Intel Edition: Use UNIX 98 APIs
- Use clean code check list on Intel web site to IA-64 clean code your application
- Use Lint in 32 bit & 64 bit environment clean code your application

Visit Sun’s developer site for technical documents and tools:
- [www.sun.com/developer/solaris64](http://www.sun.com/developer/solaris64)
Modesto (Novell)

- Transition to new APIs on NetWare 5
- Use clean code check list on Intel web site to IA-64 clean code your application
Tru64 Unix (Compaq)

- Port to Tru64 Unix (use UNIX 98 APIs)
- Use clean code check list on Intel web site to IA-64 clean code your application
- Visit the Tru64 site for more up-dates (http://www.unix.digital.com/)
## Clean App For IA-64 Architecture - Porting Checklist

(Syntax, Compiler issues)

<table>
<thead>
<tr>
<th>Issue</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid casts between pointers and ints (and/or longs)</td>
<td></td>
</tr>
<tr>
<td>Invalid printf specifiers</td>
<td></td>
</tr>
<tr>
<td>Usage of existing APIs with pointer(and/or long) as parameters, return values</td>
<td></td>
</tr>
<tr>
<td>Usage of undocumented/reserved bit fields</td>
<td></td>
</tr>
<tr>
<td>Remove hardcoded values for size of data types</td>
<td></td>
</tr>
<tr>
<td>Remove hardcoded values for Bit shift values</td>
<td></td>
</tr>
<tr>
<td>Remove hardcoded constants in Memory allocation functions</td>
<td></td>
</tr>
<tr>
<td>Fix unguarded &quot;ifdefs&quot; from defaulting to unwanted code generation</td>
<td></td>
</tr>
<tr>
<td>Fix accessing data structure members via constant offsets</td>
<td></td>
</tr>
<tr>
<td>Algorithms that make use of special bits in pointer arithmetic assuming fixed size</td>
<td></td>
</tr>
<tr>
<td>In-line assembly code</td>
<td></td>
</tr>
<tr>
<td>Self modifying code</td>
<td></td>
</tr>
<tr>
<td>Appropriately modify portions of code storing/restoring on-disk structures</td>
<td></td>
</tr>
<tr>
<td>Portions of code utilizing data-packing</td>
<td></td>
</tr>
</tbody>
</table>