**TERMS**

**Analysis**
Identify Year 2000 problems in a specific component and determine the impact on the system. Used interchangeably with ‘Assessment’.

**Apparently Compliant**
Vendor assertion stating the Year 2000 compliancy of their product or service; however, the individual department has not verified the vendor statement by testing.

**Application**
One or more related programs that are processed together to perform a task or series of tasks such as FFE or AP/CAR. Sometimes used interchangeably with ‘system’.

**Assessment**
Identify Year 2000 problems in a specific component and determine the impact on the system. Used interchangeably with ‘Analysis’.

**Common Component**
A component that is used by many departments and whose Year 2000 compliancy has been researched by the Year 2000 Project Office. Common components include general use software, hardware, networks, etc. as well as applications (software) that was written and/or maintained by AIS or.

**Component**
A tool or item that supports a function. Examples include software, computer hardware, networks, devices with embedded devices, data exchanged with other departments, schools or outside institutions, providers of technology dependent services, suppliers of critical materials, etc.

**Continuity Plan Testing**
Plans must be tested to validate execution. See Year 2000 Continuity Plan

**COTS**
Commercial Off The Shelf. An item that is purchased from an outside vendor and that is not customized, or is only minimally customized.

**Criticality**
Extremely important to the continued viability of a function.

**Embedded Device**
Computer programs, instructions, or functions implemented and integrated as part of a hardware system or microchips. Includes programs or instructions that are stored permanently in programmable, read-only memory (PROM), and constitute a fundamental part of system hardware.

**Firmware**
Computer programs, instructions, or functions implemented in hardware. Such programs or instructions —stored permanently in programmable, read-only memory—constitute a fundamental part of system hardware.

Glossary.doc 1/13/1999
### Function
An activity (level educational, research, administrative, etc.) performed by a department or school to support its mission or reason for existence. Examples might include such as student support, instructional support, research projects and departmental accounting; see list of “Typical Functions in Schools & Departments” in the *Year 2000 Guidance Manual*.

### In-house/Custom Application
A program or series of related programs written to provide one or more functions unique to a department or school. May have been written by an employee or a consultant.

### Remediation
Work or procedures performed to fix a year 2000 date problem. May involve modifications to computer programs; replacement or upgrades to software packages, hardware and device with embedded microprocessors; or phase-out (retire) systems no longer used.

### Risk
The combination of business risk (or risk to the University) and date risk that defines the criticality.

### High Local Risk
A risk that could impact most or all of an entire school or department. (e.g., failure of a device that is essential to the completion of several research projects; failure of computers, computer applications, computers or devices required to complete a significant research project; failure of a LAN or other component that is widely used throughout a school or department; etc.)

### Low Local Risk
All other less significant local risks

### University Risk
A risk that could impact large sections or all of the university. By definition, anything that could adversely affect the health and/or safety of humans or animals is automatically defined as University Risk. Similarly, if a research project or research data is shared between departments or schools or an outside institution, the risk level is University. Other examples include: inability to conduct teaching on a university-wide basis due to the loss of a basic service such as heat or electricity; inability to perform high cost and/or high profile research, inability to perform essential administrative or accounting functions, etc.

### Test/Testing
Validation procedures to ensure that a program or piece of equipment stills functions properly – changes made to correct a Year 2000 date problem did not produce unexpected results in the rest of the program or system. We generally use ‘test’ to indicate procedures performed after remediation. Sometimes actual testing must be performed as part of the analysis/assessment process, i.e., Testing to Prove Compliance.

### Unit Date Test
Validate date functionality for a single program module

### System Date Testing
Validate date functionality across the entire system
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Testing to Prove Compliance</strong></td>
<td>Perform date testing to validate vendor assertions</td>
</tr>
<tr>
<td><strong>Year 2000 Compliant</strong></td>
<td>A component has been tested and determined that it will not experience failures or produce incorrect calculations or results. A functional definition: Year 2000–compliant systems; Accept, process, calculate, sort, store, display, and report dates using the correct century and year.</td>
</tr>
<tr>
<td><strong>Year 2000 Continuity Plan</strong></td>
<td>A plan or set of tasks and/or procedures to be used to recover from a failed function. May consist of manual work around procedures or alternatives that are used to sustain a function for a period of days or weeks, until the failed component(s) are repaired and reinstalled.</td>
</tr>
<tr>
<td><strong>Year 2000 Failure</strong></td>
<td>The failure of a component because it cannot properly process dates after December 31, 1999.</td>
</tr>
</tbody>
</table>