NEI/NUSMG 97-07

NUCLEAR UTILITY YEAR 2000 READINESS

OCTOBER 1997

NEI/NUSMG 97-07

NUCLEAR ENERGY INSTITUTE
NUCLEAR UTILITIES SOFTWARE MANAGEMENT GROUP

NUCLEAR UTILITY
YEAR 2000 READINESS

OCTOBER 1997

ACKNOWLEDGMENTS

This document, *Nuclear Utility Year 2000 Readiness*, NEI/NUSMG 97-07 was developed with the assistance of an issue task force of industry managers dealing with current software issues. Timely development of this document was facilitated by use of the combined resources and expertise of the Nuclear Energy Institute and Nuclear Utilities Software Management Group. NEI wishes to acknowledge the extensive review and comments by the industry representatives who shaped the final form of this document.

NOTICE

TABLE OF CONTENTS

APPENDIXES

NUCLEAR UTILITY YEAR 2000 READINESS

1. INTRODUCTION

Nuclear utilities, like many other industries and government agencies, cannot satisfy their operating commitments without software. As nuclear utilities approach the turn of the century, they face a significant and complex task in resolving the Year 2000 (Y2K) problem in their software.

The problem occurs in some software because two-digit date fields were used to represent the year. In some software the logic fails when the "00" of year 2000 is inserted in the two-digit field. Others do not correctly identify the year 2000 as a leap year. The Y2K problem can affect software in mainframes, desktop computers, local area networks (LAN) or digital control systems.

No utility can escape the deadline and none are immune to the costs and responsibilities associated with this problem. Defining the exact severity and extent of Year 2000 problems is complicated by many factors:

- A large and diverse software inventory (typically 300 applications per nuclear unit),
- Numerous embedded systems that are difficult to inventory and test,
- The potential for operability issues or unreviewed safety questions (USQ) in safety system software,
- Costs - $1 to $3 million estimated at many units,
- The need to obtain information from vendors, and
- Limited time to identify and correct the problem, and significant staff requirements.

Nuclear facility licensees must ensure facilities are operated safely and in compliance with all license provisions. The Nuclear Regulatory Commission expressed their concern over this issue in Information Notice 96-70, "Year 2000 Effect on Computer Systems Software."

This document is the result of actions taken by the Nuclear Utility Software Management Group (NUSMG) in conjunction with the Nuclear Energy Institute (NEI) to provide the nuclear industry with an approach to resolve the Y2K problem.

This document suggests a strategy for a nuclear utility Year 2000 Project. This strategy recognizes management, implementation, quality assurance, and documentation as the fundamental elements of a successful Project. The NEI/NUSMG Task Force recognizes that any solution to the Year 2000 problem is an iterative process and many steps overlap as methods improve the testing and management will evolve through the feedback process.

2. PURPOSE AND SCOPE

2.1 Purpose

The purpose of "Nuclear Utility Year 2000 Readiness" is to recommend methods for nuclear utilities to attain Y2K readiness to ensure that their facilities remain safe and continue to operate within the requirements of their license. These methods and suggestions are designed to expedite completion and control costs.

2.2 Scope

"Nuclear Utility Year 2000 Readiness" applies to software, or software based system or interface, whose failure due to the Y2K problem would prevent the performance of the safety function of a structure, system, or component. This document also applies to any software, or software based system or interface, whose failure due to the Y2K problem would degrade, impair, or prevent operability of the nuclear facility. It is intended to supplement and use existing procedures used for software quality control, configuration management and problem reporting.

3. DEFINITIONS

3.1 **Year 2000 (Y2K)** — A term used to describe a set of date-related problems that may be experienced by a software system or application. These problems include: not representing the year properly, recognizing that the year 2000 is a leap year, and improper date calculations.

3.2 **Y2K Compliant** — Computer systems or applications that accurately process date/time data (including but not limited to, calculating, comparing, and sequencing) from, into and between the twentieth and twenty-first centuries, the years 1999 and 2000, and leap-year calculations.

3.3 **Y2K Ready** — A computer system or application that has been determined to be suitable for continued use into the year 2000 even though the computer system or application is not fully Y2K Compliant.

3.4 **Validation** — A process that evaluates the functional characteristics of the software, and certifies the achievement of acceptable comparisons with Objective Evidence.

3.5 **Objective Evidence** — Any statement of fact, information, or record, either quantitative or qualitative, pertaining to the quality of an item or service based on observations, measurements, or tests that can be verified.

3.6 **Remediation** — Remediation is the process of retiring, replacing, or modifying software or devices that are to be retained in service, but have been determined to be affected by the Y2K problem.

4. MANAGEMENT PLANNING

The management plan suggests an approach to establish, organize, manage, integrate, and complete a nuclear utility's Y2K project. The recommended components for the management plan are shown in the following systems.

4.1 Management Awareness

The scope and nature of the problems that may occur in software systems at the turn of the century are not generally appreciated or understood at many levels of utility management. Correcting this condition is essential for a Y2K project to obtain the necessary levels of support, cooperation, and funding.

Communicating an awareness of the Y2K issues ensures that senior management, and their management team, understand the vulnerability of their utility. Senior management's attention to this problem indicates their commitment to maintaining the margin of safety and the operability of their facilities.

4.2 Sponsorship

The Y2K project requires significant commitments of personnel, facilities, and funds. The project also requires support between, and by, many organizations within the utility. Available estimates indicate that even a single-unit utility project requires resource variances that are typically authorized by senior management. Senior management sponsorship is recommended.

4.3 Project Leadership

The Y2K project requires a significant commitment of people knowledgeable of the commitments, strategic intent, culture, vulnerabilities, and capabilities of the utility. The project must be sufficiently staffed during the planning stage and continuing through completion. Since many tasks are required and many software systems evaluated, the resources must be allocated and managed effectively. The project requires strong, effective leadership.

The authority given to the project manager is largely a function of the number of units, the extent of the interfaces with suppliers, the complexity of the problems encountered, and the culture of the utility. The reporting level of the project manager should be established to ensure appropriate corrective measures are completed.

The project management may be a matrixed function that includes project managers from major organizations within the utility. However, this does not diminish the ultimate authority and responsibility of the individual that manages the overall project.

4.4 Project Objectives

The project manager should fully understand the project sponsor's expectations concerning the major objectives of the project. This includes allocation of resources and schedule. The project manager should document the understanding of the project and obtain written agreement from the sponsor.

4.5 Project Management Team

The project manager selects a project team that may include other managers, technical specialists and support staff seconded from functional organizations. This team may benefit from the participation of professionals that specialize in project management, cost and scheduling, outage management, and other disciplines.

The project team requires access to the skills of a multi-disciplined cross-section of the organization. This includes the process owners (system engineers, technicians, subject matter experts, etc.), information technology professionals, and support organizations such as engineering, licensing, quality assurance, procurement, and financial management. The use of consultants and contractors may be desired, or even necessary for completion of the project.

4.6 The Management Plan

The project team may use this document to develop the approach that the utility will use to address the Y2K problem. The management plan documents the major milestones of the project and the schedule for completion. The management plan includes a description for addressing each item in Section 4 of this document.

The management plan identifies those responsible for creating the implementation plan (see Section 5), its content, and the procedures and controls to be used to manage the implementation the Y2K project. The management plan should indicate the strategies used to address or establish:

- Ownership of software changes,
- Vendor relationships and responsibilities,
- Communication and feedback from affected parties, and
- Contingency plans for unanticipated events at the point of the turn of the century.

The project sponsor and management of participating organizations approve the management plan and subsequent revisions.

4.7 Project Reports

The project manager documents the progress of the project in status reports to the project sponsor and appropriate members of management. These reports should include details of key performance indicators such as numbers of systems addressed, expenditures, the current disposition of resources in the field, and schedule status.

4.8 Interfaces

The project manager should ensure that interfaces to other organizations (electric utilities, telecommunications utilities, suppliers, emergency services, government offices, etc.) are considered for their importance to the objectives of the project. Interfaces that are identified should be addressed by ensuring that the responsible organization institutes an appropriate Y2K effort.

4.9 **Resources**

Significant resources will be required to complete the Y2K project. The project sponsor is accountable for allocating the resources agreed to in the project plan and meeting additional requests as the project matures.

4.10 **Oversight**

This project requires the project sponsor to remain actively engaged in the oversight of the project through completion. The project sponsor may also engage the services of outside organizations to supplement the oversight of the project.

4.11 **Quality Assurance**

Quality assurance measures are applied throughout the Y2K effort to include both the management and the implementation activities. These measures are structured to ensure that the performance of essential activities is supported by objective evidence. These measures are to ensure:

- Personnel participating in this project are qualified for assigned tasks,
- Activities that could affect safety or operability are accomplished using appropriate procedures, and
- Non-conforming conditions discovered during the conduct of the Y2K project that are determined not to be Y2K issues are identified and dispositioned in accordance with appropriate procedures.

Further details regarding quality assurance measures are presented in Section 6, Quality Assurance.

5. **IMPLEMENTATION PLAN**

Each project team defines the process and methods used to carry out the requirements of the management plan by developing implementation plans. The implementation plans for the project is approved by appropriate levels of management. The suggested phases of implementation include awareness, an initial assessment, a detailed assessment and notification.

5.1 Awareness

The purpose of the initial communications is to raise general awareness of the issue and to communicate its importance to the organization. The communication or indoctrination must be aligned to the audience. The audience includes the following:

- Management,
- Subject matter experts,
- System engineers,
- Software or system sponsors,
- General employees, and
- Support organizations such as procurement and engineering programs.

At a minimum, communications should include a description of the Y2K problem, the process or plans to address and remediate the problem, the significance or priority of the problem, the resources required and the schedule. An example of a communications plan is included as Appendix A.

5.2 Initial Assessment

The initial assessment consists of identifying software in use by the utility. This requires input and support from all participating organizations. Initial assessment consists of several steps.

5.2.1 Inventory

An inventory of all potentially affected items is required. The data collected is used to make initial decisions on categorization, classification, and prioritization. The data is also used to determine budget and resource estimates for the detailed assessment phase. The information collected may include the following:

- Software or device name,
- Version or model number,
- Description and use,
- Priority based upon importance to safety, operability, regulatory commitments, business considerations, etc.,
- Vendor or manufacturer, and
- Owner or support group.

Embedded systems are particularly difficult to inventory. The software components are often not recognized or apparent. Particular

care should be taken to ensure that embedded systems are included in the inventory. When systems are being examined to determine whether embedded components are within, the individuals tasked for this activity should be highly skilled in their design or use. Some suggested indicators that may be used to determine the presence of embedded software include:

- Searching procedures and documentation for the occurrence of phrases that would indicate the existence of an internal clock or processor,
- Surveying vendors for information on their equipment,
- Performing system walk-downs, and
- Reviewing schematics, programming listings, and reference manuals.

The guidance for collection of the inventory should include the types of items to list, the use of existing inventories, and information required for future actions and decisions. This is to ensure that all software within the scope of the project is evaluated for the Y2K problem and documented. Examples of inventory instructions are in Appendix B.

5.2.2 Categorization

After the inventory is collected, a categorization of the inventoried items is performed. Categorization is the process that groups software, allowing management to efficiently assign resources to the classification and prioritization activities. Examples of categories are:

- Mainframe applications,
- System software (operating systems, databases, utilities, etc.),
- Client/server applications,
- Telecommunication equipment,
- Embedded devices,
- Process systems,
- PC's and servers,
- Test equipment, and
- Software interfaces.

5.2.3 Classification

After the inventory is categorized, each item within each category is classified. The process employed should reflect the importance of the item to the objectives of the project. Examples of classifications are:

- Safety-related,
- Important-to-safety,
- Required by regulations,
- Required by license commitments,
- Important to operation,
- Personnel safety,
- Continuity of business, and
- Non-essential.

5.2.4 Prioritization

Prioritization is the process of reviewing all items within the inventory after classification and assigning an order to the performance of the detailed assessment. Criteria used to set the priorities are established by the utility in their management plan. Examples include:

- Number of systems of a given type,
- The availability of individuals with required talents or experience, and
- Competing schedules such as equipment replacement and outages.

5.2.5 Analysis of Initial Assessment

The final step of the initial assessment is to determine the scope, schedule and estimated resources required for the detailed assessment based on the initial prioritization and categorization. This is a critical business consideration that requires significant resources to perform. Analysis of the data may require substantial management and technical resources and will certainly be an iterative process.

Note: Some items may not require detailed assessment and may be dispositioned as used-as-is.

5.3 Detailed Assessment

The purpose of the detailed assessment is to obtain sufficient information about each inventoried item to determine its expected performance beyond

December 31, 1999. Written instructions, checklists or test procedures should be developed to describe the detailed assessment process and provide for documentation and quality assurance of the work performed. Assessment results are used to make decisions regarding actions required to ensure the continued operation of the software. Several activities occur during the detailed assessment phase.

5.3.1 Vendor Evaluation

It is essential to determine whether the software in question is vendor supplied so that responsibility for subsequent activities can be established. For software determined to be vendor supplied, but for which no vendor support is available or forthcoming, the software must be evaluated by the utility using their Y2K processes (see 5.3.2).

For vendor supplied software that the vendor supports, the utility needs to determine the appropriate commercial instrument (contract, license agreement, interface plan, etc.) to use, or institute, for subsequent activities. These activities may include remediation by the vendor, cooperative efforts with the vendor, or the issuance of a request for Y2K information and certification.

The development of a generic Y2K compliance specification for communicating the definition of compliance to vendors, the type of information requested, and the desired extent of documentation is beneficial. The vendor compliance specification may also be used for current purchases to ensure that only Y2K compliant software is purchased. Refer to Appendix C for an example of a compliance specification and Appendix D for an example of a vendor readiness questionnaire.

For vendor responses that indicate an application or device is Y2K ready or compliant, a decision on whether or not to perform validation testing is required. This decision may be based on the criticality of the item, prior experience with the vendor, the extent of documentation provided, or utility knowledge of the item.

5.3.2 Utility Owned or Supported Software Evaluation

An assessment of the utility owned and supported applications and devices is performed using procedures or checklists. Appendix E contains examples of detailed assessment procedures and checklists. There are many methods for determining the Y2K operability of applications and devices including knowledge-based decisions,

scanning (used for mainframe and some large client server applications) and testing.

Testing (see 5.5) may be used for Y2K assessments and requires the development of test specifications or procedures. Testing results often reveal the best strategy for remediation. Appendix F contains examples of test specifications.

5.3.3 Interface Evaluation

It is essential to coordinate interfaces between the software applications modified by the Y2K project and those maintained by other internal or external organizations. For example, the utility should ensure that all telecommunication equipment required under the scope of this project is Y2K compliant or ready.

The coordination and timing of such efforts presents many challenges and may require a high level of project management attention. Interfaces with external organizations should be identified early in the process and require regular management attention.

5.3.4 Remediation Planning

After an application or device has been determined to be susceptible to the Y2K problem, a business decision must be made. At issue is whether the software can be used as-is, or whether it must be retired, replaced or modified (RRM). This evaluation must be documented and should include the options evaluated, their cost, schedule, benefits, and risks. The results of the RRM decisions provide the input to the scope, schedule, and cost estimates for the remediation phase.

5.4 Remediation

The purpose of remediation is to retire, replace or modify software identified in the detailed assessment. The remediation phase requires the project to develop a process for tracking progress and evaluating the risks for items remediated. The process should track replacement projects, purchases, conversions, deletions, retirements, and vendor efforts (see Appendix G for examples of a readiness tracking process).

During remediation the utility should ensure proper software quality assurance controls and procedures are utilized. For unit equipment remediation, the work will need to employ existing station modification procedures.

5.5 Y2K-Testing And Validation

The purpose of Y2K-testing in support of evaluation efforts is to determine whether the Y2K problem is present. This testing is performed during detailed assessments.

The purpose of Y2K-testing subsequent to remediation is to determine whether those efforts have eliminated the Y2K problem and no unintended functions are introduced.

Y2K-testing may be performed at several levels:

- Unit testing focuses on functional and compliance testing of a single application or software module,
- Integration testing tests the integration of related software modules and applications, and
- System testing tests the hardware and software components of a system.

The purpose of validation is to determine that the software is capable of performing its intended function. Validation is performed subsequent to remediation and Y2K-testing.

Upon satisfactory validation, the project manager obtains from those performing the validation certification and documentation consistent with the requirements of the project. The certification should clearly indicate Y2K ready or compliant.

5.6 Notification

Affected parties, including users, and vendors, shall be notified of changes to the software or hardware. This includes changes to documentation that may also result from this project.

6. QUALITY ASSURANCE

Quality assurance measures are applied to processes and systems to provide a level of assurance that they will adequately perform their intended function. In the context of Y2K, processes refer to those activities that are managed by the project manager and performed to ensure the accomplishment of project objectives. Systems refer to software, digital

processors, and associated files, documentation, and equipment pertinent to the Y2K Project.

The project manager should consider the quality assurance programs that exist within the utility and determine applicability to the Y2K project. This includes the nuclear programs, business programs used for non-nuclear applications, and commercial programs that apply to products that are supplied to others. The quality assurance measures may be graded in their application so the extent of the quality assurance activities is consistent with the importance of the item or process to safety and operability.

A nuclear quality program governs some systems addressed under this project. They are subject to the provisions in CFR 50 Appendix B, certain regulatory guides, and commitments in the licensee's Safety Analysis Report. The project manager ensures that the nuclear quality assurance program adequately implements applicable requirements to software systems.

6.1 Project Management Quality Assurance

Quality assurance measures applied to the Y2K project should be performed in accordance with approved procedures. The measures should ensure that an appropriate level of oversight of the Y2K project is performed. This oversight may take the form of planned periodic audits, inspections at documented hold points, or reviews of approved documents. Oversight should be provided by individuals or groups not directly involved in the management of performance of Y2K project activities.

6.2 Implementation Quality Assurance

Quality assurance measures should be applied to the implementation phase of the Y2K effort. In addition to those measures identified in Section 5, Implementation Plan, additional measures should be applied as follows:

The project manager should ensure:

- The system is classified and categorized according to nuclear safety,
- Pertinent system procurement information is obtained,
- Systems are placed, or retained, under a system of configuration management, and
- All systems completed are validated and their design and licensing basis are documented using approved procedures.

The measures should ensure that required remediation changes to the software, hardware, and affected documents are made and that affected groups and individuals are notified of the change.

7. REGULATORY CONSIDERATIONS

Appropriate reviews and/or evaluations are performed and documented. Those that apply are dependent upon the classification of the system. Examples include, but are not limited to:

- 10 CFR 50.59:
  - Safety Screenings
  - Safety Evaluations
  - Determination of Unreviewed Safety Questions
- NRC Safety Evaluation Reports
- Reportability Evaluations per:
  - 10 CFR 50.72
  - 10 CFR 50.73
  - 10 CFR 21
- Operability Determinations
- Reviews to determine the need for changes to:
  - Safety Analysis Reports
  - Technical Specifications
  - Technical Requirements Documents
  - Design Basis Documentation Procedures
  - Licensing commitments
- Radiological/non-radiological reviews
- Emergency data response system reviews
- Purchasing review
- Legal department reviews

8. DOCUMENTATION

Documentation of Y2K program activities and results serves several important purposes:

- Provide management's expectations and guidance on the conduct of the project,
- Collect the data needed to monitor and manage the progress of the project,
- Allow independent parties to review of the project during and after completion,
- Record the basis of the ready or compliance certification,

- Record the justification for leaving an application "as-is" but neither compliant or ready, and
- Record utility management and technical decisions in the event of litigation.

This section provides basic requirements and examples for organizing data collection and developing records for the project. Utilities should use existing Software Quality Assurance (SQA) and configuration management procedures as primary records of change control.

8.1 **Documentation Requirements**

Utilities should prepare documents that demonstrate the completeness of their Y2K program efforts and record the disposition of each item in their inventory. Records should be formatted to support information retrieval. They should support the audit and oversight activities of the project.

8.2 **Project Management Documentation**

Project management procedures and the documents generated through their use should be retained. They document the utility efforts to resolve the Y2K problem and the results of the many activities performed. The procedures and documents will also serve as legal records of the utility efforts to resolve a problem that has generally recognized legal liabilities.

Examples of records used to document management of the Y2K project are:

- Program procedures or plans used to define the requirements of the project,
- Inventory lists,
- Project tracking data,
- All records signed by management, and
- Status reports and financial reports.

8.3 **Vendor Documentation**

Since most utilities use vendor software extensively, the management of vendor documentation poses a significant task. The records resulting from this task will be challenging to manage, understand, integrate with internal efforts, and disposition. The project should consider dedicating specific management resources to this topic. Vendor documentation includes:

- Letters to vendors,
- Vendor responses,

- Utility disposition or additional testing of the item, and
- Other correspondence files.

8.4 Inventory Lists

An accurate inventory list is an essential document and forms the basis for generating other records. It also stands as the record of a complete and thorough assessment process. Inventory records that should be developed were identified in the implementation plan section.

8.5 Checklists for Initial and Detailed Assessments

Checklists should be used for each application indicating progress through each step in the Y2K project. The checklist should be reviewed and completed by both the business subject matter experts and the technical team members who are responsible to support the item. The checklist should lead the responsible persons through the entire process in a manner that helps them properly evaluate the items and record their responses and comments to specific questions. An example of a Y2K checklist is provided in Appendix H, Certificates of Completion.

A Certificate of Completion should be prepared and signed by appropriate personnel for each application to indicate its final disposition. It represents management's approval both from a technical and business perspective. management's approval also indicates acceptance of risk when an application is not certified as compliant or ready.

It is prudent to have both the technical owner and the business owner document their concurrence with the final resolution and disposition of the application.

8.6 Record Retention

Project records should be maintained in accordance with the utility's Software Quality Assurance (SQA) procedures, configuration management programs, recommendations of the legal department, and the project plans.

NUCLEAR UTILITY YEAR 2000 READINESS

Appendix A

COMMUNICATIONS PLAN

Introduction

This document describes the communications plan for the Year 2000 Program. This plan addresses the range of communications needs required to raise awareness and inform COMPANY employees about the Year 200 Program.

The strategy ensures that stakeholders are kept informed about the program's goal, objectives, risks and progress according to plan.

It is important that all participants in the program are provided with materials to prepare them for their roles and responsibilities.

Audience

The target audience for this document is COMPANY employees who use information technology and work in conjunction with the Year 200 Program Office.

This plan also can be used for external communications with third-party vendors, government regulatory agencies and the media.

Communications Planning Requirements

Information in the matrix below outlines the Year 2000 communications needs.

- The target audiences -- internal and external.
- The objectives in communicating to the target audiences.
- The communication vehicles to use in communicating to the target audiences.
- The recommended messages for each target audience.

INTERNAL Target Audience	Communications Objectives	Communications Vehicles	Recommended Messages
General COMPANY population. Includes all INTERNAL target audiences.	General awareness including: · Provide project background · Explain benefits to COMPANY · Describe risks to COMPANY · Provide overview of how the program is proceeding including a timeline and status.	· COMPANY Week · Technology Connection · Emphasis · Update Video · SCN Broadcast · Y2k Hotline · Y2k E-mail box · Y2k Web Page	COMPANY management understands the severity of the problem and has a team in place who is working to solve it in a timely and cost effective manner. Y2k impacts all employees who use information technology.
Client Contacts	Clarity about project progress including: · Status · Timeline · Impact · Explain work required. · Identify who will do that. · Identify how work will be accomplished.	· Inventory Spreadsheets · Detailed project schedules And work plans · Compliance sign off sheets · Presentations · Y2k Hotline · Y2k Web Page · Periodic briefings for department meetings · Steering Committee Briefings/Reports	Business Units are the owners of the technology and need to participate in the process. Client contacts are our single point of contact for all communications about the project. Client contacts need to dedicate time to the program on a periodic basis. Client contacts are essential to the success of he project. Client contacts will decide priority, and whether to repair, replace or retire applications. Client contacts will be required to sign off on Y2k compliance.

INTERNAL Target Audience	Communications Objectives	Communications Vehicles	Recommended Messages
C&TS Support Staff	Clarity and project progress and delivery of technical information. · Status · Timeline · Impact to work · Describe when work will be required? · Identify who will do what. · Identify how work will be accomplished.	· Technical Documentation · Inventory Spreadsheets · Detailed project schedules and work plans · Compliance sign off sheets · Y2k Hotline · Y2k E-mail box · Y2k Web Page · TSC Help File · Briefings for department meetings · Presentations	C&TS support staff are essential team members. Their knowledge of the technology is essential to the success of the project. C&TS support staff will need to dedicate time to the project on a periodic basis. Y2k Program Team will work with C&TS to solve technical issues. Y2k "fixes" will impact C&TS work. If "outsourcing" is necessary, C&TS will conform to Y2k standards and schedule. When necessary, C&TS support will be required to sign off on Y2k modifications.
Senior Management including the IT Policy Committee and UPC	· Identify the objectives and Magnitude of the program. · Identify the business Issues. · Explain the business/legal Risks. · Outline strategic decisions that need to be made on an ongoing basis.	· Program Office Documentation · Regular status reports · Summary level schedules and work plan · Program Job Estimate · Issues list · Risk Assessment	Business units are responsible for funding and strategic decisions about the Y2k program. Senior management's commitment and involvement is essential to the success of the program. The program runs a high risk of failure if senior management is not committed. Senior management runs the risk of legal liability if due diligence is not exercised. Low priority and unidentified applications will not be Y2k compliant by 1/1/2000. Significant competitive advantage can be obtained by a successful Y2k program implementation. Senior managers and/or a designee will be required to sign off on Y2k modifications.

INTERNAL Target Audience	Communications Objectives	Communications Vehicles	Recommended Messages
Law	· Describe the legal issues related to Y2k. · Identify the third-part software contractual issues related to Y2k. · Define Y2k compliance for COMPANY.	· Trade press articles · Briefings from law firms with Y2k practices · Legal analysis drafted by COMPANY's law department · Y2k third-party contract Warranty language · Y2k third-party compliance Sign off document	The Corporation and its officers run the risk of legal liability if due diligence is not exercised. Third-party vendors must deliver a Y2k compliant product. Legal action will be taken if third-party vendor products are not made Y2k compliant in a timely fashion.
Third-Party Software Vendors	· Identify the third-party software contractual issues related to Y2K. · Define Y2k compliance for COMPANY.	· Y2k third-party contract warranty language · Y2k third-party compliance sign off document	Third-party vendors must deliver a Y2k product. Legal action will be taken if third-party vendor products are not made Y2k compliant in a timely fashion.
Government Agencies-CPUC, FERC	· Taking a proactive approach, describe how COMPANY is working towards Y2k compliance. · Respond effectively to any required regulations.	· COMPANY Week · Summary level schedules and work plan · Responses to regulatory requests	COMPANY management understands the severity of the problem and has a team in place who is working to solve it in a timely and cost effective manner.
Media	· Describe how COMPANY is working towards Y2k compliance. · Cost to rate payers.	· News Articles · Press Releases	COMPANY management understands the severity of the problem and has a team in place who are working to solve it in a timely and cost effective manner.

Communications Responsibility Matrix

The Communications responsibility matrix outlines who does what in the communications process. The communications process requires the participation of all members of the Year 2000 team.

Communications Process	Program Manager	Communications Manager	Program Office	Steering Committee
1. Establishes Recommended Messages	▪	·	·	·
2. Identifies/Confirms Target Audiences	·	▪	·	·
3. Selects Communications Vehicle(s)	·	▪	·	·
4. Designs Communications Message	·	▪	·	·
5. Develops Communications Message	·	▪	·	·
6. Reviews Communications Message	▪	·	·	·
7. Approves Communications Message	·	·	·	▪
8. Secures Communications Approval(s)	▪	·	·	·
9. Delivers Communications Product	·	▪	·	·
10. Incorporates Lessons Learned into Future Communications Products.	·	▪	·	·

▪ Leads · Contributes

Program Office Staff

Name Position	Phone Extension	ID	Cube Number

Communications Vehicles

The following chart shows the various vehicles used for Year 2000 communications within the company. Contacts are also listed.

Written	Contacts	Phone
E-mails	Program Manager/Communications Mgr.	
E-mail box Y2kemail@AsiCms@CTS	Communications Mgr.	
Fact Sheet (Q&A, Scripts, etc.)	Program Manager/Communications Mgr.	
Help Browser	C&TS	
Intranet/Web Page	Corp. Comm. Info. Tech.	
Mailers - Letters/Memos (internal)	Program Manager/Communications Mgr.	
Media Contact Information	Corp. Comm.	
News Papers (external)	Corp. Comm.	
Technology Connections	C&TS	
COMPANY Week	Corp. Comm.	
Emphasis	Corp. Comm.	
Posters in Lobby	Corp. Comm. .	
Press Release (external)	Corp. Comm.	
Printers (internal)	Corp. Comm.	
Trade Journals (external)	Corp. Comm.	

Verbal	Contact	Phone
Booths at Special Functions	Program Manager/Communications Mgr.	
Employee Year 2000 Hotline	Program Manager/Communications Mgr.	
Manager Presentations	Program Manager/Communications Mgr.	
Radio Spots (external)	Corp. Comm.	

Visual	Contact	Phone
Broadcasts - SCN One-way	Corp. Comm.	
Television Spots (external)	Corp. Comm.	
Update Video	Corp. Comm.	
Video Projects	Corp. Comm.	

The following chart shows the various vehicles used by the Year 2000 Program Office to disseminate Program status information. Contacts are also listed.

Written	Contacts	Phone
Status Reports to Utility Policy Committee - Quarterly Basis	Program Manager	
Status Reports to the Strategic Information Technology Policy Committee	Program Manager	
Monthly Project Status Reports to the PMO	Program Manager	
Help Browser	C&TS	
Intranet/Web Page	Corp. Comm. Info. Tech.	

NUCLEAR UTILITY YEAR 2000 READINESS

Appendix B

INVENTORY INSTRUCTIONS

Y2K - Device Survey Site:
Completed by: Date:

This survey is intended to identify any device/equipment that may have YEAR 2000 implications.

Information needed to complete this form:

1) Device/Equipment # - Device name or equipment number.

2) Device location - where is the device located.

3) Primary System/User Group - who owns/maintains the device.

4) Number Used - Total number of devices on-site.

5) Functional Description - Brief business description of the devices function.

6) Business Criticality:
5 - The date implications impact personnel safety, safety systems, or lost generation.
4 - The date implications impact systems important to safety or regulatory commitments .
3 - The date implications could cause substantial financial impact.
2 - The date implications could cause some financial impact, but work arounds exist.
1 - The date implications can cause minor financial impact, but are not a priority.
Please choose one of the above.

7) External Agent/Vendor - Name of Company/Contact
(External Agent - Information exchanged outside of Company)

8) Vendor Address

9) Y2K Impact: 5- Dates are used to determine calculation outputs.
3- Dates are only used in printed output; no calc impact.
1- No impact
Please choose one of the above.

10) Planned Retirement: Is device planned for replacement if so when will replacement be complete. If replacement is not planned enter NO.

11) Comments

Y2K - Application Survey Site:
Completed by: Date:

This survey is intended to identify any application software that may have YEAR 2000 implications.

Information needed to complete this form:

1) Application - The name the application is commonly called.

2) Functional Description - Brief business description of the application function.

3) Business Sponsor: Who is the primary business sponsor for this application. ex: BEST, group, individual.

4) User(s) Group: What groups use this application.

5) Supported by: LIT(L), Individual(I), Vendor (V)

6) Business Criticality:
5 - The date implications impact personnel safety, safety systems, or lost generation.
4 - The date implications impact systems important to safety or regulatory commitments.
3 - The date implications could cause substantial financial impact.
2 - The date implications could cause some financial impact, but work arounds exist.
1 - The date implications can cause minor financial impact, but are not a priority.

Please choose one of the above.

7) Targeted for Replacement/YR: Is device planned for replacement, if so when will replacement be complete. If replacement is not planned enter NO.

8) Y2K Impact, if known: 5- Dates are used to determine calculation outputs.
3- Dates are only used in printed output; no calc impact.
1- No impact

Please choose one of the above.

9) External Agent/Vendor - Name of Company/Contact
(External Agent - Information exchanged outside of Company)

10) Vendor Address

12) Development Tools: Tools used to develop application. (ex. Tool/Database/Operating System: VB/Sybase, EXCEL, WIN95, DOS, etc)

13) Comments:

NUCLEAR UTILITY YEAR 2000 READINESS

Appendix C

COMPLIANCE SPECIFICATION

YEAR 2000 COMPLIANCE WARRANTY

This Agreement is made this _____day of _____, 1997 by and between

_____("Seller"),

a _____ corporation and _____ ("Buyer")
(State)
a _____ corporation.
(State)

WITNESSETH:

WHEREAS, Seller and Buyer have entered into an AGREEMENT dated_____
for the _____of _____.

WHEREAS, when computational resources (hardware, software and firmware), begin mixing dates from 19xx and 20xx, various and uncertain results can be produced and

WHEREAS, Buyer is aware that various and uncertain results in computational resources can be produced; therefore, it has created the **TECHNICAL CRITERIA FOR YEAR 2000 COMPLIANCE**, a copy of which has been furnished to Seller; and

WHEREAS, Buyer and Seller desire to modify the above referenced AGREEMENT to make it comply with the **TECHNICAL CRITERIA FOR YEAR 2000 COMPLIANCE**.

NOW, THEREFORE, the parties agree to amend the above referenced AGREEMENT as follows:

A. YEAR 2000 COMPLIANCE

1. Seller represents and warrants that the Product sold, licensed, or provided by Seller to Buyer for Buyer's use is and will continue to be "Year 2000 Compliant", as defined in Buyer's **TECHNICAL CRITERIA FOR YEAR 2000 COMPLIANCE**.

B. TESTING

1. Seller warrants that the Product has been tested by Seller and has determined that the Product is Year 2000 Compliant.

2. Seller shall deliver the test plans and results of such test upon written request from Buyer.

3. Seller shall deliver documentation listing for each remediation, the location within the Product and the technique used to remediate, upon written request from Buyer.

4. Seller agrees to participate in additional tests of the Product at no charge to Buyer, to determine Year 2000 Compliance.

5. Seller shall notify Buyer immediately of the results of any tests or any claim or other information that indicates the Product is not Year 2000 Compliant.

C. LIABILITY

Notwithstanding any provision in the above referenced agreement to the contrary, Seller agrees to indemnify and hold Buyer and its shareholders, officers, directors, employees, agents, successors, and assigns harmless from and against any all claims, suits, actions, liabilities, losses, costs, reasonable attorney's fees, expenses, judgments, or damages, whether ordinary, special, or consequential, resulting from any third-party claim made or suit brought against Buyer or such persons, to the extent such claim or suit results from Seller's breach of the warranties contained herein.

D. OBLIGATION

1. To the extent that it is determined by Buyer in its reasonable discretion that the Product is not Year 2000 Compliant, Seller agrees to immediately formulate and implement a written plan of action within ninety (90) days to modify the Product to make it Year 2000 Compliant.

2. A copy of such plan of action shall be delivered to Buyer within ten (10) business days after completion of same.

E. PROVISIONS

1. This warranty shall begin as of the date of this Agreement, shall be perpetual, and shall survive any other expiration of warranty period or the termination of this Agreement. This warranty shall not be modified except by written agreement signed by both parties.

2. Any provisions of the License or other Agreements which limit or eliminate the liability of either party shall have no application with respect to the Year 2000 Compliance Warranty set forth herein.

3. In the event that Buyer is entitled to modify the Product pursuant to any Licensed or other Agreement, Buyer agrees that it shall not modify the Product in any manner which would affect the performance of the Product in such a manner as to cause it to fail to meet the Year 2000 Compliance Technical Criteria (as defined in Section A).

4. There shall be no Liability on the part of Seller for any failure of the Product to conform to the Year 2000 Compliance Technical Criteria (as defined in Section A) to the extent that any such failure is attributable to a modification of the Product by Buyer.

5. In the event of any conflict or apparent conflict between the terms and conditions of the License or other Agreements and the terms and conditions of this Year 2000 Compliance Warranty, the terms and conditions of the Compliance Warranty shall take precedence. Except to the extent otherwise set forth herein, the terms and conditions of the License or other Agreement shall remain in full force and effect.

6. This Compliance Warranty, together with the License or other Agreement, constitutes the entire agreement between the parties with respect to the subject matter hereof.

F. Except as modified herein the Agreement dated _____shall remain in full force and effect.

IN WITNESS WHEROF, THE PARTIES HAVE EXECUTED THIS AGREEMENT AS OF THE DATE FIRST ABOVE WRITTEN.

BUYER:	SELLER:
By:_____	By:_____
Title:_____	Title:_____

TECHNICAL CRITERIA FOR YEAR 2000 COMPLIANCE

I. INTRODUCTION

BACKGROUND

The Year 2000 situation arises from the use of only two digits for the year, ignoring the two digits which denote the century. When computational resources, both hardware and software, begin mixing dates from 19xx and 20xx, various and uncertain results can be produced. At one extreme, the computation may fail immediately at the point of an error, thus alerting users of the problem. At the other extreme, a particular computation may use dates and times in such a way as to not experience any problem whatsoever. The most dangerous results fall in the middle ground. In that case a date usage problem occurs, but the process continues using the incorrect data, without being noticed.

Addressing the Year 2000 problem is an urgent matter. However, addressing the problem without some up-front analysis could impact the overall goal of achieving timely and cost-effective Year 2000 compliance. The organization should begin its Year 2000 compliance program by clearly defining criteria for compliance and establishing baseline standards for going forward.

The purpose of this document is to provide a standard enterprise-level definition of "Year 2000 compliance" and how compliance will be implemented. It is intended to be used by all Company entities as a framework for achieving enterprise-wide compliance. The enterprise-level definition of compliance will identify the technical elements of the Year 2000 challenge for criteria for Year 2000 compliance, and a standard interpretation of the criteria.

Official notification and communication of the compliance definition and standards will be made with the publication of this document to the entire enterprise. This document will undergo revisions as we move forward with the Year 2000 program. Any changes or revisions to this document will be reviewed and approved by the Year 2000 program management team, and formally communicated to the organization as part of the Year 2000 communication strategy.

DEFINITIONS

For the purposes of this document, the following definitions apply:

1. Computational Resources

All applications of programming, primarily software but also including firmware or embedded programming in hardware. All criteria apply to computational resources in combination.

a. Software

Includes operating systems, end-user application programming, third party vendor software, networking software, real-time and batch programming software, telecommunications programming, process control and monitoring software, etc.

b. Firmware and Microcode

Includes PLCs, EPROMs or any other programmable hardware changeable by persons other than the OEM.

c. Hardware

Primarily BIOS chipsets, but also includes embedded programming in automobiles, elevators, clocks, HVAC systems, telecommunications, etc. Also applies to the entire combination of electronic equipment used for calculational processes.

2. DBMS

Data Base Management System, such as DB2, Oracle, Sybase, ADABAS, etc.

3. Julian (Ordinal)

Refers to a method of displaying date in which the 2-digit year and the sequential day within that year are shown as “YY.DDD”. For example, September 20, 1996 is the 264th day of that year, so the Julian representation of that date would be “96.264”.

II. TECHNICAL ELEMENTS

The technical elements of the Year 2000 remediation involve the computational processes of accepting, creating, manipulating, and outputting calendar-related information. The primary study effort has been on whether computational resources can properly process the change of century to the year 2000. This is of course a high-risk concern, and should be of primary importance to remediation efforts. However, several other date-related problems exist in association with the Year 2000 date rollover. These are summarized in Table 1.

Table 1. Technical Elements of the Year 2000 Challenge

Element	Description	Example Event	Probable Timing
Century Ambiguity	This is the most common element. Computer represents dates with a 1- or 2-digit year. When computer does not recognize that dates are not all in the 19xx range, the results are unpredictable.	Examples of century ambiguity can appear in the following events:	Examples of events can occur with timing as early as:
	a. Data edits reject years in early 20xx as invalid	a. Bank ATM rejects an otherwise valid bank or credit card with an expiration date of "00".	a. First use of cards issued in 1995.
	b. User interface does not allow 4-digit year to clarify century.	b. Lotus 1-2-3 accepts only 2-digit years which it assumes to be in 19xxly.	b. First need to enter values later than 1999. Has already occurred.
	c. Sorting leaves dates in 20xx and 19xx in jumbled order.	c. Itemized monthly bill lists transaction for Jan 1, 2000 through Jan 15, 2000 followed by Dec. 19, 1999 through Dec. 31, 1999.	c. First monthly data processing in 2000.
	d. Durations such as invoice aging are calculated incorrectly.	d. Invoice age calculated as a ridiculously large number or as negative number, erroneously triggering overdue notices and staggering interest penalties.	d. January, 2000
	e. The century is truncated or changed between entering and retrieving a date.	e. Software stores dates in the 20xx range using DBMS but only passes 2-digit years to the product. DBMS defaults to 19xx and stores.	e. Could happen in 1996 for systems with 5-year time horizon.
	f. Comparing a date in 19xx with a date in 20xx assumes both are in 19xx.	f. Payroll-deduction calculations for years in 20xx incorrectly mistake the year as 19xx and fail to apply recent changes in tax laws.	f. First quarter of 2000.

Table 1. Technical Elements of the Year 2000 Challenge (continued)

Element	Description	Example Event	Probable Timing
Extended Semantics	In general, specific values for a date field are reserved for special interpretation. The most common example is interpreting "99" in a 2-digit year files as an indefinite end date, i.e. "does not expire". Another is embedding a date value in a *non-date* data element.	Some software may erroneously process a transaction with a valid end date in 1999 - such as not terminating an expired software license or failing to age back-up tapes for recycling as scratch tapes.	Will occur on various days after Dec. 31, 1998.
Calendar Errors	Errors typically include failing to treat 2000 as a leap year and converting incorrectly between date representations. Day-of-week may also be incorrect, since the year 2000 begins on a Saturday, while 1900 begins on a Monday.	Logic sensitive to day-of-week will be two days off at the beginning of the year, and an additional day off after February 28, 2000. Calculating day of week for all dates following this will be incorrect.	Day of week error will occur Jan 1, 2000. Leap year error will occur the first time input data contains Feb. 29, 2000.
Date Overflow	Many computer products represent dates internally as a base date/time plus an offset in days, seconds, or microseconds since that base date/time. Integers holding the offset value can overflow past the maximum corresponding date - an event which may lead to undefined behaviors.	Value for date can revert to a date near the base date/time, to a negative value, or crash the computer because of an illegal operation.	Happened in the 1980s on certain Tandem hosts. Could happen again at any time to any product depending on how product stores dates.
Inconsistent Semantics	At interface between systems, each side assumes semantics of data passed; systems must make same century assumptions about 2-digit years.	Software on one side assumes all dates in 19xx. Software on other side assumes years 51-99 are 19xx, and 00-50 are 20xx.	Could happen in 1996 for software that stores date values 5 or more years into the future.

III. CRITERIA FOR YEAR 2000 COMPLIANCE

This document requires that computational resources satisfy the General integrity and Date integrity criteria, and either the Explicit or Implicit century criteria. It is preferred and recommended that both the Explicit and Implicit criteria be met if possible, although meeting one or the other of these criteria is acceptable. Resources (hardware, software, or "firmware") that meet these conditions will be considered "Year 2000 Compliant". These criteria are listed in Table 2.

Table 2. Four Criteria for "Year 2000 Compliance"

Criterion	Description
General integrity	No value for *current date* will cause interruptions in desired operation.
Date integrity	All manipulations of calendar-related data (dates, durations, days of week, etc.) will produce desired results for all valid date values within the operational domain.
Explicit century	Date elements in interfaces and data storage permit specifying century to eliminate ambiguity.
Implicit century	For any date element represented without century, the correct century is unambiguous for all manipulations involving that element.

Each criterion as described in this table is intended to be a general requirement. The following sections describe the criteria in more detail.

A. General Integrity

As a system date advances normally on a computer resource, each date roll-over must not lead the computer resource (including, but not limited to, the host processor and any software executing there) to erroneous processing. This must also be true if the system date is regressed to a prior date. All date roll-overs must be transparent to the user.

The best-recognized, high-risk date change is roll-over to 2000, although all other roll-overs such as Feb. 29 also apply. The term "desired operation" in Table 2 is intentionally broad and must be interpreted for specific technologies and applications.

B. Date Integrity

This criterion primarily covers the correctness of manipulations of date data as described in Table 3. These manipulations need to be reliable only over the range of dates that a computer resource is expected to handle.

For example, sales-order processing may handle dates from 5 years in the past to one year in the future. In contrast, an employee database may store dates of birth from early in the 20th century to planned retirement dates well into the 21st century.

Table 3. Variety of Manipulation of Date Data

Category	Examples of Manipulation
Arithmetic	• Calculate the duration between two dates • Calculate date based on starting date and duration • Calculate day of week, day within year, and week within year • Hashing calculation using year as divisor
Branching	• Compare two dates
Format	• Convert between date representation (YMD, Julian, etc.) • Reference same data address with different variables
Data Storage	• Storing and retrieving • Sorting and merging • Searching • Indexing on disk file or database table • Moving data within primary memory
Extended Semantics	• "99" as special value for year • "99.365" as special value for Julian year • "00" as special value for year

C. Explicit Century

This criterion essentially requires the *capability* to store explicit values for century.

For example, third-party products that can use a 4-digit year in all date data elements stored and passed across each interface (including the user interface) would satisfy this criterion. A base-and-offset representation of dates that covers all centuries of interest would also satisfy this criterion. Whether this capability *should* be used to eliminate century ambiguity is part of the last criterion.

D. Implicit Century

This last criterion requires that, if the century is not explicitly provided, its value can be correctly inferred with 100% accuracy from the value of date provided.

For example, the range of values for an "invoice date" would very rarely span more than 10 years. Because the century can always be guessed correctly from an invoice date with a 2-digit year, this date data element would satisfy this criterion.

Note that this criterion permits cost-risk trade-offs that minimize changes to existing date formats.

IV. INTERPRETATION OF THE CRITERIA

A. STANDARD INTERPRETATION

Although these four criteria fully define Year 2000 Compliance, compliance represents a balance between cost and risk rather than an absolute yardstick. Such a balance will vary with each organization, according to its business needs and technological base. Consequently, organizations will possibly require a greater level of detail to absolutely interpret how these criteria apply to that organization.

Table 4 contains the standard interpretation of these criteria. Any deviation from this interpretation in a Company organization must be documented and approved by both the organization and by the provider of the computational resource.

Note the importance of clearly identifying the specific date ranges for compliance, reasonable latitude in date format, and situations under which implicit century values will be tolerated. Also note that certain exceptions are included to support important options for cost/risk trade-off.

Table 4. Interpretation of Year 2000 Compliance Criteria

Criterion	Description of Criterion	Interpretation of Criterion
General Integrity	No value for current date will cause interruptions in desired operation.	• All computational resources will function correctly, without human intervention, and transparent to the user, for all values of system date between 1900-01-01 and 2050-12-31 • Of special interest are the following dates and the ability to roll over forwards and backwards to the correct next date: 1998-12-31, 1999-09-09, 1999-12-31, 2000-01-01, 2000-02-28, 2000-02-29, 2000-03-01, 2000-12-31, 2001-01-01, 2027-12-31.
Date Integrity	All manipulations of calendar-related data (dates, durations, days of week, etc.) will produce desired results for all valid date values within the operational domain.	• Computing resources must correctly handle all representation and manipulation of dates with values between 1900-01-01 and 2050-12-31. Especially important is that all years divisible by 4 in this 150-year range are leap years except 1900. • All computational resources developed for the Company must initialize all date elements with either all zeros (0000-00-00) or null values. Null values are defined for each application by the development facilities, such as the language compiler. A null-value feature is strongly recommended in third-party product selection. • All developed software must not contain literals or constants for dates unless required to capture specific business rules such as calculations of payroll deductions. • All developed software must not use special date values as logical flags, such as “99” as year to mean “no end date” or “00” to mean “does not apply”. **Exceptions:** • Valid date ranges in existing developed or existing third-party software may start with the oldest date value in the application’s archived data rather than 1900-01-01 when there is no business need to support earlier dates.

Table 4. Interpretation of Year 2000 Compliance Criteria (continued)

Criterion	Description of Criterion	Interpretation of Criterion
Explicit Century	Date elements in interfaces and data storage permit specifying century to eliminate date ambiguity.	• All developed and third-party software must permit the use of date formats which explicitly specify century in all date data stored or transmitted. The format of these date elements must be YYYYMMDD or YYYYJJJ as specified by ANSI X3.30-1985(R1991) unless superseded by another application-specific standard or convention. • In storing or transmitting date data, some applications must conform to domain-specific standards, contractual agreements, or APIs to necessary third-party products whose date formats must supersede ANSI X3.30 as appropriate in the application. Examples in Table 5. • Third-party products must permit formatting data with explicit century in the user interface. • All developed applications using third-party products must always explicitly supply century and never rely on those products' default value for century. **Exceptions:** • For date data formatted for a user interface, it is acceptable to use punctuation (slash, hyphen, period, comma) within a formatted date, to spell out or abbreviate the name of the month, or to reorder year-month-day to serve customs among the end users. • DBMSs which cannot store date in conformance with SQL standards but do store century explicitly (such as DD-MMM-YYYY) are acceptable. • Default values for century are permitted only when supplied by data-entry aids and the end-user can verify the defaulted value before committing the data.
Implicit Century	For any date element represented without century, the correct century is unambiguous for all manipulations involving that element.	• Century must be explicit in all date data stored or transmitted unless the correct century can be inferred with 100% accuracy based on the value for date. Explicit century is preferred where practical. • Developed and third-party software may imply century in the user interface in the format YYMMDD or YYJJJ (as specified by ANSI X3.30). • In storing or transmitting date data, some applications must conform to domain-specific standards whose requirements for dates may supersede ANSI X3.30 as appropriate within the application. Examples of these standards are listed in Table 5. **Exceptions:** • For date data formatted for a user interface, it is acceptable to use punctuation such as slash within a formatted date, to spell out or abbreviate the name of the month, or to reorder year-month-day to serve customs among the end users.

Table 5. Additional Year 2000 Compliance Criteria Interpretations

Criterion	Description of Criterion	Interpretation of Criterion
Leap Year Calculation	For any year that is either evenly divisible by 400 or evenly divisible by 4 and not evenly by 100, there are potential exposures.	• Day-in-year calculations must address 366 days not 365. • Day-of-the-week calculations must address the fact that 28 February 2000 is a Monday and 1 March 1 is a Wednesday, not a Tuesday which is February 29, 2000. • Week-of-the-year calculations. The 11th week of the year 2000 is 5 through 11 March, not 6 through 12 March.
Special Values	Fixed dates cannot be used as a global indicator.	• Certain years cannot be used as an "end of input" flag, e.g. 99 and 00. • Certain dates cannot be used to indicate "no-expiration", e.g. 12/31/99.
Century Calculations	All manipulations of century data will produce desired results for all valid date values within the operational domain.	• Rollover to 1/1/2000 - The calculation of 12/31/1999 23:59:59 plus 1 second must produce 1/1/2000 00:00:00. • Pre-2000 Calculations - The calculation of 12/31/1999 plus 1 day must produce 1/1/2000. • Post-2000 Calculations - The calculation of 1/1/2000 less 1 day must product 12/31/1999.

B. STANDARD DATE FORMAT

Standardizing the format for date data is an important part of Year 2000 compliance. However, although several standards for date data format are available, the criteria in this document take precedence over other date standards. These other date standards may be used, as long as the criteria in this document are met.

Furthermore, two considerations must be made when evaluating computing resources for compliance.

1. Limitations in Standards

None of the 3 standards for date representation (ANSI, ISO, FIPS) mandates a 4-digit year for ALL calendar data. For example, conformance to ANSI X3.30 does not eliminate century ambiguity from all date variables and interfaces. Instead, conformance simply reduces the variety of formats occurring in the computing resource.

2. Accommodating Conflicts

While trying to conform to ANSI X3.30, some applications may need to satisfy other standards or conventions for date representation. Table 5 lists examples of standards with date representations that may supersede ANSI X3.30 in specific applications. In addition, the criteria and performance expectations set forth in this document take precedence over all other standards or conventions.

Table 6. Examples of Standards which may Supersede ANSI X3.30

Domain	Standard
Interoperability with international concerns	ISO 8601 (1988)
SQL	ANSI X3.135-1992, ISO-IEC 9075:1992, or FIPS 127-2
Electronic commerce (EDI)	ASC X12 EDI draft std for trial use, ISO 9735, UN/EDIFACT

NUCLEAR UTILITY YEAR 2000 READINESS

Appendix D

VENDOR READINESS QUESTIONAIRE

Year 2000 Ready?	**Software and Hardware Vendors** **All Other Suppliers**	**answer Sections A, B, C, & D** **answer Sections A, B, & E**
Section A: Company General Information		
1. Vendor name, Address		
2. Vendor Internet Page(s) dedicated to Year 2000	http://	
3. Y2K Vendor Contact, Address		
4. Y2K Vendor Contact's E-mail Address		
5. Y2K Vendor Contact's FAX (XXX) XXX-XXXX		
6. Y2K Vendor Contact's Phone Number (XXX) XXX-XXXX		
Section B: Year 2000 Ready?		
1. Product Name 2. Product Type / Category	☐ Software ☐ Business Application ☐ System Software ☐ Office Productivity Software ☐ Product has software or microprocessor component ☐ Hardware ☐ Computer Hardware ☐ Equipment / Device ☐ PC / Workstation ☐ Other ☐ Product does not have software or microprocessor component	
3. If Software, Current Release Number 4. If Hardware, Model Number		
5. To the best of your knowledge, is this product Year 2000 ready? *(A product is Year 2000 ready when it can be proven that date changes between 19xx to 20xx can be performed without error.)*	☐ Yes Please review & sign Warranty Letter attached. ☐ No	

6. If Yes, what is the basis for your answer?	☐ No Date/Time Dependencies in Product ☐ Code Analysis Performed ☐ 4-digit year is used (ccyy) ☐ Date encoding is used (Convert yy from decimal to hexadecimal, etc.) ☐ Windowing technique is used (yy less than 50 means cc equals 20) ☐ Century indicator is used (1 digit where 0=cc of 19 and 1=cc of 20) ☐ Product has been tested and is proven to be ready ☐ Other**:** _____.
7. If No, do you have a solution to make the product Year 2000 ready?	☐ Yes Scheduled release number: _____ Scheduled release date of ready product (mm/ccyy) _____ Indicate what method will be used? ☐ 4-digit year (ccyy) ☐ Date encoding (Convert yy from decimal to hexadecimal, etc.) ☐ Windowing technique (yy less than 50 means cc equals 20) ☐ Century indicator is used (1 digit where 0=cc of 19 and 1=cc of 20) ☐ Other: _____ ☐ No ☐ No plan exists at this time ☐ Patch available/being developed for limited readiness Scheduled release number: _____ Scheduled release date of patch product (mm/ccyy) _____ ☐ Work is in process to make product Year 2000 ready Scheduled release number: _____ Scheduled release date of ready product (mm/ccyy) _____ ☐ Replace existing product. No further support of this product is planned beyond 2000: Recommended replacement: _____

Section C. Strategy/Solution Identification for Year 2000 Readiness (Hardware & Software vendors)	
1. Will there be an additional charge to the client for upgrading?	☐ Yes Cost = _____ ☐ No Upgrade is part of maintenance/contract agreement ☐ Undecided
2. Will you warrant product against failure? 3. Will you provide a maintenance agreement?	☐ Yes ☐ No ☐ Yes ☐ No
4. Will you provide a copy of the test plan used to ensure readiness? 5. Will you provide a copy of the test data used to ensure readiness? 6. Will you provide written confirmation of readiness?	☐ Yes ☐ Attached ☐ No ☐ Yes ☐ Attached ☐ No ☐ Yes ☐ Attached ☐ No
7. Will installation of the Y2K-compliant release require upgrades to the operating environment (i.e. Operating System, DBMS, etc.)? 8. Will installation of the Y2K-compliant release require modification to existing application data? 9. Will the changes implemented in the Y2K-compliant release have any additional performance impact on data? 10. What functionality is impacted by date processing/where do dates play a role in processing?	☐ Yes ☐ No ☐ Don't Know If Yes, please describe: ☐ Yes ☐ Conversion utility will be supplied ☐ No ☐ Don't Know ☐ Yes ☐ No ☐ Don't Know If Yes, please describe:

Section D: Readiness Details/Checklist for Determining Year 2000 Readiness (Software & Hardware Vendors)	
1. Expected Fail Date: When will the product be impacted by a year 2000 date field (mm/dd/ccyy)?	☐ _____ ☐ N/A
2. Does the product represent the year using 4 digits:	
On Screens	☐ Yes ☐ No ☐ Don't Know
On Reports	☐ Yes ☐ No ☐ Don't Know
Within Programs	☐ Yes ☐ No ☐ Don't Know
Within Databases	☐ Yes ☐ No ☐ Don't Know
3. Does this product perform date calculations?	☐ Yes ☐ No ☐ Don't Know If Yes, please describe:
4. Does this product perform logical ordering / sequencing of dates?	☐ Yes ☐ No ☐ Don't Know If Yes, please describe:
5. Does this product have date fields or date-related variables in the programming code?	☐ Yes ☐ No ☐ Don't Know
6. Does the product's files/databases contain 1- or 2-byte indicators to indicate the century? (e.g. 1 for 19, 2 for 20)	☐ Yes ☐ No ☐ Don't Know Please describe:
7. Are there identifier fields that use dates embedded within the field? (e.g. Policy Number X3700**121096**)	☐ Yes ☐ No ☐ Don't Know Please describe:
8. Are there hard-coded dates (e.g., literals 99, 01, 19) within the product? For example, product uses 19 as century and/or 99 as an end-of-file indicator.	☐ Yes ☐ No ☐ Don't Know Please describe:
9. Does the product use the computer operating system date within calculations or comparisons?	☐ Yes ☐ From Server? ☐ From Workstation? ☐ No ☐ Don't Know
10. Does the product use common date routines?	☐ Yes ☐ No ☐ Don't Know
11. Are future dates used (e.g. 1998, etc.)?	☐ Yes ☐ No ☐ Don't Know

12. Does the product currently process dates beyond the Year 2000 in this product? (e.g. 2000, 2005, etc.)	☐ Yes ☐ No ☐ Don't Know If Yes, please provide an idea of how long Year 2000 dates have been used in this product ☐ Less than 1 year ☐ 1-3 years ☐ Greater than 3 years
13. How far into the Year 2000 do the dates extend? (e.g. 2010, 2034, etc.)	
14. Do date fields require expansion from 2 digits to 4 digits: On Screens? On Reports? Within Programs? Within Databases?	 ☐ Yes ☐ No ☐ Don't Know ☐ Yes ☐ No ☐ Don't Know ☐ Yes ☐ No ☐ Don't Know ☐ Yes ☐ No ☐ Don't Know
15. Are there any regulatory requirements that stipulate expansion of date fields from 2 digits to 4 digits? (e.g., adherence to govt. standard for expiration of pharmaceutical products.)	☐ Yes ☐ No ☐ Don't Know If Yes, please describe:
16. Does this product interface with other vendors' products?	☐ Yes ☐ No
17. If Yes, what are they and have the interfaces been tested?	Product: Tested: ☐ Yes ☐ No Product: Tested: ☐ Yes ☐ No Product: Tested: ☐ Yes ☐ No Product: Tested: ☐ Yes ☐ No
18. Does your product recognize Year 2000 as leap year?	☐ Yes ☐ No
19. Are there any hardware attachments to the application? i.e. An inventory system may require use of a Bar Code Wand.	☐ Yes ☐ No If Yes, please describe:

Section E: Readiness Details/Checklist for Determining Year 2000 Readiness (All other Suppliers)	
1. Is the manufacturing of your product dependent on any other critical suppliers or third party vendors?	☐ Yes ☐ No ☐ Don't Know
2. If Yes, have you had any discussions with those suppliers regarding their Year 2000 readiness?	☐ Yes ☐ No ☐ Don't Know
3. If Yes, will your suppliers be Year 2000 ready?	☐ Yes ☐ No ☐ Don't Know
4. Do you have any manufacturing equipment with Year 2000 issues?	☐ Yes ☐ No ☐ Don't Know
5. If Yes, will your manufacturing equipment be Year 2000 ready?	☐ Yes ☐ No ☐ Don't Know
6. Have you assessed the impact of Year 2000 on your business systems?	☐ Yes ☐ No ☐ Don't know
7. Are your business systems Year 2000 ready?	☐ Yes ☐ No ☐ Don't know
8. If No, do you have a plan for making your business systems Year 2000 ready?	☐ Yes ☐ No ☐ Don't know
9. If Yes, what is your targeted Year 2000 readiness date?	

Please return completed questionnaire(s) within 14 days to:

NUCLEAR UTILITY YEAR 2000 READINESS

Appendix E

DETAILED ASSESSMENT PROCEDURES AND CHECKLISTS

Assessment Plan

I. Purpose

The purpose of assessment is to complete the assessment begun by IBM and to determine the magnitude of the impact from the year 2000 and the risk from the year 2000 on a company wide basis. It is expected that at the completion of this plan a company wide inventory of systems and non-IT assets with an assessment of year 2000 impact and risk will be produced.

II. Methodology

Assessment was initiated by IBM using a survey. Response to this survey was inadequate to complete the assessment. Further information will be gathered by the corporate year 2000 project team by continuing the collection of surveys from the Business units and through meetings with each Business unit's year 2000 project point. This information will then be used to complete the assessment plan.

The business units will use the following assessment plan to inventory and analyze impact and risk of their assets. The business units will make the preliminary assessment. The corporate project team will review the business units findings and audits will review the overall findings. In cases where the business unit assessment does not agree with the corporate assessment a consensus process will be followed to achieve a consensus assessment. If audits does not concur with the consensus assessment then the corporate project team will coordinate resolution of the concern with audits.

III. Assessment Plan

A. Prepare an Inventory

1. Prepare an inventory of all Information Technology assets. This includes networks, operating environments, databases, application programs, CASE tools, off the shelf products, etc.. These are assets usually maintained by IS personnel.

2. Prepare an inventory of all Non-IT assets. These are systems or devices that are usually device driven chips, EPROMS, or other PLCs and which may be maintained by users. Examples include data acquisition systems, badge control systems, environmental control systems, engineering applications, plant control systems, workstations, end user maintained spreadsheets and databases,

trending applications, plant monitoring systems, LANs and LAN equipment, PBXs and other telephony equipment, PCs, test equipment, and metering systems.

3. Prepare an inventory of all external interface systems that transfer electronic data. These include any EDIs, and interfaces with other companies, regulatory agencies, public domain networks such as the Internet, interfaces with other utilities of Qualified Facilities, interfaces with other distribution systems.

B. Assess the Size of the Asset In Terms of Amount of Code

The purpose of this part of the assessment is to get a feel for the size of assets in terms of Lines of Code or functionality. This assessment is essential for determining resource allocation and is used to bias the assessment of an asset's Y2K impact and risk. It should be noted that a large variety of asset types are being inventoried and a Lines of Code metric for size is not applicable to all assets, hence the inclusion of functionality. The purpose is to attempt to create a common size rating system for all assets. The following definitions/categories are provided as a guideline for assessing asset size.

1. Applications written and maintained by the Business unit or Corporate IS organization shall be assessed for size using the following categories:

    a. Minor, 0 -1000 Lines of Code
    b. Medium, 1001 - 10000 Lines of Code
    c. Major, Greater than 10000 Lines of Code

2. Applications written and maintained by Vendors should be evaluated based on functionality if a Lines of Code count is not available. Assess size using the following categories:

    a. Minor Vendor, 0 - 1000 Lines of Code, or very limited functionality, probably dedicated to a single, limited function, or limited to operating on a single CPU with limited memory resources
    b. Medium Vendor, 1001 - 10000 Lines of Code, or moderate functionality, probably able to generate and print reports, perhaps handle multiple functions, or limited to operating on a single CPU with several megabytes of memory resources, or operating on a few CPUs with limited memory resources
    c. Major Vendor, greater than 10000 Lines of Code, or incorporates major functionality, probably able to generate and print reports, maintain and manipulate data, has a sophisticated user interface, perhaps handles several major functions, or operates on a single CPU with large amounts

of memory resources, or operates on several CPUs with large amounts of memory resources

3. Applications that are end user generated and maintained should be evaluated based on functionality if a Lines of Code count is not available. Assess size using the following categories:

    a. Minor Ad hoc, 0 - 1000 Lines of Code or applications written to achieve a specific purpose such as generate a report from a database or performing a specific type of calculation using a spreadsheet format, or a single user, stand alone system that perhaps uses the network to access data, but has no networking capability of its own
    b. Medium Ad hoc, 1001 - 10000 Lines of Code or applications written to perform complex user purposes but limited to a few reports, or a few data manipulations, or capable of supporting a small number of users in a small network or workstation
    c. Major Ad hoc, greater than 10000 Lines of Code, or applications written to perform complex user purposes with large numbers of different reports and data manipulations, can provide what if type analysis, or capable of supporting large numbers of users in a organization wide network

4. Applications that are purchased off the shelf should be evaluated based on functionality if a Lines of Code count is not available. Assess size using the following categories:

    a. Minor Off Shelf, 0 - 1000 Lines of Code or applications written for a single user, single machine
    b. Medium Off Shelf, 1001 - 10000 Lines of Code or applications written for a small number of users in a small network or workstation.
    c. Major Off Shelf, greater than 10000 Lines of Code, or applications written for a large numbers of users in a organization wide network, or a client server application.

5. Operating systems should be evaluated based on the platform they are for. Assess size using the following categories:

    a. Minor Op Sys, Operating systems for PCs
    b. Medium Op Sys, Operating systems for minis, work stations, or LANs
    c. Major Op Sys, Operating systems for mainframes, client server, intranets, or WANs

6. Embedded systems should be evaluated on functionality. Assess using the following categories:

    a. Minor Embedded, single or limited function, has a single CPU

b. Major Embedded, large amount of functionality, has multiple CPUs

7. Commercial products like programming languages/environments, database managers, spread sheets, and word processors should be evaluated based on the platform they are designed to service. Assess using the following categories:

   a. Minor Package, those designed to operate as stand alone on PCs
   b. Medium Packages, those designed to operate on minis, workstations, or LANs.
   c. Major Packages, those designed to operate on mainframes, client server, or WANs.

8. Miscellaneous assets such as PBXs, Data Acquisition Systems, Relays or other smart devices, CASE tools, etc. should be evaluated based on their perceived size. This is a quality judgment. Assess these components using the following categories:

   a. Minor Misc, those assets perceived to be of small size (example is a relay)
   b. Medium Misc, those assets perceived to be of moderate size (example is a Data Acquisition System)
   c. Major Misc, those assets perceived to be of major size (example is a PBX)

C. Assess Importance of the Asset to the Business unit

Use the following definitions to determine the importance of the asset to the Business unit:

Critical	Has life threatening implications to employees/customer Required by regulatory agencies for Business unit/company operation Major implications on financial status/stability Major impact on service to customers Major impact on stockholders/public relations Is a binding contractual obligation to customer Daily loss of revenue of greater than $750,000.00
Severe	Severe impact to Business unit/company operation; becomes more critical over time Business continues but with great difficulty Mandated by regulatory agencies but can be lost for short periods of time Cash flow implications increase as outage duration extends Lost productivity to most of the employees Daily loss of revenue of greater than $500,000.00 Asset is used solely as a backup to an asset of critical importance

High	Business operation continues but with serious difficulty Mandated by regulatory agencies but which have compensatory measures Lost productivity to a majority of employees Daily loss of revenue of greater than $100,000.00 Asset is used solely as a backup to an asset of severe importance
Medium	Business operation continues but is cumbersome Compensatory measures are more costly to use than the asset Minimal impact on the Business unit/company's core business Minimal impact on cash flow Lost productivity to a significant number of employees Asset is used solely as a backup to an asset of high importance
Low	Minimal impact to Business unit operation Lost productivity to a minimal number of employees Customer service not affected Compensatory measures are minimally more costly to use than the asset Asset is used solely as a backup to an asset of medium importance
None	No impact to business operations No lost productivity Compensatory measures are no more costly to use than the asset Asset isn't being used or has no identified users Asset is used solely as a backup to an asset of low importance Asset has been replaced or superseded

Assets determined to have no importance should be evaluated for abandonment. If it is determined that these assets can be abandoned then no further resources should be used in evaluating them.

D. Assess Impact of Year 2000

1. This is done to solely assess the impact of year 2000 dates on the asset. Impact will not be used as the sole criteria for determining corrective actions.

2. Evaluate frequency of use of date/time data. Range of frequencies will be not used extensively. This is a qualitative judgment, for applications or systems with thousands of lines of code, infrequent use may be once per 1000 lines while frequent use may be once per 100 lines. Non-IT assets may be considered to use dates frequently if it is used once in PLCs, as part of the system clock, if used for timing, or if used to date stamp data.

3. Evaluate how the date/time data is used. Range of how used will be from display only to used to calculate critical values. Display only uses are for graphs, printouts, or screen displays that show a date, critical values would use dates to determine plant or process control functions, generate billing, control functions which could impact personnel safety, or if use of the date would cause the system or program to crash. Intermediate impact values include applications where dates are used to format record lengths, for forecasting, for determining reporting intervals, for generating required filings, for date stamping on legal records, controlling building access, generating trend reports or graphs, etc. These functions should be evaluated qualitatively for their importance to the business unit when determining where the function falls with relation to critical values and display uses only.

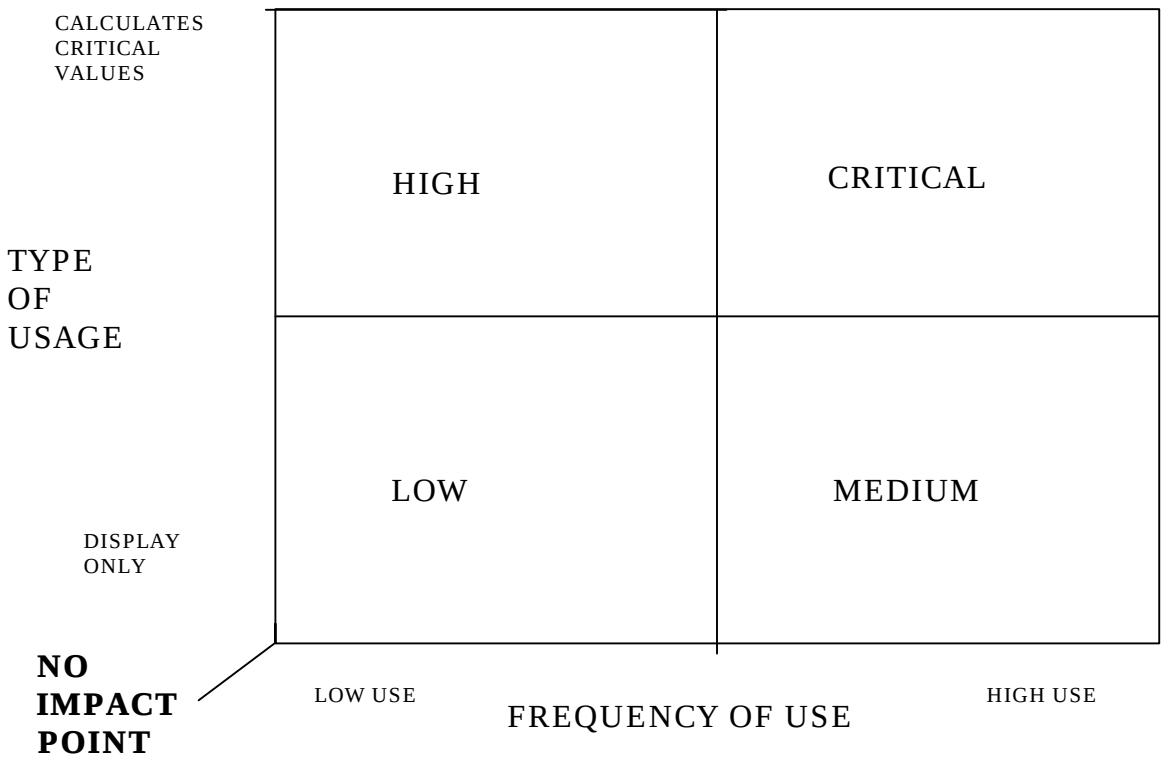
4. Several methodologies may be used to perform the above evaluations. The method that generates results with the highest level of confidence is testing. However, this method is time and resource intensive for large systems and may not be practical for non-IT assets. Other acceptable methods, in order of highest confidence to lowest confidence are use of a tool to evaluate code, vendor certifications of year 2000 compliance or notices of problems, code inspections, and engineering analysis. It is important to record the method used to perform the evaluation as that information will be used in evaluating risk.

5. Evaluate overall impact using the below grid and rules to classify year 2000 impact for each system.

    a. Plot each asset on the grid. The asset may be represented by a series of points based on frequency of use.

    b. Determine the overall impact of each asset by choosing the plotted point with the greatest impact. Note that any asset with a date influenced critical value will be given a critical impact rating. Also note that only assets that do not use date data will be rated as No Impact, any use of date data requires some ranking.

c.



IMPACT EVALUATION GRID

E. Assess Risk of Year 2000

1. The purpose of assessing risk is to prioritize assets for determining resource allocation. Assets with the highest risk will have the most detailed corrective actions, will receive the most resources, and will be done first. Assets with lessor risk will have corrective actions and resources commensurate with their risk and will be done after the higher risk assets have been completed. Items with little risk may not be completed prior to the year 2000 but will have contingency/action plans in place so that productivity is minimally impacted. These assets will be completed to the year 2000 if resources permit.

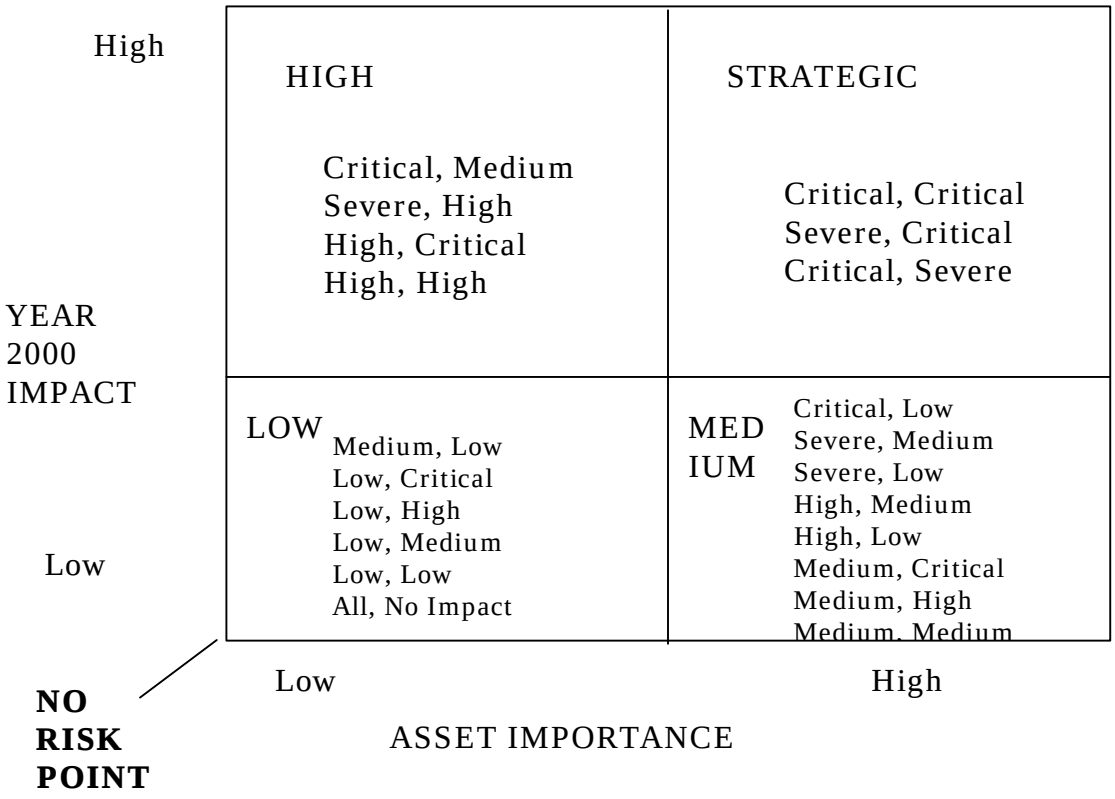
2. Risk is determined based on a combination of asset importance and year 2000 impact. Four levels of risk have been established based on the below risk grid.

3. Note that assets with no importance and no impact will be assigned to the No Risk Point.

4. Risk combination pairs are read as standard Cartesian coordinates, i.e. (x,y) ordered pairs using the ordering (Importance, Impact)

5. The risk rating may be changed or biased based on the confidence in the assessment of impact. If there is low confidence in the methodology used to determine impact, or if the impact rating is suspect, than the risk rating should be raised a level. The risk rating may also be changed based on using the size and frequency of date use of an asset to bias the impact rating. As an example, an asset that is large with many date uses of a less important nature may have its impact rating raised a level due solely to the large number of lessor important type of usage items.

6.



RISK EVALUATION GRID

F. Determine Corrective Actions

1. The purpose of corrective actions is to ensure that the company is ready to operate with the asset once the year 2000 is reached.

HIGH - 2nd Asset repaired/replaced prior to 2000 Asset tested prior to 2000 Asset has contingency plan should failure still occur Optional: backup obtained prior to 2000	STRATEGIC - 1st Priority Asset repaired/replaced prior to 2000 Asset tested prior to 2000 Asset has contingency plan with compensatory measures should failure still occur Optional: backup obtained prior to 2000
LOW - 4th Priority Asset has a repair/replacement plan which may go beyond 2000 Asset has contingency plan with compensatory measures should failure still occur	MEDIUM - 3rd Asset repaired/replaced prior to 2000 Asset tested prior to 2000 Asset has contingency plan should failure still occur

CORRECTIVE ACTION GRID

2. Corrective actions can consist of repairing/replacing the asset, testing the asset, generating compensatory/action plans, doing nothing, or a combination of these items.

3. Corrective actions should be commensurate with the risk to the asset. The higher the risk the more extensive and proactive the actions. Only assets with low risk should be given actions that are reactive or post year 2000.

4. Minimum corrective actions are specified in the above corrective action table.

5. Format for action and contingency plans will be published in a later document.

6. Assets with no risk shall have nothing done unless the confidence in the no risk rating is low, for those assets a compensatory action plan should be prepared.

IV. Document Results

A. The Business units shall report assessment results to the Corporate Project Office, G.O. 1, Room 115 addressed to Keith Wilcox or Murray Jennex

B. Reports shall be in a Microsoft Excel/Access compatible format.

C. The Corporate Project Office shall serve as the repository for all the reports and shall be responsible for generating the overall inventory as well as any required sorts of the inventory. The format for the final inventory will be decided using corporate data standards. However, it is anticipated that the final inventory will be published on the year 2000 web page (currently under development).

D. Required data fields are as follows:

1. Asset Acronym: provided by the Business unit if one exists

2. Asset Name: provided by the Business unit

3. Asset Version: provided by the Business unit

4. Asset Description: provided by the Business unit

5. Asset Language, i.e. what language the asset is written in or uses: provided by the Business unit

6. Asset Size: use rating from the assessment plan

7. Asset Importance: use rating from the assessment plan

8. Y2K Impact: use rating from the assessment plan

9. Y2K Impact Assessment Basis: testing, vendor certification, inspection/engineering evaluation

10. Asset Y2K Risk: use risk rating from assessment plan

11. Correction Strategy: Business unit will stipulate, Corporate Project Office will review, disagreements to be resolved via consensus decision process

12. Correction Priority: Corporate Project Office will establish this based on overall inventory results, Business units will review, disagreements to be resolved via consensus decision process.

13. Correction Estimate: Business unit will stipulate, Corporate Project Office will review, disagreements to be resolved via consensus decision process

14. Source Code Location: provided by the Business unit. Indicate the physical location where the source code is stored or indicate “Not Available” if the source code is not available. Availability of the source code should be taken into consideration when determining corrective actions. Replacement should be

considered for any asset which does not have available source code. If the source code is not available because the vendor kept it, so indicate and the corporate project team will initiate actions to either obtain the source code, obtain assurance of compliance from the vendor. If an upgrade or replacement package is required then the Business unit will need to decide if the asset is to be upgraded or replaced and should initiate the appropriate actions.

15. Primary Users: indicates which organization or group is the primary users of an asset. This is provided by the Business unit

16. Contact Name: indicates the individual or lead individual responsible for maintaining the asset

Year 2000 Detailed Assessment Package

1. Purpose

The purpose of this package is to guide the user through the Year 2000 Detailed Assessment Process of a particular application, and it also serves as documentation of the work performed. The purpose of a detailed assessment is to obtain enough information about an application to determine its expected performance beyond December 31, 1999. From this assessment, a decision is made (and documented) regarding any action needed to maintain continuous performance.

2. Application Information

Information specific to this application is required in order to complete this detailed assessment. Enclosure A has been pre-populated with as much of that information as we currently have available.

2.1. Review Enclosure A for accuracy and fill in any missing information as applicable.

3. Scanning the Application

In order to determine if an application is Year 2000 ready, scanning may be required. Scanning is a process (manual or automated) that locates all date references and potential calculations in an application. In order to be able to do scanning the source code of the application is required along with all of the applications associated with interfaces, modules, screen layouts, etc. Because of the complexity involved, only the application developer or comparable expert should undertake the process. If the number of lines of code exceeds 1000 then you can contact the NY2K Project Manager or your local NY2K Core Team member and they can make arrangements to have your code electronically scanned for date impacts. If you have <1000 lines you can manually view the code looking for date impacts

NOTE: *Testing is required for SDQA category A or B applications to ensure Year 2000 readiness. For those that are not Category A or B, the business sponsor should determine the appropriate level of scanning or testing, and document appropriately.*

3.1.1. Perform application scanning (if applicable) and complete Enclosure B.

4. Testing the Application

In order to determine if an application is Year 2000 ready testing may be required. Testing involves taking the application out of the normal production environment (into a "safe" test environment where any failures have no impact on production) and performing a series of controlled scenarios that will mimic the application's performance in the Year 2000. Specific testing criteria have been established and documented by the Year 2000 program, and may be found in the Year 2000 Technical Compliance Criteria.

NOTE: *Testing is required for SDQA category A or B applications to ensure Year 2000 readiness. For those that are not Category A or B, the business sponsor should*

determine the appropriate level of scanning or testing, and document appropriately.

4.1. Testing

NOTE: *See Enclosure F for more information on testing.*

4.1.1. Develop Test Plan. (See Enclosure E for additional Test Plan Information)

4.1.2. Identify and List All Application Components.

4.1.3. Identify Baseline Data with 19xx Dates.

4.1.4. Determine Appropriate Test Environment

4.1.5. Setup Test Environment

4.1.6. Follow Appropriate Change Control Procedures for Test Platform.

4.1.7. Load Application into Test Environment.

4.1.8. Perform Test Cycles

4.1.9. Restore Test Environment (if necessary).

4.1.10. Complete Enclosure C.

5. Year 2000 Impact Sign Off

The business sponsor is required to review the entire assessment package to this point (including enclosures), determine if application is Year 2000 Ready, and sign off Enclosure D

5.1. Review assessment package including enclosures

5.2. Determine if application is Year 2000 Ready.

5.3. Complete Enclosure D – Year 2000 Impact Sign Off.

5.4. Mail the completed Detailed Assessment to:

NY2K Project Manager

NOTE: *If the application is not Year 2000 ready then complete a Year 2000 Business Case Package for the application.*

6. NY2K Project Management Review & Sign Off

6.1. The NY2K Project Manager is required to review the entire assessment package to this point (including enclosures) for completion

6.2. Complete Enclosure D – Year 2000 Impact Sign Off.

6.3. File the completed Detailed Assessment.

(ENCL A) APPLICATION SUMMARY

The following information is required to complete the detailed assessment, and is/will be stored in the Database. Please complete any areas that have been left blank.

General Application Information

Application Name:	*application name*
Application Number:	*application number from Database*
Functional Description:	*Brief description of the application*
Business Sponsor Name:	*Sponsor name*
Business Sponsor Area:	*Sponsor location/organizational area*

Programmer Information

List the person who is currently responsible for the source code.

Primary IT Contact Name	
Phone	
Team Name	

Vendor Information ❑ N/A

List vendor information (if applicable). Vendor may provide update/upgrade to software, operating system, etc., which may be necessary to achieve Year 2000 readiness.

Vendor Name	
Contact Name	
Address	
City	
State	
Zip Code	
Phone Number	

User Information

User Groups	
User Sites	

(Encl B) Application Technical Summary

(to be completed by application developer or support)

Hardware Platform	
Operating System(s)	
Development Tool(s)	
For databases only: Database type & name	
Server Name	
Executable File Name	
Executable Server Name	
Version Number	**Date Implemented:**

Application Component List ❑ N/A

List all components of the application (any of the separate pieces, such as programs, data tables, interfaces, or any other stand-alone modules that provide functionality to the application).

Component Name	Component Type	Language

Application Interfaces (Internal) ❑ N/A

List any other applications within xxx company that may exchange information with this application (whether receive, provide input, or both), if applicable.

Interfacing Program	Interface Name	Scheduled Interface (Real Time, On Demand, Daily, etc.)	Description of the Interface

Application Interfaces (External) ❑ N/A

List any applications or entities that are external to xxx company (vendors, government agencies, banks, etc.) that exchange information with this application (whether receive, provide input, or both), if applicable.

Interfacing Program	Interface Name	Scheduled Interface (Real Time, On Demand, Daily, etc.)	Description of the Interface

(Encl C) Scanning Results

Was scanning conducted for this application? ❑ Yes ❑ No

Scan Summary

Was the scanning performed electronically? ❑ Yes ❑ No
If YES, attach Detailed Scan Reports.

If NO, provide the following information:

Method used to scan code:	
Number of Lines Impacted:	
Program File Name:	
Line #	Contents of Line with Date Impact
Program File Name:	
Line #	Contents of Line with Date Impact
Program File Name:	
Line #	Contents of Line with Date Impact

Testing

Was testing used to determine Year 2000 Impact? ❑ Yes ❑ No

Testing Summary

Provide a summary of the application test and results or attach a copy of the completed test plan.

Results Table:

Readiness Test	Pass	Fail	N/A
Year 2000 Rollover Warm			
Year 2000 Rollover Gregorian Warm			
Year 2000 Rollover Julian Warm			
Year 2000 Rollover Cold			
Year 2000 Rollover Gregorian Cold			
Year 2000 Rollover Julian Cold			
Year 2000 Leap Year Rollover Warm			
Year 2000 Leap Year Rollover Gregorian Warm			
Year 2000 Leap Year Rollover Julian Warm			
Year 2000 Leap Year Rollover Cold			
Year 2000 Leap Year Rollover Gregorian Cold			
Year 2000 Leap Year Rollover Julian Cold			
High Risk Date 9/99/99			
Date Integrity 2/29/01			

(Encl D) Year 2000 Impact Sign Off

❑ This Line of Business (LOB) supported application has been assessed and is capable of functioning properly in the year 2000 and beyond, as defined in the Year 2000 Technical Compliance Criteria, and by the Business Project Manager. The signature below indicates Y2K certification.

❑ This application is impacted by the Year 2000 and is not ready for 1/1/2000.

Business Sponsor _____ Date _____

❑ Package Reviewed - Complete

NY2K Project Manager _____ Date _____

(Encl E) Test Plans

Overview

Describe the overall testing approach.

Assessment Of Level Of Testing Required

The extent of testing required will depend on company's view of the level of confidence required that the application will function correctly through the Year 2000. This will influence the number and type of test cases produced for Century Test. Other considerations may include the SDQA level of the application.

Specific Aspects to be Tested and not to be Tested (i.e. Dead Code)

List any particular functions that must be tested or that do not need to be tested (i.e. on-line panels known to have critical date processing, batch processing - month-end, year-end, quarterly, weekly, etc.). List particular century test dates to be tested based on the application's date processing.

Quality

Describe when formal quality control checks are to be conducted and what these controls are (i.e. specific sign offs required and when, user involvement and when, etc.)

Acceptance

Describe any specific criteria for user (or support team) acceptance of this application (other than acceptable results of the no-damage testing).

Test Timeline

Give key dates in test cycle, where known, and staff involved (i.e. data ready, test environment in place, Unit Test completed, System Test completed, Century Test completed, Acceptance, and retrofits (if any).

Application Test Environment

Summary of Hardware & Software

- *Describe test areas/regions to be used and any necessary set up for those areas (i.e. CICS region setup, DB2 table setups, etc.).*
- *State hardware to be used for testing.*
- *Identify communication links if required.*
- *Describe access arrangements for testers and any other security issues to be resolved.*

- *Describe operating system and the use of proprietary products i.e. packages such as Changeman.*

Test Data

Describe what test data will be used and how this data will be captured. Describe any backup and restore jobs for this data and where the JCL and/or files reside. Describe any dependencies that this data may have upon another application's data, or common files between applications.

Test Tools

Describe how testing tool(s) are to be used, if at all, by staff involved.

Test Control Procedure

- *Describe how problems will be logged in the Problem Tracking Database (application name used) and what statuses will be used for this application for tracking purposes.*
- *Describe how scripts will be developed and by whom. If multiple scripts are used, describe the order in which they must operate, and any other related dependencies.*

Test Team Organization and Responsibilities

State who is involved and their responsibilities.

Configuration Management

State where all data files and JCL for application set up and testing reside. State where testing documentation and results will reside for this application. State the change control process to be used (i.e. Changeman checkout, Source Safe, etc.)

Assumptions

List assumptions. All decisions based on assumptions should be confirmed in light of new knowledge gained during the course of the project.

(Encl F)Testing Considerations

The following is a guideline for defining the major steps in performing Year 2000 tests on applications. The same process can be followed whether it is for Century Testing or User Acceptance Testing. These steps apply to verifying applications that claim to be Year 2000 ready as well as those that have just been remediated.

- Identify and List All Application Components.
  - Programs
  - Files
  - Database Tables
  - JCL
  - Scripts
  - Sort And Other Utility Control Statements
  - Special Devices (Scanners, Magnetic Strip Readers, Etc.)
- Identify Baseline Data with 19xx Dates.
  - Transaction File Data
  - Test Scripts
  - Test Results (Reports, Screens, Etc.)
- Determine Appropriate Test Environment
  - Determine All Hardware Platform Components Required That Can Be Set To Year 2000 Date(s).
    - Mainframe
    - Midrange
    - Database Server
    - LAN Server
    - Notes Server
    - Workstation
    - Intelligent Peripheral Devices
  - Determine All Components Of The Operating Environment
    - Operating System
    - System Utilities (Sorts, DBMS, Etc.)
    - Run Time Components For All Platforms
  - Perform Sizings To Determine If Adequate Resources Are Available On Test Platform (DASD, Communications, Etc.)
  - Determine If All Components Other Than The Application Are Year 2000 Ready. If Any Are Not Year 2000 Ready, Assess The Risk Of Proceeding With The Test With Platform Elements That Are Not Year 2000 Ready.
- Setup Test Environment
  - Schedule Equipment Required For Test. Make Sure Date(s) for System Initialization Are Clearly Specified.
  - Verify That All Supporting System Software and Components of Other Required Applications Are Properly Installed and Date Initialized.
  - If Production Equipment Is Used, Make Sure That Safeguards Are In Place To Keep Test Data From Bleeding Into Production Environment.

- Obtain Security Access if necessary.
- Schedule Any Equipment Interconnects Required.

- Follow Appropriate Change Control Procedures for Test Platform.
- Load Application into Test Environment.
  - Load Programs, Files, Database Tables, Etc.
  - Warp Dates if necessary.
- Perform Test Cycles.
- Restore Test Environment if necessary.
  - Remove All Test Programs, Data, Etc.
  - Verify That Everything Is Reset To Pre-Test Conditions (IP Addresses, Etc.)
- Obtain Test Sign-Offs.

If these test are on code that has been remediated, then any non-Year 2000 changes that have been made since the code was first checked out for Year 2000 remediation have to be applied. Year 2000 testing should then be re-run to verify that these latest changes have not corrupted the Year 2000 readiness. After all final sign-offs have been received that application can then be put into production.

If a production application was reported to be Year 2000 ready and the tests confirmed that it is, then the process is complete.

All new applications purchased by xxx company are to be Year 2000 ready. Year 2000 tests must constitute part of the normal acceptance testing and the above process should be followed to verify that readiness.

NUCLEAR UTILITY YEAR 2000 READINESS

Appendix F

TEST SPECIFICATIONS

EMBEDDED SYSTEMS TESTING

1. INTRODUCTION

In any Year 2000 project, testing is perhaps the key element of the project. There are two distinct phases in testing.

The first phase is ***Investigative testing*** to ascertain whether a software program, product or integrated system complies with a predetermined set of specifications for the Year 2000.

The second phase is ***Post remediation testing*** to establish that any modifications made as a result of errors found either in the first phase of testing or by other analytical methods, are valid and the system, product or program can be certified to comply with the Year 2000 specification.

Year 2000 compliance may be defined differently for different purposes. In addition, local installations will vary in the way dates and times are formatted and represented. These differences notwithstanding, the kind of compliance testing that needs to be performed can be categorized into date and date-and-time functionality testing.

Most "traditional" business processing environments are concerned more with the date-category than with the date-and-time-category of functionality. In an Instrumentation & Control environment, the date-and-time-category functionality takes on more importance because of the "hard" real-time requirements of process and device control, monitoring, event signaling etc. performed by embedded systems. Of course, some functionality in these systems is centered on (real) time, with no date-related requirements. Since this functionality is not considered a year 2000 compliance issue, it is not addressed here.

This document presents a comprehensive set of test guidelines and methodology for the testing of embedded systems. In the first section, some generally accepted standards and a brief background of date and time notation is laid out. This is followed by a consideration of the unique challenges presented by the embedded systems. Then a set of specifications based on industry standard guidelines is set out. This forms the basis from which test parameters and methods are drawn.

Testing strategy, which includes precautions, preparations, and considerations of functionality, is then explained. This is followed by the detailed test procedures.

Finally extracts from industry sources such as the IEEE and the NIST along with references for further reading are included in the appendix.

2. BACKGROUND OF DATE AND TIME NOTATION

2.1 *International calendar*

The international calendar currently followed is almost all countries is the ***Julian calendar*** with the Gregorian correction, or simply called the ***Gregorian calendar.***

This is a solar calendar i.e. a year is based on the time taken for the earth to revolve round the sun. It consists of 12 months in a year. Each month consists of a specified number of days. Only the second month February consists of 28 days in common years and 29 in leap years. Thus common years have 365 days and leap years have 366 days.

2.2 ***Definition of a leap year***

A leap year is a year where is an extra day (i.e. February 29th). This intercalation of day is to adjust for the discrepancy arising out of the normal year period of 365 days and the actual solar year based on the earth's revolution which is 365.242199 (365 days, 5 hours, 48 minutes and 46 seconds).

2.3 ***How is a leap year determined?***

As per the Julian Calendar every year divisible by four was a leap year. This led to a discrepancy of 3.12 extra days over four centuries. Pope Gregory corrected this in 1582. As per the correction century years are leap years only if they are divisible by 400. There was also a further refinement, which stated that years divisible by 4000 are common years or non-leap years. With these refinements, the discrepancy between the calendar time and the actual solar time is reduced to one day over four thousand years.

As per the current Gregorian calendar the determination of a leap year is as follows:

1. All non-century years divisible by four are leap years.
2. All century years divisible by 400 are leap years.
   This means that 1900 and 2100 are not leap years, while 2000 is a leap year.
3. As per the refinement to the Gregorian calendar we also have the additional clause: All years divisible by 4000
   are common years or non-leap years.

2.4 ***Julian representation of a date***

The Julian representation of a date is the format DDD, YY or DDD, YYYY where DDD is a number from 1 to 365 or 366 depending on whether the year was a leap year or a common year and YY or YY is the two digit or four digit representation of the year.

2.5 ***Gregorian representation of a date***

The Gregorian representation of a date is the format DD/MM/YY, MM/DD/YY or any of the other common formats currently used that incorporate date, month and year.

In a system ,dates may be stored in Gregorian or Julian representation, or a combination of both. There are also situations where all internal representation and calculations are done using the Julian representations and all external interfaces and displays use the Gregorian representation.

3 THE CHALLENGE OF EMBEDDED SYSTEMS

Embedded systems pose many challenges for testing and remediation of the Year 2000 problem. These can be broadly categorized as follows:

3.1 Architectural

- There is a wide prevalence of four bit and eight bit processors such as those manufactured by Intel, Zilog and Advanced Micro Devices. Many of these have a limited instruction set. Many of these microcontrollers have a two-digit date representation for arithmetic and logical operations.

- Date representation may be different for 'power on ' conditions and in battery backup condition

- There is no standard way to encode dates between different vendors.

3.2 Programming

- Source code is not available for many of these systems.
- Object code may be stored in different levels of firmware i.e. Programmable logic arrays (PLA's), Flash ROM, CMOS or BIOS.
- Object code may be hard coded, reloadable or re-entrant.
- Program may be recording real time intervals based on calendar dates rather than actual dates themselves.

3.3 Configuration

- System may be consisting of upstream and downstream devices that have data interfaces between them.
- Downstream devices may have dates that are set or overridden by upstream devices.
- System may have external interfaces that transmit and receive date information.

3.4 Operational

- The system may be in a production environment that it cannot be taken out of without severe impact.
- Backup systems may not be available, in case of failure during testing.
- Many systems may not revert back to current dates after dates are advanced during testing.
- Warranties, inspection and service logs may be voided by date advancement.

4 SPECIFICATIONS FOR CENTURY COMPLIANCE

The rules that follow are taken from the following source:

http://www.year2000.com/archive/gte-article/NFgte-table3.html

They resemble but are not identical with the rules issued by BSI/DISC. In particular the BSI/DISC rules explicitly cover the point that Year 2000 is a Leap Year. However the rules below have been cited by a English lawyer as a possible standard; and given the source, they might be assumed to be a de facto standard for North America. These rules are also currently being studied by the IEEE as the framework for an IEEE specification on Century Compliance.

4.1. General integrity

No value for current date will cause interruptions in normal operation. As a system date advances normally on a system, each system date must not lead to erroneous operation of the system or its software processes. The best

recognized high-risk date change is the roll over to 2000. However there are a number of high risk dates such as 9/9/99, 2/28/00, etc. which must also be considered.

4.2. Date integrity

All manipulations of calendar-related data (dates, durations, days of week, etc.) will produce desired results for all valid date values within the application domain.

4.3. Explicit century

Date elements in interfaces and data storage permit specifying (i.e. specification of the) century, to eliminate date ambiguity. This criterion essentially requires the capability to store explicit values for the century. It must be noted that this must be interpreted as applicable to embedded systems. Not all embedded systems and their component microcontrollers will have this capability.

4.4. Implicit century

For any date element without century, the correct century is unambiguous for all manipulations involving that element. This last criterion requires that if the century is not explicitly provided, its value can be correctly inferred with 100% accuracy from the date provided.

Although the four criteria defined above fully define century compliance, it must be noted that compliance represents a balance between cost and risk rather than an absolute measure. The application of these criteria will vary depending on the system, the criticality to the line of business, the availability of the system for testing and certification, and the test process itself.

5 TESTING STRATEGY

The testing strategy can be divided into several areas:

5.1 Test Parameters.

Based on the compliance criteria defined above, each individual device or system must be studied to determine the characteristics of the device that will certify functionality. It must be noted that not all the functional characteristics of the device need be tested, such as real time functionality or other characteristics that do not have a time related impact.

To illustrate the kind of functionality, from which testbeds can be drawn, testing can be further categorized by functionality as shown below. These examples are not exhaustive of the kind of functionality found in each category. Subject matter experts should be used to determine what date and time related functions need to be tested for a given device or system.

<u>Conversion and Extraction Functionality</u>

The kind of routines to be tested here include such functionality as:

DayOfYear (YYYYMMDD). This kind of routine might be invoked in systems where dates are represented at some point inside a program using the Julian date format. For example, dayOfYear (20000101) should return 1, whereas dayOfYear (20000229) should return 60. Error conditions are candidates for testing here too. For example, dayOfYear (20010229) should *not* return 59, 60, 61 or any other number, as the input is *not* a valid date. Testing for this kind of condition may be difficult, since a fully year 2000-compliant system should *not* allow the system date to be set to an invalid date!

Conversely, routine such as **date (YYYYJJJ)** and **month (YYYYJJJ)** should correctly convert to Gregorian equivalents of Julian dates. For example, date (1999365) should return 31 and month (1999365) should return 12, or 'DEC' or 'DECEMBER,' depending on the system requirements. Similarly, date (2000060) should correspond to February 29th, *not* March 1st. As before, although testing may be difficult, error conditions should be detected and handled; for example, 19990, 1997-10, 2004366, 2020367 etc.

Systems may differ in terms of whether they represent date and time information independently of each other, or compounded into some kind of timestamp structure. (Some systems may use both representations for different purposes.) In the case of compounded, timestamp representations, routines similar to these may be defined over inputs of the form **YYYYJJJhhmmss** or **YYYYMMDDhhmmss**. In real-time systems, representations may typically be defined to greater levels of precision than seconds.

Depending on how a system boundary has been drawn, date and/or date-and-time *formatting* may need to be tested. Date and/or time outputs may appear on terminals, printers, LED and LCD displays, analog meters, digitally simulated analog meters etc. Even if date and/or time data does not display directly, it may be used to derive or calibrate data that is displayed on these types of device, or data that is used for annunciation. These systems should be validated for year 2000 compliance through to the data display portion of the system boundary, particularly if some critical operator intervention might depend on the accuracy of the data.

Arithmetic Functionality

The kind of routines to be tested here include such functionality as:

daysBetween (startDate, endDate). This kind of routine might be invoked on a regular basis by software that calculates inspection, maintenance, replacement schedules etc. or that statistically analyzes raw data. Year 2000 compliance testbeds should test for correctness of cases such as days between (19991231, 20000301), which should calculate 61.

addDays (startDate, numberOfDays). Again, this kind of routine might be relevant in systems where schedules are being set as well as forecasting systems, simulators etc. A testbed might include addDays (1999365, 2), which should return 20002.

subtractDays (startDate, numberOfDays). SubtractDays would be relevant in systems similar to those where addDays might be a part of the system functionality.

The same considerations apply to arithmetic routines as well as conversion and extraction routines when date and time representations are compounded. Of particular importance here is the consideration of correctly interpreting the
time 12:00 as midnight or noon when a 12-hour time representation is used.

Date Comparison Functionality

The kinds of routines to be tested here include standard sorting and searching functionality. This kind of processing represents the majority of date usage in software

sort (list, ascending). Given a list of dates, or time-and-date timestamps, returns a list sorted correctly in ascending or descending order, depending on the second parameter.

LessThan (YYYYMMDD, YYYYMMDD). No sorting routine can exist without complementary comparison routines to support it. Comparison routines are at the heart of the entire year 2000 compliance issue. These routines should be tested thoroughly.

5.2 Test Environment.

This involves preparation of a test environment to test the functional characteristics. This can be further categorized as follows:

Device level testing – Testing of a device in an environment isolated from its normal production environment.
System level testing – Testing of a complete system

It would always be preferable to test a device or complete system *in situ*, i.e. in the normal production environment. This may however not always be possible for various reasons – the system cannot be taken off-line, the time to prepare a test setup in the production environment may be excessive, error recovery may not be possible, etc. Subject matter experts should be consulted for preparation of a valid test environment. Some additional guidelines on preparation of a test environment are as follows:

1. If the test environment is a modified production environment, error recovery procedures must be clearly laid out.
2. All data and software where applicable, must be backed up prior to testing.
3. If a separate test environment is being set up, it must be ensured that all hardware models, revision levels of software etc., are exactly the same as the production environment.
4. All external data interfaces must be isolated so as to avoid any clash or discrepancy with any dates from other systems.

5.3 Control Group testing.

Following the setup of a test environment, testing must be carried out using current dates. This will establish the validity of the test environment.

A different kind of control group testing will need to be carried out for post remediation testing. In this case the modified system should first be tested using current dates to establish that no new errors arise.

5.4 Century testing
Following the successful completion of the control group testing , the system should be tested for century compliance based on the test parameters defined earlier.

6 TESTING PROCEDURES

The guidelines will be used for century testing of devices are defined below. These guidelines are based on the four century compliance criteria defined in Section 4. It must be noted that for each individual system all tests may not apply, and that a checklist should be drawn up based on functionality and the specific application that the system is performing.

6.1 *Definitions*

Century date – Jan 01 2000
Leap Year – Year 1996, 2000, 2016
High-risk dates – 12/31/98, 9/9/99, 12/31/99, 2/28/00,2/29/00,3/01/00

6.2 *Testing Guidelines*

6.2.1 Date setting and Representation.

- System can be set to any date in a range e.g. between 1995 and 2005.
- System can be set to dates both in Julian and Gregorian formats where applicable
- System can be set to high risk dates
- System can be re- initialized from cold start using high risk dates

6.2.2 Date Rollover

- System rolls over correctly on high risk dates
- System rolls over correctly both in powered up and powered down states
- System rolls over correctly both in Gregorian and Julian formats where applicable

6.2.3 Date Arithmetic

- System correctly calculates elapsed dates on either side of century rollover
- System correctly calculates days of the week, based on dates
- System correctly computes leap year dates
- System correctly converts between Julian and Gregorian representations

6.2.4 Date Comparison

- System is able to make correct date comparison e.g. 99 < 00
- System is able to correctly sort date fields on both sides of century.

6.2.5 Date Interface

- System is correctly able to pass date values to external devices and systems
- System is correctly able to maintain date information in the upstream/downstream chain

A MILLENNIUM SURVIVAL GUIDE FOR IT PERSONNEL

OVERVIEW

Failure to resolve Millennium issues will:

- Compromise our commitment to the health and safety of our workers and the public
- Force generating plant shutdowns
- Impair our ability to deliver energy
- Adversely impact how we realize and account for revenue
- Create consequential liabilities

The *Millennium Survival Guide* is a document that provides the Application Developer with an understanding of the Year 2000(Y2K) date problem, and methods to resolve today's non-compliant code problems and methods to prevent non-compliant application development in the future.

The company has over 1500 applications. Each application has to be reviewed and a millennium strategy decision has to be made for each. Is the application acceptable as is? Does the application require coding modifications? Is the application obsolete? Will we replace the application? Does the application require a version upgrade?

A recommended approach is to first read the guide in its entirety. Then, depending on whether you are developing a new application, validating an application for Y2K compliance, converting a non-compliant application, replacing a non-compliant application, or upgrading a non-compliant vendor package, follow the appropriate steps outlined in the *Action List* section.

THE PROBLEM

The Year 2000 problem is easy enough to describe. Most computer systems represent dates in the format MMDDYY, where 12/31/95 represents December 31, 1995. The century is not represented in the date, and we simply assume that 12/31/95 refers to 12/31/1995. Most computer programs that perform arithmetic and logic operations on these date fields use only the last two digits of the year when they make their calculations. As long as all the dates in question are in the same century, this works fine. Problems arise, however, when the century changes. Subtracting 12/31/95 from 12/31/05 to determine someone's age, for example, does not produce the correct answer of 10. It actually produces a result of -90.

Although the problem is easy to describe, it is very difficult to solve for a number of reasons, and can be compared to looking for a needle in a haystack. The visual image of looking through hay is not difficult to conjure up, but the painstaking execution of the solution is awesome. The sheer size of the problem is the first of these. Dates are everywhere, which means that all program code must be examined to determine if a change is necessary. Utilities, like most large corporations, has thousands of programs containing millions of lines of code. A programmer will have to examine each of those lines and make a decision as to whether or not it has to be changed for the Year 2000. A date field can be called date, or it can be called ball game. Many people in the data processing industry, when confronted with the Year 2000 issue, refuse to believe the size or scope of the problem. Many of them argue that changing dates to include a century should be a relatively easy process. This fails to take into account the large number of changes that must be made, as well as, the coordination and testing of those changes. Ownership of the problem is critical to its solution.

DEFINITION OF MILLENNIUM COMPLIANT

The term, "**Millennium Compliant,**" is the quality of a system to provide all of the following functions:

- Handle date information before, during, and after January 1, 2000, including, but not limited to, accepting date input, providing date output, and performing calculations using dates or portions of dates

- Function accurately and without interruption before, during, and after January 1, 2000, without any change in operations associated with the advent of the new century

- Respond to two-digit year date input in a way that resolves the ambiguity as to century in a disclosed, defined, and predetermined manner

- Store and provide output of date information in ways that clearly define century

PURPOSE OF THE MILLENNIUM PROGRAM

The Millennium Program has been put in place to ensure against the unacceptable business consequences of computer systems failing as a direct result of millennium date incompatibility.

The Millennium Program has been put in place to protect and preserve investment in information technology by preventing significant computer system failures that would result from the inability of existing systems to accurately manage dates in the Year 2000 and beyond**.**

The Millennium Program will provide focus and consulting to business units in their efforts to fix non-IT equipment. IT equipment is any equipment that is under the maintenance and support accountability of any professional IT provider in the company or a contractor thereto.

The Millennium Program will provide focus, standards, program management, and resources to the IT community to fix all computer systems which they and the business unit system owners determine will fail as a direct result of millennium date incompatibility resulting in unacceptable business consequences.

The Millennium Program will seek out and require all computer system vendors to certify compliance of their systems in writing to the company, or, in the absence of such certification, will recommend a course of action to the appropriate managers.

The Millennium Program will budget all costs associated with enterprise level initiatives (e.g., awareness campaigns, outsourcing of work), as well as costs to analyze, define, design, test, implement, and verify compliance.

Costs associated with any end user labor resources needed to validate the business functionality of the systems will be budgeted by the business units.

Costs associated with fixing non-IT equipment will be budgeted by the business units.

WHO DO I CALL FOR HELP?

The Millennium Team is here to help. We welcome your questions, comments, suggestions, and ideas. We are all located at XXXX. Here is how to contact us:

	Internet Address	Telephone
Name	xxxxxx	XXXX

YEAR 2000 TESTING

Year 2000 testing requires that the Application Developer develop test cases primarily for input data testing of numerous conditions including leap year, date transaction validation, day/week/month in week/month/year calculations, data integrity, sequencing (i.e., JCL sort parameters, internal program sort), and time-sensitive data. In addition, every user must determine that his/her PC's system clock is Year 2000 compliant. Conditions at the Application Environment and Platform levels must be taken into account, as well.

The Software Millennium Test Development Guidelines Section should assist you in preparing these test cases:

- This section contains testing conditions that application developers must consider in preparing for YEAR 2000 changes.

- It also provides test conditions and associated date values–valid or invalid—for test cases especially applicable to unit testing. This is an ever-changing document and is updated and stored on the Lotus Notes *Millennium Document Library* under "Y2K Software Millennium Test Development Guidelines".

Conversion Methods

The Millennium Team is identifying and categorizing all applications by surveying application owners and compiling an application inventory. Application owners were able to identify their applications as applications that are required to be in service after the Year 2000, applications that are intended to be rewritten or replaced with a vendor package, or applications that are no longer necessary and considered obsolete.

All applications that are deemed required after the Year 2000 can be broadly categorized as either compliant (correctly processes date logic) or non-compliant (incorrectly processes date logic). These applications must be tested for compliance regardless of whether the application was originally categorized as compliant or non-compliant. From a high level perspective the following must occur for each application required to be in service after the Year 2000:

1. Identify the application as needed to be Year 2000 validated or modified to make it Year 2000 Compliant.

2. Develop and run a Year 2000 Test appropriate for the application.

3. Evaluate results. The process is complete if the Year 2000 Test proves that one of the following conditions is true:

   - *The application is compliant.*

   or

   - *The Year 2000 non-compliance is such that we can continue to use the application and do our work without the need for additional work. In other words, if the consequence of non-compliance is acceptable (i.e., a minor problem such as a report date or system that will only be non-compliant for a short period of time) the system may not be converted.*

The process will continue with the remaining steps only if the Year 2000 Test proves that all or a portion of the application needs further work

1. Run a Baseline Test—using an existing (modified, if necessary) or a new Baseline Test—to provide a reference to ensure that the functionality of the application has not changed after the code has been changed. This Baseline Test can be limited to only those portions of the code that have been changed.

2. Direct the programmers to make the Year 2000 coding changes.

3. Rerun both the Baseline Test and the Year 2000 Test to guarantee that there are no functionality changes and that the application is Year 2000 compliant.

The actual conversion (Step 5 above) can take place several different ways. We can convert an application with in-house resources, supplement in-house resources with contractors, or package the application and send it through the Conversion Factory that has been established. The Millennium Team will ensure a smooth conversion.

The Millennium Team has developed several detailed methodology templates for conversions and validations for both mainframe and client server applications. Each methodology addresses different situations by including different steps to follow to complete the conversion/validation. Each template has an associated checklist of detail procedures to follow for each of the steps. The following is a list of the currently developed templates and a brief description.

Detailed Methodology Templates

Mainframe Full Conversion Vendor with Baseline

A mainframe application will be converted by the selected conversion vendor off-site or on COMPANY premises. *(Template #1)*

Mainframe Full Conversion In-house With Baseline

A mainframe application will be converted by company personnel. *(Template #2)*

Mainframe Limited Conversion In-house Without Baseline

A mainframe application will be converted by company personnel. *(Template #3)*

Mainframe Validation With Baseline

A mainframe application will be validated for Year 2000 compliance by company personnel. *(Template #4)*

Mainframe Validation Without Baseline

A mainframe application will be assessed for Year 2000 compliance by the selected conversion vendor. *(Template #5)*

Mainframe Vendor Package Validation with Baseline

A mainframe application will be validated for Year 2000 compliance by company personnel. *(Template #6)*

Mainframe Vendor Package Validation Without Baseline

A mainframe application will be assessed for Year 2000 compliance by the selected conversion vendor. *(Template #7)*

Non-Mainframe Validation

A non-mainframe application will be validated for Year 2000 compliance by company personnel. *(Template #8)*

Non-Mainframe Conversion

A non-mainframe application will be converted by company personnel. (*Template #9)*

These templates can be found in the Lotus Notes *Millennium Document Library* under "Conversion Templates." If you do not have access to the Millennium Document Library or do not have access to Lotus Notes, please contact a representative of the Millennium Team for assistance.

Overview of the Conversion Options

**Once an application has been identified, and a methodology template has been chosen, the application developer must decide upon the specific technique to be used to bring each application program into compliance. The specific techniques include bridging, windowing, or date expansion,
or simply not convert. The following briefly describes each technique:**

Bridging

Bridging is the conversion method of choice if there are more than a few dates within a program. Bridging logic is added at the beginning of each program to expand the year to include the century. Bridging logic is also added at the end of each program to remove the century from the year. Therefore, data files coming in and going out of the program will remain in the same format. The century will be determined by a common program that will be accessed by each program in the application. Century will be determined by the following rule:

- If a year field is greater than 35 (e.g., “36” through “99”), “19” will be assumed to be the century. However, if the year is less than or equal to 35 (e.g., “00” through “35”), “20” will be assumed to be the century. This rule applies only if the application does not use dates prior to 1935.

The previously mentioned bridging process requires substantial setup such as cloning copybooks, creating new modules, and adding program logic containing a series of “Moves.” Thus, the bridging process is not efficient unless there are more than a few dates.

Windowing

Windowing is the conversion method of choice if there are only a few dates within a program. Instead of adding logic at the beginning and end of the program, Windowing logic is added following each date reference in the program. As in Bridging, each date is expanded to include the century. Again, data files coming in and going out of the program will remain in the same format. The century will not be determined by a common program; it will be determined by its own logic. The following rule will be used to determine the century:

- If a year field is greater than 35 (e.g., “36” through “99”), “19” will be assumed to be the century. However, if the year is less than or equal to 35 (e.g., “00” through “35”), “20” will be assumed to be the century. This rule applies only if the application does not use dates prior to 1935.

Date Expansion

Expanding the field in a file or column in a database is another conversion option. With this approach data will be physically expanded to reformat dates to a four digit year or other compliant format (DATE data type). *This is not a recommended method for mainframe applications.*

No Conversion

As a final option, company may choose not to convert an application that is non-compliant. Company may choose to accept a certain level of non-compliance if the consequence of non-compliance is acceptable (i.e., a minor problem exists, such as a report date or system that will only be non-compliant for a short period of time).

DATE STANDARDS RECOMMENDED TO BE YEAR 2000 COMPLIANT

As the need to exchange information across network boundaries increases, lack of common standard practices will become a formidable barrier to interoperability. It was identified that there are a variety of date standards being used within Company's IT complex. Depending on the environment and software/language or coding method, each application seems to maintain its own technique of expressing and storing date data. Many applications maintain different formats for input, output, display, and storage. Some have Julian formats, other have Gregorian formats, and even among those Julian and Gregorian formats, there are differences in representations (e.g., MMDDYY, YYMMDD, YYDDD). Some applications maintain numeric date formats *as binary, display, or packed* and others have alphanumeric representations. Applications sharing different date formats may be subject to additional risk of failure such as DATE data being distributed between different technologies (i.e., DB2, SYBASE) or downline feeds (i.e., Indus to other ad hoc applications).

For date input, report output, and screen displays, the USA standard date is to be utilized at COMPANY. This standard provides consistency for viewing and entering date data. The format of the USA Standard is:

Std	Name	Format	Length	Display
USA	**IBM USA Standard**	**MM/DD/YYYY**	**10**	**01/15/1996**

For data storage, most applications should use the current System date/time data type format supplied by their software. The advantage to this is that numbers representing dates and times can be stored in columns with numeric data types. Applications such as DB2 or SQL Servers (i.e., SYBASE) have the ability to recognize and load date or time values from outside sources, converting valid input values to their internal format. Another advantage is that they can store date and time information from January 1, 1753 through December 31, 9999.

There are some applications that cannot conform to a date data type format (i.e., ADABAS), and, therefore, should default to a character "8" ISO format (listed below). Although some COMPANY applications areas developed ADABAS systems in the past, the use of ADABAS is not the strategic direction for the company. In the future, as the larger ADABAS systems get replaced by packages and the smaller ones are converted into existing client server applications, the inconsistency between the two formats will become less of an issue. The format of the ISO Standard is:

Std	Name	Format	Length	Display
ISO	**International Standards Organization**	**YYYYMMDD**	**8**	**19960115**

ACTION LISTS

For any application developer at COMPANY, surviving the Y2K challenge will mean developing new applications that are Y2K compliant, validating existing applications for Y2K compliance, version upgrading for a non-compliant vendor package, replace with new vendor package or the actual conversion of applications for Y2K compliance. From a high level perspective, the following items should be performed for each unique application challenge.

New application development requires the following:

Understand the problem

Application developers must first understand the Y2K issues outlined in the beginning of the survival guide. Please refer to *Background, Magnitude of Problem,* and *Definition of Millennium Compliant* in the Survival Guide.

Understand Application Conditions

There are many things that an application can do that can cause non-compliance. Please refer to the *Application Conditions* section of the Survival Guide

Follow COMPANY date standards

The Millennium Team has developed display and storage date standards. Please refer to *Date Standards at COMPANY* section of the Survival Guide.

Develop new application using COMPANY date standards

Develop an application following all COMPANY standards and guidelines for new development including COMPANY date standards to ensure Y2K compliance.

Validate compliance using Y2K test plan

Develop a test plan that ensures all application transactions and conditions are Y2K compliant. Please refer to *Year 2000 Test Conditions* section of the Survival Guide or Step 4 Prepare Baseline/Y2K test cases in any methodology template identified in *COMPANY Conversion Method* section of the Survival Guide.

Version Upgrading of existing applications requires the following:

Understanding the problem

Application developers must first understand the Y2K issues outlined in the beginning of the survival guide. Please refer to *Background, Magnitude of Problem,* and *Definition of Millennium Compliant* in the Survival Guide.

Check for new version application conditions that may cause non-compliance

There are many things that an application can do that can cause non-compliance. Please refer to the *Application Conditions* section of the Survival Guide

Validate compliance using Y2K test plan

Develop a test plan that ensures all application transactions and conditions are Y2K compliant. Please refer to *Year 2000 Test Conditions* section of the Survival Guide or Step 4 Prepare Baseline/Y2K test cases in any methodology template identified in *COMPANY Conversion Method* section of the Survival Guide.

Replacing with new vendor applications requires the following:

Understanding the problem

Application developers must first understand the Y2K issues outlined in the beginning of the survival guide. Please refer to *Background, Magnitude of Problem,* and *Definition of Millennium Compliant* in the Survival Guide.

Ensure that the Millennium compliance language is in contract with vendor.

Company requires that all purchased and/or leased products meet date compliance requirements into and beyond the Year 2000 , with no interruption of service or additional expense. Any and all costs including, but not limited to, product upgrades and direct expenses incurred due to failures caused by the change in century, shall be the responsibility of the vendor**.**

If the product is, or will not be designed to meet Year 2000 compliance, the vendor must notify in writing prior to entering into any purchase agreement.

Check for application conditions that may cause non-compliance

There are many things that an application can do that can cause non-compliance. Please to the *Application Conditions* section of the Survival Guide.

Validate compliance using Y2K test plan

Develop a test plan that ensures all application transactions and conditions are Y2K compliant. Please refer to *Year 2000 Test Conditions* section of the Survival Guide or Step 4 Prepare Baseline/Y2K test cases in any methodology template identified in *Conversion Method* section of the Survival Guide.

Converting non-compliant applications requires the following:

Understand the problem

Application developers must first understand the Y2K issues outlined in the beginning of the survival guide. Please refer to *Background, Magnitude of Problem, and Definition of Millennium Compliant* in the Survival Guide.

Check for application conditions that may cause non-compliance

There are many things that an application can do that can cause non-compliance. Please refer to the *Application Conditions* section of the Survival Guide

Follow the recommended standards for conversion

The Millennium Team has identified several methodology templates for conversions and validations for both mainframe and client server applications. Please refer to the *Conversion Method* section of the Survival Guide to select the appropriate methodology template.

Validate compliance using Y2K test plan

Develop a test plan that ensures all application transactions and conditions are Y2K compliant. Please refer to *Year 2000 Test Conditions* section of the Survival Guide or Step 4 Prepare Baseline/Y2K test cases in any methodology template identified in *Conversion Method* section of the Survival Guide.

Y2K Millennium Project

Roles and Responsibilities

High Level Tasks	Y2K Team	IT Staff Managers	Software Owner
Planning			
1. Take ownership of the problem.		X	X
2. Validate for completeness the inventory of all applications.	X	X	X
3. Identify all developed application software.	X	X	X
4. Identify all vendor hardware and software.	X	X	X
5. Assume responsibility for a selected set of applications - Management Staff Responsibility (MSR) List		X	
6. Identify applications that are maintained by IT staff Managers.	X	X	X
7. Identify quality software/applications.	X	X	X
8. Initiate vendor Y2K compliance process.	X		
Scheduling			
1. Choose project conversion option.		X	X
2. Determine whether the work can be done by programming environment, or supplemented by the Y2K team resources. Find resources if staff is not available.	X	X	X
3. Identify/Commit/Coordinate resource to do the validation and/or conversion work.	X	X	X
4. Provide start date and a projected completion date for application to be validated or converted.		X	X
Conversion/Validation			
1. Provide Departmental Instructions for application testing or conversion.	X		
2. Develop a test plan for the applications for which you have responsibility.		X	X
3. Convert/Validate the application.	X	X	X
4. Test application for Y2K compliance.	X	X	X
Sign Off			
1. Sign off on the application indicating that it is obsolete, compliant, or ignored.		X	X

SOFTWARE MILLENNIUM TEST SIGNOFF

Software Title: _____ Revision No. ____

Application (if different from Software Title): _____ CR No._____

Software Owner, Title: _____

Prepared By, Title: _____

Check Appropriate Selection(s):

Millennium Testing Performed		**IF Millennium Testing *not* Performed**
___ Software is Millennium Compliant		___ Vendor certified software is millennium compliant. (Attach copy of vendor certification)
___ Software is ***not*** Millennium Compliant*		___ Software does ***not*** perform date input, output, or processing.
___ Conversion will be performed		___ Software can ***not*** be tested* - (reason is attached)
___ Conversion will ***not*** be performed*		___ Software is retired
		___ Software will be retired prior to date related problems*

*Contingency Plan is required to address actions if software conversion or retirement is ***not*** completed prior to date problems. The contingency plan must be attached to this document.

_____ _____
Software Owner/Computer Owner Date

_____ _____
Supervisor of Software Owner/Computer Owner Date

_____ _____
Manager of Department that Owns Software Date

SOFTWARE MILLENNIUM TEST DEVELOPMENT GUIDELINES

Software Title: _____ Revision No. ____

Application (if different from Software Title): _____

Software Owner, Title: _____

Prepared By, Title: _____

During the process of testing, apply a combination of verification and validation techniques. These techniques include:

1. Unit Testing
- 1.1. Testing the System Clock
- 1.2. Input Testing
- 1.3. Data Testing

2. System Testing
- 2.1. Stress Testing
- 2.2. Recovery Testing
- 2.3. Regression Testing
- 2.4. Error Handling Testing
- 2.5. Manual Support Testing
- 2.6. Parallel Testing

3. Integration Testing
- 3.1. Intra- and Inter-System Testing

4. PC Testing

The following sections will cover some useful testing techniques and scenarios for Year 2000 testing. They are not meant to be all inclusive. Therefore, it is important that additional tests be developed, as appropriate, for the application.

Attention: **By nature, Year 2000 exposures are time-sensitive and time-driven. Be cautious before resetting the system timer. Some system resources and functions are time-sensitive and may be activated or de-activated when the system clock is reset. Such effects can occur when the system clock is either set forward or backward. Without careful planning, you could cause the loss of these system resources and/or functions, some of which might contaminate the production system or production data bases when running various test scenarios.**

Unit Testing

Unit test is performed on one piece of software module at a time and is an exhaustive test of all logic within the module to demonstrate correctness and adherence to applicable specification and design requirements. Unit test should focus on exposing defects within the module logic (try to make it fail).

Testing the System Clock - This test involves resetting the system clock to identify problems which could occur (software, firmware, hardware, system access, etc.) when the century changes.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Expiration Test	Mainframe/Client Server	
			- User IDs		
			- Passwords		
			- Authorization/protection access		
			- Network access		
			- Automation functions		
			- SMS (System Managed Storage)/HSM (Hierarchical Storage Management) migrated data sets earlier than expected		
			2. Label driven tape datasets - are tapes expired earlier than expected? (i.e., validate label parameter expiration (99365, 99366))	Mainframe/Client Server	
			3. Archiving data expired earlier than expected?	Mainframe/Client Server	
			4. (12/31/1999 23:55 hrs) Monitor screen and transaction behavior	Mainframe/Client Server	

Unit Testing-Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			- Validate that dates are calculated and displayed correctly (i.e., 1999 rolls into 2000, ***not*** 1900)		
			5.Validate End of Processing logic to see if dates will be incorrectly interpreted and/or used.	Mainframe/Client Server	

Input Testing - Apply requirements testing to verify that the system performs its function correctly and that it remains functional over a continuous period of time.					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Will program respond correctly if “00” or “2000” is entered.	Mainframe/Client Server	
			2. Is a 4-digit year accepted or is it truncated?.	Mainframe/Client Server	
			3. Ensure xx/xx/xx date =xx/xx/xxxx after expansion or conversion for all databases and tables.		

Data Testing - Set the clock to test process cycles and automatic functions that are activated on a regular basis. These scenarios can be used to identify Year 2000 exposures that need to be fixed as well as to validate programs after applying Year 2000 solutions.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Leap year - Ensure that year 2000 is processed as a leap year.	Mainframe/Client Server	
			- 1996/2/29 should pass (1996 is a leap year)		
			- 2000/2/29 should pass (2000 is a leap year)		
			- 2004/2/29 should pass (2004 is a leap year)		
			2. Invalid Leap Year Test	Mainframe/Client Server	
			- 2/29/1999 non-leap year		
			- 2/29/2001 for non-leap year		
			3. Date Transaction Validation	Mainframe/Client Server	
			- (01/01/2000) Test processing for the first calendar day of the year		
			- (01/31/2000) Test and validate processing for the last business and calendar day of the month		
			4. Day-in-year calculation test	Mainframe/Client Server	
			- Does year 2000 have 366 days (not 365)?		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			5. Day-of-the-week calculation test	Mainframe/Client Server	
			- 02/28/2000 should be a Monday		
			- 03/01/2000 should be a Wednesday		
			- 01/03/2000 First business day of week		
			- 01/03/2000 First business day of month		
			- 01/03/2000 First business day of year		
			- 01/07/2000 Last business day of week		
			6. Week-of-the-year calculation test	Mainframe/Client Server	
			- The 11th week of the year 2000 is 3/5 to 3/11		
			7. End-of-Week Test	Mainframe/Client Server	
			- 01/08/2000 should be a Saturday		
			- 01/09/2000 should be a Sunday		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			8. Data Integrity	Mainframe/Client Server	
			- Are years 1800, 1900, 2000 distinguishable between one another?		
			- Validate for hard coded century occurrence of "19" and/or "20" in program code.		
			- Calculations - Look at programming logic to see if the usage of dates/date ranges in calculations will be correct		
			- Calculations - Check calculation when extends coverage into Year 2000 and verify future billing amounts are not impacted.		
			9. JCL/DCL CONTROL LANGUAGES	Mainframe/Client Server	
			- Ensure sorts use dates properly in processing		
			- Validate and test sort parameters		
			- Review sorts internal to programs		
			- Validate sort data sequence		
			- Record length adjusted - validate that increase records size are reflected in Record Length (LRECL - Logical RECord Length) field.		
			- Validate that Blocksize a multiple of LRECL		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			10. Age Test	Mainframe/Client Server	
			- Use 12/31/1899 to verify age and date of birth calculations		
			- Validate processing for roll-over to 2001		
			11. Time-sensitive data (may ***not*** be applicable to some applications)	Mainframe/Client Server	
			a.) Use current system clock and test data with dates:		
			- Before 01/01/2000		
			- After 01/01/2000		
			b.) Set system clock to 12/31/1999 and test data with dates:		
			- Before 01/01/2000 (12/15/1999) validate transaction calculations are correct within 10, 15, and 30 day period)		
			- After 01/01/2000 validate that everything behaves normally as 2000 approaches		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			c.) Set system clock after 01/01/2000 and test data with dates:		
			- Before 01/01/2000 validate backdated calculations are correct within 10, 15, and 30 day period)		
			- After 01/01/2000 Test - Set system clock to (02/29/2000) validate backdated calculations are correct within 45, 60 and 90 day period		
			- Set system clock to (03/31/2000) validate processing for the last business and calendar day of the quarter		
			- Set system clock to (03/31/2000) validate processing for the last business and calendar day of the quarter		

System Testing

System Testing ensures sufficient testing of a function's implementation and helps determine that all structures of the system are integrated to form a cohesive unit.

Stress Testing - apply stress testing to determine if the system can function when transaction volumes are larger than normally expected. The typical areas that are stressed include disk space, transaction speeds, output generation, computer capacity, and interaction with people. When testing Year 2000 changes, it is essential to verify that the existing resources can handle the normal and abnormal volumes of transactions after the restructuring of the code and the possible expansion of the data fields. For example, apply stress tests to determine:

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Can environment sufficiently accommodate the additional disk space required to support 2 to 4 digit expansion (DASD)?	Mainframe/Client Server	
			2. Are additional CPU cycles required to support code conversion (i.e., 2 digit encoding/ compression scheme) region size?	Mainframe/Client Server	
			3. Is response time adequate to support user turn around time?	Mainframe/Client Server	
			4. Do file definitions need to be reformatted (i.e., CI Splits, Data Dictionaries)?	Mainframe/Client Server	

Recovery Testing - Recovery Testing is used to ensure that the system can restart processing after losing system integrity. This is essential for systems in which the continuity of operation is critical to end users. Recovery processing normally involves the ability to go back to the last checkpoint, then reprocess up to the point of failure.
Can system restart processing after losing system integrity?

Any data integrity or unresolved exposures that lead to inconsistent data or code after you have implemented appropriate Year 2000 solutions will affect the completeness of backup data.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Can application go back to the last check point then reprocess up to the point of failure?	Mainframe/Client Server	
			2. Is documentation complete to support the manual manipulation of data?	Mainframe/Client Server	
			3. Can the system handle unconverted data (bridging available)?	Mainframe/Client Server	
			4. Verify results when a date is entered in one format (e.g. yymmdd/ccyymmdd) and displayed in a different format (e.g. mmddyy/mmddccyy). (2-byte-MF…4-byte-C/S format). Test for Julian dates, especially for calculations and Job Schedule Calendar.	Mainframe/Client Server	

Regression Testing- Ensures that all aspects of a system remain functionally correct after changes have been made to a program in the system. Because the potential exists for a tremendous amount of data and programs to be involved in your Year 2000 transaction, any change to an existing program in the system can have a snowballing or cascading effect on other areas in the system. A change that introduces new data or parameters, or an incorrectly implemented change can cause a problem in previously tested parts of the system, simply because of the way data can be shared between software entities.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Are user requirements followed (i.e., quality assurance)?	Mainframe/Client Server	
			2. Changes meet design specifications?	Mainframe/Client Server	
			3. Changes compliant with organization's policies and procedures?	Mainframe/Client Server	
			4. Validate data output records - data field following date field expansion.	Mainframe/Client Server	
			5. Validate data output records - data field in front of date field expansion.	Mainframe/Client Server	
			6. Validate on-line screen display field for error.	Mainframe/Client Server	
			7. Ensure all scheduling based on date return the same results before and after Y2K changes.	Mainframe/Client Server	
			8. Ensure conditions cover time zone differences.	Mainframe/Client Server	
			9. Ensure all extracting basedate returns the same results before and after Y2K changes.	Mainframe/Client Server	

Regression Testing- Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			10. Ensure all index processing based on date returns the same results before and after Y2K changes.	Mainframe/Client Server	
			11. Ensure all subscripting based on e returns the same results before and after Y2K changes.	Mainframe/Client Server	

Error Handling Testing - Determines if the system can properly process incorrect transactions that can be reasonably expected as types of error conditions. Error-handling testing is necessary to determine the ability of the system to properly process incorrect transactions that can be reasonably expected ass types of error conditions. For example, programs that accept only 4-digit year data entry format need to provide error messages for data entry in 2-digit year format, and vice versa for programs that accept only 2-digit year data entry format. When changing from 2-digit year format to 4-digit year format, you need to apply error-handling testing to verify the appropriate error-handling functions.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Normal error handling for current 4 digit year data entry when 2 digit data entry occurs.	Mainframe/Client Server	
			2. Normal error handling for current 2 digit year data entry when 4 digit data entry occurs.	Mainframe/Client Server	

Manual Support Testing - Evaluate the process by which the end user handles new data generated from the automated applications with Year 2000 support. Types of data from these applications include data entry and report generation. Any new data format should be easy to understand and ***not*** ambiguous. This method includes testing the interfaces (for example screens, procedures, operation manuals, and online HELP panels) between end users and the application

program. End users should be trained and use procedures provided by the system personnel. Testing should be conducted without any other assistance.					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Are new field on on-line screens ambiguous?	Mainframe/Client Server	
			2. Operation manuals updated with new procedures?	Mainframe/Client Server	
			3. On-line HELP panels updated?	Mainframe/Client Server	

Parallel Testing - Determine whether the processing and results of an application's new program version are consistent with old program version. Parallel testing requires that the same input data be run through the two versions of the application. However, if the new application changes data formats, such as reformatting the year-date notation to 4-digit format, you must modify test input data before testing.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Validation output report - Determine if data displays will be acceptable (compliant) in Year 2000	Mainframe/Client Server	
			- Date fields		
			- Non-date fields		
			- Report headers		
			- Report footers		
			2. Validate on-line screens - Determine if data displays will be acceptable (compliant) in Year 2000	Mainframe/Client Server	
			- Date fields		
			- Non-date fields		
			- Screen headers		
			- Screen footers		
			- On-line screen help		
			3. Validate that hard-coded dates or century indicators are ***not*** located in output records.	Mainframe/Client Server	

Parallel Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			4. Validate that hard-coded dates or century indicators are ***not*** located on output reports.	Mainframe/Client Server	
			5. Validate that hard-coded dates or century indicators are ***not*** located on screen displays.	Mainframe/Client Server	
			6. Validate output reports for zero suppression (i.e., year "00" would ***not*** display).	Mainframe/Client Server	
			7. Validate screen displays for zero suppression (i.e., year "00" would ***not*** display).	Mainframe/Client Server	
			8. Verify that the portion of the system that have *no* changes still runs properly as changes are made to other portions of the system.	Mainframe/Client Server	
			9. Verify that the program handles all its transactions correctly and remain stable for a defined period of time.	Mainframe/Client Server	
			10. Ensure that programs (or table subscripts) can handle date ranges that cross Millennium. - Some programs used dates as subscripts. (i.e., "00" for Year 2000 would be an invalid entry, September 1999 "999" my be considered as an end of file marker.)	Mainframe/Client Server	

Integration Testing

Intra- and Inter-System Testing - Applications are frequently connected with other applications to provide a higher or deeper level of functionality. Data may be shared between applications or systems. Intersystem testing is required to ensure that the connection functions properly between the applications. This test determines that the proper parameters and data are correctly passed between applications, and proper coordination and timing of each function exists between applications.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Proper parameters passed between applications?	Mainframe/Client Server	
			2. Data transferred in the proper format to inter-system?	Mainframe/Client Server	
			3. Verify that the system can accept input from, and provide output to, other systems with which it interfaces as interfaces change.	Mainframe/Client Server	

PC Testing

Some older models of the PC may ***not*** have the capability to set or roll over the system clock beyond the year 2000 because the Basic Input/Output System (BIOS) is unaware of the century digit.
(Ex. ROMBIOS in most older PCs can ***not*** handle the year 2000 rollover correctly. Different versions of BIOS behave in different manners: Some roll the date to 1/4/00 or 3/1/00; Some roll the date to 1/1/80 or 3/1/80; Some just leave the date at 12/31/99)

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Test if the system clock can be set beyond the year 2000.	PC	
			- Set the system clock to 01/01/2000, reboot PC and recheck the date.		
			2. Test system clock automatic update function.	PC	
			a.) Test the system clock automatic update function when the power is on. - Set clock to 12/31/1999, 23:58:00, keep power on, validate date when clock reaches the year 2000. - Power off PC and recheck the DOS date.		
			b.) Test the system clock automatic update function when the power is off. - Set system clock to 12/31/1999, 23:58:00, power off PC and wait until the clock reaches the year 2000. - Power on PC and recheck the DOS date.		

PC Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			3. Test time update by the operating system	PC	
			a.) Update After Suspension of a time-sensitive program: - Set system clock to 12/31/1999, 23:58:00; suspend a time display program without a "wake up" timer; keep power on; wait until the clock reaches the year 2000; resume time display program; and check the date.		
			b.) Update After Suspension and Wake Up of time-sensitive program: - Set system clock to 12/31/1999, 23:58:00; suspend a time display program with a "wake up" timer set at 01/01/2000, 00:01:00; keep power on; wait until the time display program "wakes up"; check the date.		
			4. Leap Year Test	PC	
			a.) Change date 02/29/2000. If an error occurs, then BIOS is incorrect.		
			5. Test CPU	PC	
			a.) Use different machines (286/386, etc.) when executing tests to ensure processing time isn't impacted.		

Assessing Computer Software

for

Millennium Compliance

1. OBJECTIVE

This instruction establishes a method for assessing computer software to ensure software will be millennium compliant by the year 2000. The assessment consists of testing, and if required, the effort necessary to bring non-millennium compliant software into compliance (i.e., "conversion"). "Millennium Compliant" is the capability of a system to provide the following functions, if applicable to the system:

NOTE: Date processing is obvious when the date is entered manually. However, the date may be input into software automatically (e.g., the date can be a value from another software program, the software can 'read' a device that provides the date, or it can be calculated using an offset from a pre-established entry.

- Process date information before, during, and after January 1, 2000, including, but ***not*** limited to:
  - ◊ the ability to enter the date,
  - ◊ the ability to output the date, and
  - ◊ the ability to perform calculations on, or using, the date or portions of the date.

- Operate accurately and without interruption before, during, and after January 1, 2000, without any change in operations associated with the advent of the new century.

- Recognize a two-digit year date input (e.g., '98', '99', '00') in a way that resolves the ambiguity as to century.

- Store and provide output of date information in ways that will ***not*** be ambiguous between the centuries 1900 and 2000. For example, many computer programs perform arithmetic and logic operations on their data field and use only two digits of the year instead of four. This presents a problem when the century changes. For example, subtracting 12/31/95 from 12/31/05 to determined someone's age does ***not*** produce the correct answer of 10. It actually produces a result of -90, therefore missing the intent of that calculation.

2. REFERENCES and ATTACHMENTS

References

2.1 DC 11, “Computer System Use and Control”

2.2 DC 14, “Administration of Controlled Software”

2.3 ACP-QA-2.27, “Infrequently Performed Tests and Evaluations”

2.4 ACP-QA-9.03, “Inservice Plant Testing”

2.5 NGP 3.12, “Safety Evaluations”

2.6 NRC Notice Nuclear Safety Engineering Report, “Year 2000 Effect on Computer Systems” (web site: http://www.nrc.gov/NRC/NEWS/in96070.txt)

2.7 Year 2000 Understanding the Problem (Copy available from Computer Services department)

2.8 A Survival Guide For IT Personnel in Application Development (Copy available from Computer Services department)

2.9 NGP/QS-11, “Quality Software Manual (QSM)”

Attachments

Attachment 1.0 - DEFINITIONS AND ACRONYMS

Attachment 2.0 - SOFTWARE MILLENNIUM TESTING

Attachment 3.0 - SOFTWARE MILLENNIUM TEST DEVELOPMENT GUIDELINES

3. PROCEDURE

3.1. Assign Software Owner and Obtain Assistance.

3.1.1. IF ***not*** already assigned, ASSIGN Software Owner in accordance with DC 11, Rev 1, Section 1.2.

NOTE: The Millennium Project Team (MPT) was created to ensure software applications are qualified for operation during and beyond the year 2000. The MPT can be contacted by calling the Administrative Assistant,

3.1.2. IF assistance is required in performing any steps of this instruction, CONSULT the individual who has Management Staff Responsibility (MSR) for the application. CALL, for the name of the MSR for a specific application.

3.2. Document if Software will be retired prior to date related problems or is currently retired.

3.2.1. IF software will be retired prior to any date related problems, PERFORM the following:

a) INDICATE on Attachment 2, “Software Millennium Test Signoff,” that software will be retired prior to date related problems.

b) PROVIDE a contingency plan if the software will not be retired prior to date processing problems, affix plan to Attachment 2.

c) OBTAIN appropriate signatures on Attachment 2.

d) SEND a copy of Attachment 2 with contingency plan to the MSR.

e) IF Quality Software, include Attachment 2 with contingency plan in the Software Document file.

f) IF Controlled Software, MAINTAIN Attachment 2 with contingency plan.

g) IF Quality or Controlled Software, SEND a copy of attachment 2 with contingency plan to Nuclear Document Services.

h) EXIT this procedure.

☐ 3.2.2. IF software is currently retired, PERFORM the following:

a) INDICATE on Attachment 2, “Software Millennium Test Signoff,” that Software is retired.

b) OBTAIN appropriate signatures on Attachment 2.

c) SEND a copy of Attachment 2 to the MSR.

d) IF Quality Software, include Attachment 2 in the Software Document file.

e) IF Controlled Software, MAINTAIN Attachment 2.

f) IF Quality or Controlled Software, SEND a copy of attachment 2 to Nuclear Document Services.

g) EXIT this procedure.

3.3 Determine if Vendor Certification of Compliance is Available

☐ 3.3.1. IF vendor supplied software, CONTACT MSR to determine if MSR has a record that vendor certified the software as millennium compliant.

3.3.2. IF vendor software is millennium compliant, PERFORM the following:

a) INDICATE on Attachment 2, “Software Millennium Test Signoff,” that millennium testing was ***not*** performed as “vendor certified software is millennium compliant.”

b) OBTAIN appropriate signatures on Attachment 2.

c) SEND a copy of Attachment 2 and vendor certification to the MPT.

d) IF Quality Software, INCLUDE Attachment 2 and vendor certification in Software Document File (SDF).

e) IF Controlled Software, MAINTAIN Attachment 2 and vendor certification.

f) IF Quality or Controlled Software, SEND a copy of attachment 2 and vendor certification to Nuclear Document Services.

g) EXIT this procedure.

3.4 Determine if Software Performs Date Input, Output, or Processing

3.4.1. EVALUATE -developed software and vendor software that has ***not*** been certified millennium compliant to determine if date input, output, or processing is performed by software.

3.4.2. IF it has been determined that software does ***not*** perform date input, output, or processing, PERFORM the following:

a) INDICATE on Attachment 2, “Software Millennium Test Signoff,” that millennium testing was ***not*** performed as software does ***not*** perform date input, output, or processing.

b) OBTAIN appropriate signatures on Attachment 2.

c) SEND a copy of Attachment 2 to the MSR.

d) IF Quality Software, INCLUDE Attachment 2 in the Software Document file.

e) IF Controlled Software, MAINTAIN Attachment 2.

f) IF Quality or Controlled Software, SEND a copy of attachment 2 to Nuclear Document Services.

g) EXIT this procedure.

3.5. Document if Software Can *not* be Tested and Millennium Compliance Can *not* be Determined

3.5.1. IF software can ***not*** be tested AND it can ***not*** be determined that date processing is performed by software, PERFORM the following:

a) INDICATE on Attachment 2, “Software Millennium Test Signoff,” that software can ***not*** be tested.

b) PROVIDE a reason testing can ***not*** be performed with Attachment 2.

c) Refer to RP4 (Corrective Action Program) and initiate CR (Condition Report).

d) If Quality Software refer to QS-11, “Error Reporting and Corrective Action for Quality Software”, and perform any additional error reporting activities.

e) PROVIDE a contingency plan if software fails due to date processing problems, affix plan to Attachment 2.

f) OBTAIN appropriate signatures on Attachment 2.

g) SEND a copy of Attachment 2 with contingency plan to the MSR.

h) IF Quality Software, include Attachment 2 with contingency plan in the Software Document file.

i) IF Controlled Software, MAINTAIN Attachment 2 with contingency plan.

j) IF Quality or Controlled Software, SEND a copy of attachment 2 with contingency plan to Nuclear Document Services.

k) EXIT this procedure.

3.6. Develop and Perform Millennium Test, Document Results, and Perform Conversion, if Applicable

NOTE: Attachment 3, “Software Millennium Test Guidelines,” lists different types of date processing performed by software. This attachment will be used to help determine appropriate date processing tests for the software being evaluated.

☐ 3.6.1. Refer To Attachment 3, “Software Millennium Test Development Guidelines,” determine and check-off which software tests are applicable to the software being evaluated.

NOTE: Testing on some plant systems require special test procedures to be developed, reviewed and approved.

3.6.2. Refer to the following, as applicable for software, and DEVELOP a Software Millennium Test:

- Attachment 3, “Software Millennium Test Development Guidelines”
- IF applicable, ACP-QA-9.03, “Inservice Plant Testing”
- IF applicable, ACP-QA-2.27, “Infrequently Performed Tests or Evolution’s”

3.6.3. IF required by DC-12, Refer To NGP 3.12, “Safety Evaluation” and PERFORM the following:

a) ENSURE a 10CFR50.59 Safety Evaluation Screening on the test has been performed by a qualified safety evaluation screener.

b) IF required by the 10CFR50.59 Safety Evaluation Screening, ENSURE a 10CFR50.59 Safety Evaluation on the test is performed by a qualified safety evaluator.

NOTE: Testing on many systems (e.g., plant system) will require use of AWO (Automated Work Order). Ensure proper approval of AWO prior to start of work.

3.6.4. Refer To Applicable work process and PERFORM Software Millennium Test.

3.6.5. INDICATE results of software testing on Attachment 2, “Software Millennium Test Signoff.”

3.6.6. IF software is ***not*** millennium compliant:

a) INDICATE on Attachment 2 if conversion will be performed.

b) INCLUDE contingency plan if conversion ***not*** performed or ***not*** complete prior to date impact.

c) Refer to RP4 (Corrective Action Program) and initiate CR (Condition Report) to indicate the software is not millennium compliant.

d) If Quality Software refer to QS-11, “Error Reporting and Corrective Action for Quality Software”, and perform any additional error reporting activities.

3.6.7. OBTAIN appropriate approvals on Attachment 2.

3.6.8. SEND a copy of completed and approved Attachment 2 to MSR.

3.6.9. PERFORM the following:

- IF software is Quality Software, MAINTAIN Software Millennium testing documentation in Software Document file.
- IF software is Controlled Software, MAINTAIN millennium test documentation.
- IF Quality or Controlled Software, SEND a copy of Millennium Test Documentation to Nuclear Document Services.

NOTE: 1. Effort to bring non-millennium compliant software into compliance (i.e., "conversion") may take different methods which depend upon the change required and the type of software (e.g., business application versus plant system). Some examples of possible conversion methods are software upgrades from a vendor, in-house code modifications, purchase of replacement software, or total system replacement. Use of the Design Control Process, prescribed in the DCM, may be required based upon the system impacted.

2. Replacement of software, systems containing software, vendor upgrades, and in-house software modifications all require the use of DC 11, "Computer System Use and Control."

3.6.10. IF testing results indicate millennium conversion is required AND conversion is desired, PERFORM the following:

a) IF applicable, Refer to "Design Control Manual", and ENSURE appropriate corrective actions are performed.

b) Refer To DC 11, "Computer System Use and Control" and PERFORM activities necessary to acquire, modify, upgrade, or develop software to satisfy millennium conversion.

c) DETERMINE if step 3.1 applies. If it applies, PERFORM steps 3.1.1 through 3.1.2. to ASSIGN Software owner for new/converted software.

d) DETERMINE if step 3.3 applies. If it applies, PERFORM steps 3.3.1 through 3.3.2. to determine if vendor certification of compliance is available for new/converted software.

e) DETERMINE if step 3.4 applies. If it applies, PERFORM steps 3.4.1 through 3.4.2.

f) EXIT procedure.

.

Attachment 1

4. DEFINITIONS AND ACRONYMS

(Page 1 of 5)

Acceptance Testing - A test of the entire software program with data for production readiness.

Assign - To transfer or appoint to another resource/individual

AWO - Automated Work Order

Bridging - A method used to convert data to an acceptable format, external to the program logic.

Buffer - An area in memory in which data is stored temporarily to facilitate output or processing later.

Client Server - A network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are less powerful PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.

Computer- Programmable electronic device that can store, retrieve, and process data.

Computer Owner - The individual responsible for maintenance and operations of a computer system. The Computer Owner may be the Software Owner.

Contingency Plan - A formal document which contains an alternate course of action to be implemented when original plans can not be met. For example, the original application may have to be converted if the replacement project fails to meet Year 2000.

Controlled Software - See DC 11, “Software and Data Classification,” to determine if software is controlled. Software that is ***not*** identified as Quality Software and is any of the following:

- Important to plant operation
- That whose erroneous output could impact plant operations
- For all Plant Process Computers, all software ***not*** identified as Quality Software is considered Controlled Software

CPU - Central Processing Unit - The central processor of the computer that controls the processing routines, performs arithmetic functions, and maintains memory.

Attachment 1
DEFINITIONS AND ACRONYMS
(Page 2 of 5)

DASD - Direct Access Storage Device (i.e. magnetic disk drives)

DATA - Information of any type, including binary data, hexadecimal numbers, integers, character strings, ASCII characters, etc.

DCB - Data Control Block - Properties that set dataset configuration.

DCL - Digital Control language, used by Digital Equipment Corporation (DEC)

DISK - A magnetic disk used to store information.

Error - A departure from the validated function of the Quality Software.

Firmware - Software contained on non-volatile media, such as Programmable Read-Only Memories (PROMs) and erasable PROMs (EPROMs).

Functional Testing - Functional testing is designed to ensure that the system and end-user requirements and specifications are achieved. Functional testing focuses on the results of processing rather than how processing is implemented. To accomplish this, create test cases to evaluate the functional correctness of the system and programs.

HSM - Hierarchical Storage Management - a group of software components that transparently manages files between magnetic disk or some other storage device.

Integration Testing - A test of a related group of program modules.

Interface - An exchange of information between one device and another or the device that makes such exchanges possible.

JCL - Job Control language, used by International Business Machines (IBM).

LRECL - Logical Record Length.

Mainframe Systems - Hardware and Software associated with centralized computer systems. Included are the following:

- Time Sharing Option (TSO)
- Customer Information Control Systems (CICS)
- Conversational Monitor System (CMS)

Media - Material on which data my be stored, such as magnetic tape, paper, or disks.

Modify - To change or alter.

MPT - Millennium Project Team

Attachment 1
DEFINITIONS AND ACRONYMS
(Page 3 of 5)

MS-DOS - Microsoft Disk Operating System

MSR - Management Staff Responsibility, individual who has overall responsibility for millennium compliance for a particular software application.

Operating System (OS) - Software that controls program execution, resource allocation, scheduling, scheduling, input/output control and data management.

Peripheral - A device controlled by the processor that is external to the Computer. Some peripheral devices include video display, disk drives, and printers.

Personal Computer (PC) - Hardware and Software associated with single-user Microprocessor-based computer.

Plant Process Computer - Any real-time sensor-based monitoring or control computer system that assists nuclear unit operation. Included are the following:

- Systems traditionally known as a “unit Plant Process Computer”
- Other plant process computers, such as special - purpose computers, mini-computers, microprocessor computers, programmable logic controllers, programmable logic devices, application specific integrated circuits, etc. based instrumentation monitoring and process control systems
- Station security computer system

Quality Software - Software whose output is used in Quality applications. Refer to DC 11, “Computer System Use and Control” for lists of Utility’s Quality software. Quality applications, as a minimum, include:

- The design process associated with Category I structures, systems, or components.
- Support of Technical Specifications related to category I structures, systems, components, or design-basis analyses.
- Verification of compliance with Technical Specifications related to design basis analyses, when used as the sole or principle means of verification.
- Support of plant licensing with respect to Category I structures, systems, components, or design-basis analysis.
- Implementation of a safety function of a Category I system.
- Implementation of 10CFR50 Appendix B requirements.

Attachment 1
DEFINITIONS AND ACRONYMS
(Page 4 of 5)

Retired Software - Applications no longer in service (obsolete).

ROMBIOS - Read Only Memory Basic Input Output System - a collection of routines (usually stored in ROM) that control items such as the video display, disk drives, and keyboard.

Special Use Workstation - Desktop device, other than a PC, used for a single specific function. Examples include CAD and technical procedure publishing workstations.

Software - Sequence of instructions suitable for processing by a computer. Examples of software includes database applications, volatile electronic programs and non-volatile electronic programs, such as those stored in Programmable Read-Only Memories (PROMs), i.e. Firmware.

Software Document File - The file that provides or points to documentation and history of Quality Software.

Software Implementation Package - The name for the collection of all the required documentation specific to the installation of the new of modified software. The Software Implementation Package contains, as a minimum, all the documentation indicated as required on the implementation package check list. Some departments use the Software Document File itself as their Software Implementation Package. This is an acceptable substitution. For more information see DC-11.

Software Millennium Test - Test used to demonstrate compliance of software with millennium test cases developed using Attachment 3, “Software Millennium Test Development Guidelines.”

Software Millennium Test Report - Document that contains results of Software Millennium Test.

Software Owner - employee responsible for specific software. For Quality Software, the individual must be qualified in accordance with NGP 2.26, “Departmental Training,” for the purpose of preparation and performance of procedures, design packages, or validation and verification tests. The Software Owner may employ others to perform software-related tasks, but retains overall responsibility.

SMS - System Managed Storage - An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies.

Attachment 1
Definitions and Acronyms
(Page 5 of 5)

Stress Testing - A test to determine if the system can function when transaction volumes are larger than normally expected.

System Testing - A test of the integration and cohesiveness of the application.

Unit Testing - A test of a single program module.

Validation - Process that evaluates functional characteristics of Software, and certifies achievement of acceptable comparisons with Objective Evidence.

Validation Test - A test that assesses functionality of Software to the extent that Validation is accomplished.

Verification - Process that confirms that the performance of Quality Software is unchanged from that demonstrated by Validation, or that Database quality information is accurate.

Verification Test - Test that confirms the performance of Software is unchanged from that demonstrated by validation or test that confirms database quality information is correct.

Attachment 2
SOFTWARE MILLENNIUM TEST SIGN-OFF

Software Title: _____ Revision No. ____

Application (if different from Software Title): _____ AR Number:_____

Software Owner, Title, Phone #: _____

Prepared By, Title, Phone #: _____

Check Appropriate Selection(s):

Millennium Testing Performed		**IF Millennium Testing *not* Performed**
___ Software is Millennium Compliant		___ Vendor certified software is millennium compliant. (Attach copy of vendor certification)
___ Software is ***not*** Millennium Compliant*		___ Software does ***not*** perform date input, output, or processing.
___ Conversion will be performed, AR under RP4 is required		___ Software can ***not*** be tested* - (reason is attached)
___ Conversion will ***not*** be performed*		___ Software is retired
		___ Software will be retired prior to date related problems*

*Contingency Plan is required to address actions if software conversion or retirement is ***not*** completed prior to date problems. The contingency plan must be attached to this document.

_____ _____
Software Owner/Computer Owner Date

_____ _____
Supervisor of Software Owner/Computer Owner Date

_____ _____
Manager of Department that Owns Software Date

Attachment 3
SOFTWARE MILLENNIUM TEST DEVELOPMENT GUIDELINES

Software Title: _____ Revision No. ____

AR No. _____

Application (if different from Software Title): _____

Software Owner, Title, Phone #: _____

Prepared By, Title, Phone #: _____

During the process of testing, apply a combination of verification and validation techniques. These techniques include:

1. Unit Testing
- 1.1. Testing the System Clock
- 1.2. Input Testing
- 1.3. Data Testing

2. System Testing
- 2.1. Stress Testing
- 2.2. Recovery Testing
- 2.3. Regression Testing
- 2.4. Error Handling Testing
- 2.5. Manual Support Testing
- 2.6. Parallel Testing

3. Integration Testing
- 3.1. Intra- and Inter-System Testing

4. PC Testing

5. Your Own Tests

The following sections will cover some useful testing techniques and scenarios for Year 2000 testing. They are not meant to be all inclusive. Therefore, it is important that additional tests be tailored, as appropriate, for the application.

Attention: **By nature, Year 2000 exposures are time-sensitive and time-driven. Be cautious before resetting the system timer. Some system resources and functions are time-sensitive and may be activated or de-activated when the system clock is reset. Such effects can occur when the system clock is either set forward or backward. Without careful planning, you could cause the loss of these system resources and/or functions, some of which might contaminate the production system or production data bases when running various test scenarios.**

Unit Testing

Unit test is performed on one program at a time and is an exhaustive test of all logic within the program to demonstrate correctness and adherence to applicable specification and design requirements. Unit test should focus on exposing defects within the module logic (try to make it fail).

Testing the System Clock - This test involves resetting the system clock to identify problems which could occur (software, firmware, hardware, system access, etc.) when the century changes.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Expiration Test	Mainframe/Client Server	
			- User IDs		
			- Passwords		
			- Authorization/protection access		
			- Network access		
			- Automation functions		
			- SMS (System Managed Storage)/HSM (Hierarchical Storage Management) migrated data sets earlier than expected		
			2. Label driven tape datasets - are tapes expired earlier than expected? (i.e., validate label parameter expiration (99365, 99366))	Mainframe/Client Server	
			3. Archiving data expired earlier than expected?	Mainframe/Client Server	
			4. (12/31/1999 23:55 hrs) Monitor screen and transaction behavior	Mainframe/Client Server	

Unit Testing-Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			- Validate that dates are calculated and displayed correctly (i.e., 1999 rolls into 2000, ***not*** 1900)		
			5.Validate End of Processing logic to see if dates will be incorrectly interpreted and/or used.	Mainframe/Client Server	

Input Testing - Apply requirements testing to verify that the system performs its function correctly and that it remains functional over a continuous period of time.					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Will program respond correctly if “00” or “2000” is entered.	Mainframe/Client Server	
			2. Is a 4-digit year accepted or is it truncated?.	Mainframe/Client Server	
			3. Ensure xx/xx/xx date =xx/xx/xxxx after expansion or conversion for all databases and tables.		

Data Testing - Set the clock to test process cycles and automatic functions that are activated on a regular basis. These scenarios can be used to identify Year 2000 exposures that need to be fixed as well as to validate programs after applying Year 2000 solutions.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Leap year - Ensure that year 2000 is processed as a leap year.	Mainframe/Client Server	
			- 1996/2/29 should pass (1996 is a leap year)		
			- 2000/2/29 should pass (2000 is a leap year)		
			- 2004/2/29 should pass (2004 is a leap year)		
			2. Invalid Leap Year Test	Mainframe/Client Server	
			- 2/29/1999 non-leap year		
			- 2/29/2001 for non-leap year		
			3. Date Transaction Validation	Mainframe/Client Server	
			- (01/01/2000) Test processing for the first calendar day of the year		
			- (01/31/2000) Test and validate processing for the last business and calendar day of the month		
			4. Day-in-year calculation test	Mainframe/Client Server	
			- Does year 2000 have 366 days (not 365)?		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			5. Day-of-the-week calculation test	Mainframe/Client Server	
			- 02/28/2000 should be a Monday		
			- 03/01/2000 should be a Wednesday		
			- 01/03/2000 First business day of week		
			- 01/03/2000 First business day of month		
			- 01/03/2000 First business day of year		
			- 01/07/2000 Last business day of week		
			6. Week-of-the-year calculation test	Mainframe/Client Server	
			- The 11th week of the year 2000 is 3/5 to 3/11		
			7. End-of-Week Test	Mainframe/Client Server	
			- 01/08/2000 should be a Saturday		
			- 01/09/2000 should be a Sunday		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			8. Data Integrity	Mainframe/Client Server	
			- Are years 1800, 1900, 2000 distinguishable between one another?		
			- Validate for hard coded century occurrence of “19” and/or “20” in program code.		
			- Calculations - Look at programming logic to see if the usage of dates/date ranges in calculations will be correct		
			- Calculations - Check calculation when extends coverage into Year 2000 and verify future billing amounts are not impacted.		
			9. JCL/DCL CONTROL LANGUAGES	Mainframe/Client Server	
			- Ensure sorts use dates properly in processing		
			- Validate and test sort parameters		
			- Review sorts internal to programs		
			- Validate sort data sequence		
			- Record length adjusted - validate that increase records size are reflected in Record Length (LRECL - Logical RECord Length) field.		
			- Validate that Blocksize a multiple of LRECL		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			10. Age Test	Mainframe/Client Server	
			- Use 12/31/1899 to verify age and date of birth calculations		
			- Validate processing for roll-over to 2001		
			11. Time-sensitive data (may ***not*** be applicable to some applications)	Mainframe/Client Server	
			a.) Use current system clock and test data with dates:		
			- Before 01/01/2000		
			- After 01/01/2000		
			b.) Set system clock to 12/31/1999 and test data with dates:		
			- Before 01/01/2000 (12/15/1999) validate transaction calculations are correct within 10, 15, and 30 day period)		
			- After 01/01/2000 validate that everything behaves normally as 2000 approaches		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			c.) Set system clock after 01/01/2000 and test data with dates:		
			- Before 01/01/2000 validate backdated calculations are correct within 10, 15, and 30 day period)		
			- After 01/01/2000 Test - Set system clock to (02/29/2000) validate backdated calculations are correct within 45, 60 and 90 day period		
			- Set system clock to (03/31/2000) validate processing for the last business and calendar day of the quarter		
			- Set system clock to (03/31/2000) validate processing for the last business and calendar day of the quarter		

System Testing

System Testing ensures sufficient testing of a function's implementation and helps determine that all structures of the system are integrated to form a cohesive unit.

Stress Testing - apply stress testing to determine if the system can function when transaction volumes are larger than normally expected. The typical areas that are stressed include disk space, transaction speeds, output generation, computer capacity, and interaction with people. When testing Year 2000 changes, it is essential to verify that the existing resources can handle the normal and abnormal volumes of transactions after the restructuring of the code and the possible expansion of the data fields. For example, apply stress tests to determine:

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Can environment sufficiently accommodate the additional disk space required to support 2 to 4 digit expansion (DASD)?	Mainframe/Client Server	
			2. Are additional CPU cycles required to support code conversion (i.e., 2 digit encoding/ compression scheme) region size?	Mainframe/Client Server	
			3. Is response time adequate to support user turn around time?	Mainframe/Client Server	
			4. Do file definitions need to be reformatted (i.e., CI Splits, Data Dictionaries)?	Mainframe/Client Server	

Recovery Testing - Recovery Testing is used to ensure that the system can restart processing after losing system integrity. This is essential for systems in which the continuity of operation is critical to end users. Recovery processing normally involves the ability to go back to the last checkpoint, then reprocess up to the point of failure.
Can system restart processing after losing system integrity?

Any data integrity or unresolved exposures that lead to inconsistent data or code after you have implemented appropriate Year 2000 solutions will affect the completeness of backup data.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Can application go back to the last check point then reprocess up to the point of failure?	Mainframe/Client Server	
			2. Is documentation complete to support the manual manipulation of data?	Mainframe/Client Server	
			3. Can the system handle unconverted data (bridging available)?	Mainframe/Client Server	
			4. Verify results when a date is entered in one format (e.g. yymmdd/ccyymmdd) and displayed in a different format (e.g. mmddyy/mmddccyy). (2-byte-MF…4-byte-C/S format). Test for Julian dates, especially for calculations and Job Schedule Calendar.	Mainframe/Client Server	

Regression Testing- Ensures that all aspects of a system remain functionally correct after changes have been made to a program in the system. Because the potential exists for a tremendous amount of data and programs to be involved in your Year 2000 transaction, any change to an existing program in the system can have a snowballing or cascading effect on other areas in the system. A change that introduces new data or parameters, or an incorrectly implemented change can cause a problem in previously tested parts of the system, simply because of the way data can be shared between software entities.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Are user requirements followed (i.e., quality assurance)?	Mainframe/Client Server	
			2. Changes meet design specifications?	Mainframe/Client Server	
			3. Changes compliant with organization's policies and procedures?	Mainframe/Client Server	
			4. Validate data output records - data field following date field expansion.	Mainframe/Client Server	
			5. Validate data output records - data field in front of date field expansion.	Mainframe/Client Server	
			6. Validate on-line screen display field for error.	Mainframe/Client Server	
			7. Ensure all scheduling based on date return the same results before and after Y2K changes.	Mainframe/Client Server	
			8. Ensure conditions cover time zone differences.	Mainframe/Client Server	
			9. Ensure all extracting basedate returns the same results before and after Y2K changes.	Mainframe/Client Server	

Regression Testing- Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			10. Ensure all index processing based on date returns the same results before and after Y2K changes.	Mainframe/Client Server	
			11. Ensure all subscripting based on e returns the same results before and after Y2K changes.	Mainframe/Client Server	

Error Handling Testing - Determines if the system can properly process incorrect transactions that can be reasonably expected as types of error conditions. Error-handling testing is necessary to determine the ability of the system to properly process incorrect transactions that can be reasonably expected ass types of error conditions. For example, programs that accept only 4-digit year data entry format need to provide error messages for data entry in 2-digit year format, and vice versa for programs that accept only 2-digit year data entry format. When changing from 2-digit year format to 4-digit year format, you need to apply error-handling testing to verify the appropriate error-handling functions.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Normal error handling for current 4 digit year data entry when 2 digit data entry occurs.	Mainframe/Client Server	
			2. Normal error handling for current 2 digit year data entry when 4 digit data entry occurs.	Mainframe/Client Server	

Manual Support Testing - Evaluate the process by which the end user handles new data generated from the automated applications with Year 2000 support. Types of data from these applications include data entry and report generation. Any new data format should be easy to understand and ***not*** ambiguous. This method includes testing the interfaces (for example screens, procedures, operation manuals, and online HELP panels) between end users and the application program. End users should be trained and use procedures provided by the system personnel. Testing should be conducted without any other assistance.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Are new field on on-line screens ambiguous?	Mainframe/Client Server	
			2. Operation manuals updated with new procedures?	Mainframe/Client Server	
			3. On-line HELP panels updated?	Mainframe/Client Server	

Parallel Testing - Determine whether the processing and results of an application's new program version are consistent with old program version. Parallel testing requires that the same input data be run through the two versions of the application. However, if the new application changes data formats, such as reformatting the year-date notation to 4-digit format, you must modify test input data before testing.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Validation output report - Determine if data displays will be acceptable (compliant) in Year 2000	Mainframe/Client Server	
			- Date fields		
			- Non-date fields		
			- Report headers		
			- Report footers		
			2. Validate on-line screens - Determine if data displays will be acceptable (compliant) in Year 2000	Mainframe/Client Server	
			- Date fields		
			- Non-date fields		
			- Screen headers		
			- Screen footers		
			- On-line screen help		
			3. Validate that hard-coded dates or century indicators are ***not*** located in output records.	Mainframe/Client Server	

Parallel Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			4. Validate that hard-coded dates or century indicators are ***not*** located on output reports.	Mainframe/Client Server	
			5. Validate that hard-coded dates or century indicators are ***not*** located on screen displays.	Mainframe/Client Server	
			6. Validate output reports for zero suppression (i.e., year "00" would ***not*** display).	Mainframe/Client Server	
			7. Validate screen displays for zero suppression (i.e., year "00" would ***not*** display).	Mainframe/Client Server	
			8. Verify that the portion of the system that have *no* changes still runs properly as changes are made to other portions of the system.	Mainframe/Client Server	
			9. Verify that the program handles all its transactions correctly and remain stable for a defined period of time.	Mainframe/Client Server	
			10. Ensure that programs (or table subscripts) can handle date ranges that cross Millennium. - Some programs used dates as subscripts. (i.e., "00" for Year 2000 would be an invalid entry, September 1999 "999" my be considered as an end of file marker.)	Mainframe/Client Server	

Integration Testing

Intra- and Inter-System Testing - Applications are frequently connected with other applications to provide a higher or deeper level of functionality. Data may be shared between applications or systems. Intersystem testing is required to ensure that the connection functions properly between the applications. This test determines that the proper parameters and data are correctly passed between applications, and proper coordination and timing of each function exists between applications.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Proper parameters passed between applications?	Mainframe/Client Server	
			2. Data transferred in the proper format to inter-system?	Mainframe/Client Server	
			3. Verify that the system can accept input from, and provide output to, other systems with which it interfaces as interfaces change.	Mainframe/Client Server	

PC Testing

Some older models of the PC may ***not*** have the capability to set or roll over the system clock beyond the year 2000 because the Basic Input/Output System (BIOS) is unaware of the century digit.
(Ex. ROMBIOS in most older PCs can ***not*** handle the year 2000 rollover correctly. Different versions of BIOS behave in different manners: Some roll the date to 1/4/00 or 3/1/00; Some roll the date to 1/1/80 or 3/1/80; Some just leave the date at 12/31/99)

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Test if the system clock can be set beyond the year 2000.	PC	
			- Set the system clock to 01/01/2000, reboot PC and recheck the date.		
			2. Test system clock automatic update function.	PC	
			a.) Test the system clock automatic update function when the power is on. - Set clock to 12/31/1999, 23:58:00, keep power on, validate date when clock reaches the year 2000. - Power off PC and recheck the date.		
			b.) Test the system clock automatic update function when the power is off. - Set system clock to 12/31/1999, 23:58:00, power off PC and wait until the clock reaches the year 2000. - Power on PC and recheck the date.		

PC Testing - Continued

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			3. Test time update by the operating system	PC	
			a.) Update After Suspension of a time-sensitive program: - Set system clock to 12/31/1999, 23:58:00; suspend a time display program without a "wake up" timer; keep power on; wait until the clock reaches the year 2000; resume time display program; and check the date.		
			b.) Update After Suspension and Wake Up of time-sensitive program: - Set system clock to 12/31/1999, 23:58:00; suspend a time display program with a "wake up" timer set at 01/01/2000, 00:01:00; keep power on; wait until the time display program "wakes up"; check the date.		
			4. Leap Year Test	PC	
			a.) Change date 02/29/2000. If an error occurs, then BIOS is incorrect.		
			5. Test CPU	PC	
			a.) Use different machines (286/386, etc.) when executing tests to ensure processing time isn't impacted.		

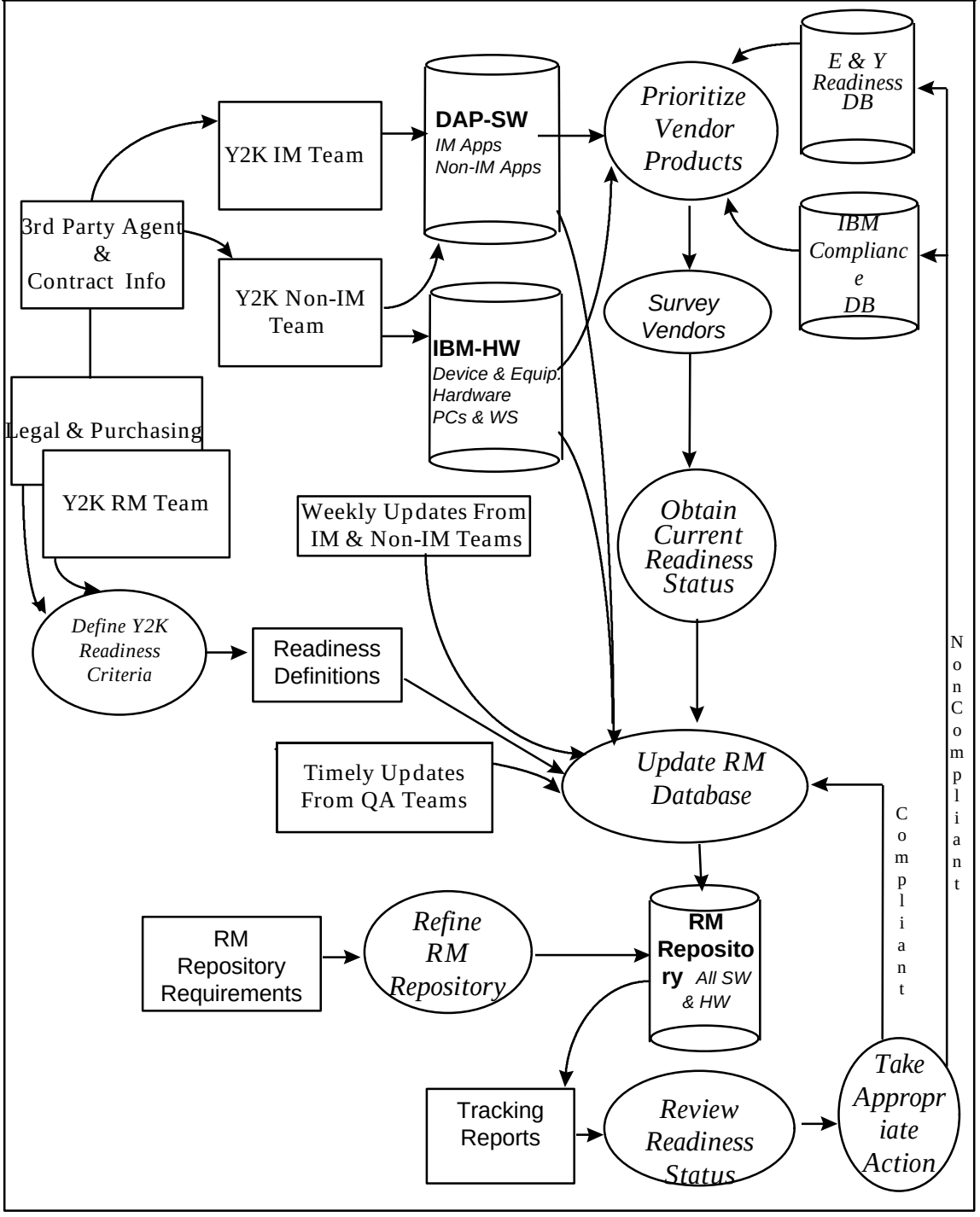
Use This Section To Document Your Own Tests					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments

NUCLEAR UTILITY YEAR 2000 READINESS

Appendix G

READINESS TRACKING PROCESS

Readiness Management (RM) Process



NUCLEAR UTILITY YEAR 2000 READINESS

Appendix H

COMPLIANCE CHECKLIST

YEAR 2000 COMPLIANCE CHECKLIST

This checklist helps application owners, application managers and the Year 2000 Program team evaluate Year 2000 compliance of an application. The checklist should be jointly reviewed and completed by both business subject matter experts and technical team members who are responsible for support of the application. Please answer all questions as thoroughly as possible. Include any documents that will help in the evaluation process, such as requirement definition, test plans, test results, etc. The answers will determine if an application is compliant.

After the Year 2000 Compliance Checklist has been completed, the application business unit owner, the application maintenance support, and the Year 2000 Program QA/QC Manager will review the checklist results. If the application is found to be Year 2000 compliant, sign-off by both the application business unit owner and the application maintenance support group will be required. If the application is found not to be in compliance, then the application business unit owner and the application maintenance support group will have two options:

1. have your application support group bring it into compliance

or

2. turnover the application to the Year 2000 Program Team to bring the application into compliance.

If the option is to have the application support personnel bring the application into compliance, all Year 2000 Program standards must be followed. The Year 2000 Program Team must be included in the setting up of timelines, deliverables, and certification process. If the option is to turnover the application to the Year 2000 Program Team for certification, the Year 2000 Program Team will take complete responsibility for bringing the application into compliance.

1. Application Identification

Please provide application information.

A. Application Name _____

B. Business Unit Owner of the Application _____

C. Sponsoring Department of the Application (VP Org.) _____

D. Application Subject Matter Expert Name _____

E. Application Technical Expert Name _____

F. Is the application in operation today? _____

Additional Comments: _____

_____

_____

_____

_____

_____

_____

2. Year 2000 Dates

Applications work with dates that are weeks, months, and years into the future, or may reference dates in the past. For example, inventory applications may need to process data that spans from 1950 to the present and need to keep its records for at least 50 years. Please verify your application's ability to successfully process data containing dates, with no adverse effect on the application's functionality and with no impact on the customer or end user. Can your application successfully process:

		VERIFIED	NO	N/A
a.	Dates in 20th century (1900s)	_____	_____	_____
b.	Dates in 21st century (2000s)	_____	_____	_____
c.	Dates across century boundary (mix 1900s and 2000s)	_____	_____	_____
d.	Crosses 1999 to 2000 successfully	_____	_____	_____

	YES	NO
Are test data sets available for regression testing on the next application release for any of the above?	_____	_____
Are test results and reports available for review for any of the above?	_____	_____

Additional Comments: _____

_____

_____

_____

_____

_____

_____

3. Other/Indirect Date Usage

Have you verified date handling process (and corrected if necessary):

		VERIFIED	NO	N/A
a.	Dates embedded as parts of other fields			
b.	Dates used as part of a sort key			
c.	Usage of values in date fields for special purposes that are not dates (for example, using 9999, 0000, 99 or 00 to mean "never expire")			
d.	Date dependent activation or deactivation of passwords, accounts, rates, etc.			
e.	Date representation in the operating system's file system (creation dates and modification dates of files and directories)			
f.	Date dependent utilities			
g.	Date dependencies in encryption/decryption algorithms			
h.	Date dependent random number generators			
i.	Hardware and/or operating system does not reset the year to 1980 or 1984 on reboots after 31 December 1999 *(corrections by operating system utilities allowed)*			

	YES	NO
Are test data sets available for regression testing on the next application release for any of the above?		
Are test results and reports available for review for any of the above?		

Additional Comments:

4. Internal Dates

Dates and date fields must be clear and explicit within the applications which use them.

		VERIFIED	NO	N/A
a.	Display of dates is clear and explicit (the ability to correctly determine to which century a date belongs either by explicit display, i.e. 4-digit year, or application or user inference, such as applications that only process and maintain year-to-date data)			
b.	Printing of dates is clear and specific, such as dates in report headings			
c.	Input of dates is clear and distinct to the application using them			
d.	Storage of dates is clear to the application that uses them.			
e.	Date compares and date manipulations within the application are processed correctly.			

	YES	NO
Are test data sets available for regression testing on the next application release for any of the above?		
Are test results and reports available for review for any of the above?		

Additional Comments:

5. External Interfaces

External interfaces are identified and validated to correctly function for all dates passed from your application.

		VERIFIED	NO	N/A
a.	Verified that interfacing application functions the same when the data passed to that interface is generated from your application (for example, an interface is two-digit year and another is four-digit year).			
b.	For each interface that exchanges date data, you and the responsible organization have discussed and verified that you have implemented consistent Year 2000 corrections that will correctly process date data passed between your applications.			

	YES	NO
Are test data sets available for regression testing on the next application release for any of the above?		
Are test results and reports available for review for any of the above?		

Additional Comments:

6. Date Field Type

Describe the type of date fields used by the application, in either application software or data bases.

		VERIFIED	NO	N/A
a.	Does the application use two-digit year data fields?			
b.	Does the application use four-digit year data fields?			
d.	**When will the windowing logic fix fail?** _____			

		YES	NO
e.	If two-digit, does the application use a windowing logic technique to correctly infer the century?		
	If yes, what windowing date ranges does it use: From _____ To _____		
f.	Are there any internal data types for date? Such as character or variable character?		
	If yes, what is the range of dates that the date field can represent? Minimum Date_____ Maximum Date_____		

If character type date, what process does the application use to convert the date data?

_____

_____

_____

	YES	NO
Are test data sets available for regression testing on the next application release for any of the above?		
Are test results and reports available for review for any of the above?		

Additional Comments:_____

_____

_____

_____

7. Vendor Provided Software

Please provide the following information with regard to "Vendor Provided" software components.

		YES	NO	N/A
a.	Does the application use vendor provided software packages or infrastructure components?			
	If yes, what is the software's name?			
b.	Has the vendor provided software been verified to be year 2000 compliant?			
c.	How was Year 2000 compliance determined? (certified by vendor or contractor, tested in-house, etc.)			

Additional Comments:

8. Year 2000 Testing Information

Please provide the following information with regard to testing the application for Year 2000 compliance:

a. Testing Organization _____

b. Name of QA/QC Manager _____

c. Date that Year 2000 compliance testing was completed _____

d. How was Year 2000 compliance determined? (certified by vendor or contractor, tested in-house, inspected but not tested, etc.) _____

	YES	NO
e. Do you follow a defined process for tracking the status of all Year 2000 problems reported, changes made, testing done, compliance verified, and applications returned to production?	_____	_____

Additional Comments: _____

9. Summary of Results

Your application is Year 2000 compliant if any of the following statements are true. Please mark as appropriate.

You completed a full independent testing of the application and you answered all the questions with a positive response (except for either 7a or 7b). _____

An independent audit of your application was completed and you answered all questions with a positive response (except for either 7a or 7b). _____

Your application was not tested or audited but, your application uses only **four-digit century** date fields and you answered all questions with a positive response except for 7a. _____

Your application is **NOT** Year 2000 compliant if any of the following statements are true. Please mark as appropriate.

Your application was **not** tested or audited and, your application uses only two-digit century fields. You answered all questions with positive responses except for 7b. _____

Your application was **not** tested or audited and, your application has ambiguous usage of dates. Questions 5-a,b,c or d (Internal Dates section) were answered with negative responses. _____

Your application was **not** tested or audited and your application needs additional work before Year 2000 processing can be assured with any level of reliability. If any of the sets of questions, 2, 3, 4, 5, or 7b were answered with negative responses. _____

Your application cannot be certified or has not yet been certified as compliant. _____

9. Year 2000 Compliance Sign-off

After review of *application name*, the undersigned certify that *application name* is Year 2000 Compliant. Attached is a listing of all certified programs associated with this application.

Sign-off Information

_____	_____
Business Application Owner (Manager)	Date
_____	_____
Application Support (Manager)	Date
_____	_____
Year 2000 QA/QC Manager	Date
_____	_____
Internal Auditor Manager	Date

Year 2000 Compliance
Certification Checklist, Non-IS Supported

Instructions: A checklist must be completed for each version of each application, equipment or system before it can be certified for continued production use. Fill out Section 1, and if the equipment or system is digitally-controlled or otherwise operates from firmware, fill out Section 2. When completed, return this checklist to the Y2K Coordinator.

The checklist will then be used to prioritize and schedule actual Y2k Compliance Testing per Section 3. This testing may be performed by the user, or by the NMIS Y2k Team. When Compliance Testing is completed, this checklist MUST be signed by the Key User Contact Supervisor or a representative of the NMIS Y2k Team and returned to the Y2K Coordinator.

This information will be reviewed by the Year 2000 QA Team and you will be notified when your application has completed the certification process. If you have any questions or comments, please add this information at the bottom of page 5.

Section 1

Site: _____ Dept/Wkgrp: _____

Application, Equipment or System Name: _____

Application Function: _____

Version: _____ Vendor: _____

& Location(s) of Other Licensed Copies: _____

Hardware Platform: _____

Operating System/File Type: _____

Key User Contact: _____ Ext: _____

Key User Supervisor: _____ Ext: _____

Outline strategy for implementing compliance (i.e., warranty upgrade, purchase upgrade, migrate to different application, date roll-back, windowing, field expansion):

Cat I			
Cat II			
Cat III			
For NMIS Use Only	Cat III	Cat II	Cat I

Please check the appropriate response.

Yes No N/A

☐ ☐ ☐ 1. Is this Version of the application or system the current Production Version?

Skip to Question 5.

Please check the appropriate response.

Yes	No	N/A	
☐	☐	☐	2. Is the Software License for application or system renewed periodically? Specify Period and Vendor: _____
☐	☐	☐	3. The application or system: Is, of itself, Nuclear Safety-Related or NSSS Provides Direct Control of Nuclear Safety-Related/NSSS Items Is Capable of Forcing Immediate or Near-immediate Plant Shutdown Is used for Nuclear Safety-Related Activities/Calculations Provides Automatic Control of Critical Plant Functions If Inoperative, Directly/Indirectly Leads to LCO's of 48 hrs or Less Is used to Protect the Health and Safety of the General Public
☐	☐	☐	4. The application or system: Is used to Protect the Health and Safety of Plant Personnel Provides Control of Plant Habitability Systems If Inoperative, Directly/Indirectly Leads to LCO's > 48 hrs Is used for Control/Tracking of Other Critical Plant Information/Operations (Specify:)
☐	☐	☐	5. The application or system: Provides Direct Control of Other Plant Systems Is used for Control/Tracking of Other Plant Information/Operations Is NOT the current Production Version
☐	☐	☐	6. The application or system: Contains Date/time Stamped Data Is Used for long-term Averaging, Integrating, Trending, Scheduling, or Reporting
☐	☐	☐	7. Is the Application or System Used for short-term Averaging, Integrating, Trending, Scheduling, or Reporting?
☐	☐	☐	8. Is the Application or System Used for Time-Independent Calculations/Operations?
☐	☐	☐	9. Does this Application or System interface with other applications? Specify Send or Receive and App/System: _____

Section 2 For every piece of equipment or system that is a PLC, digitally-controlled instrument or M&TE, or otherwise operates from Firmware, complete Section 2. Otherwise STOP, and return this checklist to the Y2K Coordinator.

Equipment or System Type: _____ MFR: _____

_____

Equipment or System Serial #: _____

Model #: _____ Asset Tag #: _____

Detailed System Location: _____

CPU Mfr/Type: _____ Date Code: _____

& Type of ROM/PROM/EPROM's: _____

Date Code(s): _____

Firmware Version Installed: _____ Firmware Vendor: _____

Vendor's Current Firmware Version: _____

Source Code Version: _____

Please check the appropriate response.

Yes	No	N/A	
☐	☐	☐	10. Does the Equipment or System have an EPN or EID number? Specify: _____
☐	☐	☐	11. Is the Equipment or System Part of, Installed on, or Interface to a system having an EPN or EID number? Specify: _____
☐	☐	☐	12. Is the Equipment or System under Warranty?
☐	☐	☐	13. Does the Equipment or System have a Maintenance Contract? Specify Vendor: _____
☐	☐	☐	14. Does the Equipment or System Operating History, Vendor Technical Manual, Restart Procedure, or Maintenance or Calibration Procedure indicate any form of Date Input or Date Check? Specify: _____
☐	☐	☐	15. Does the Equipment or System Operating History, Vendor Technical Manual, or Maintenance or Calibration Procedure indicate that Batteries are used for Retention of Default or Setup Information? Specify: _____

Please check the appropriate response.

Yes	No	N/A	
☐	☐	☐	16. Does the Equipment or System have a Data or an Event Historian?
☐	☐	☐	17. Does the Equipment or System Perform Trending?
☐	☐	☐	18. Does the Equipment or System Perform Time-dependent Calculations, such as Averaging or Integration?
☐	☐	☐	19. Does the Equipment or System Print reports that include the date?

Describe the nature or use of the Historian, Trend, Calculation, or Report, including any Tech Spec, Regulatory, or Station Commitments that it is used to fulfill. _____

_____

_____

Section 3

Please check the appropriate response.

Yes	No	N/A	
☐	☐	☐	20. Does the application use four digits (YYYY) to represent the year?
☐	☐	☐	If it does not, can the century be logically determined and dates correctly processed?
☐	☐	☐	21. Does the application perform date duration calculations? This includes the following calculations: a) the duration between two dates b) the date based on starting date and duration c) the day of week, day within year, week within year
☐	☐	☐	22. Will the application properly process decisions that require comparisons of dates from before and after the year 2000?
☐	☐	☐	23. The application has been tested with the following date data and can successfully roll over to the next date: a) 09/09/1999 - could be set to mark end of file b) 12/31/1999 - ability to roll over to year 2000 c) 01/01/2000 - Saturday (In 1900, this is a Monday) d) 01/02/2000 - Sunday e) 01/03/2000 - Monday (The 1st workday of year)

f) 02/28/2000 - 2000 is a leap year (Monday)
g) 02/29/2000 - Tuesday (Leap Day)
h) 03/01/2000 - Wednesday
I) 04/01/2000 - Saturday
j) 12/31/2000 - ability to roll over to year 2001
k) 01/01/2001 - Monday, first day of year

☐ ☐ ☐ 24. The application can successfully convert between date representations (YYMMDD to Julian).

☐ ☐ ☐ 25. If date/time date is stored as an offset since a base date/time, the storage capacity has been checked so that it will work correctly through the 21 century.
Indicate Storage Cap'y End Date _____

☐ ☐ ☐ 26. Does the application use special date values as logical flags? (for example, "99" to mean "no end date" or "00" to mean "does not apply")

☐ ☐ ☐ 27. Do reports print correctly? Specifically, reports do not contain any hard coded literals such as '19' for the century.

☐ ☐ ☐ 28. Do screens contain four digit years or can the correct century be inferred? NOTE: Screens should not contain any hard coded literals such as '19' for the century.

☐ ☐ ☐ 29. Will the application correctly sort by date when the dates are from both before and after the year 2000?

☐ ☐ ☐ 30. Has the key function or calculation been tested? Have the results been verified with the appropriate technical support group?

Testing Performed By: _____ Date: _____

Key User Supervisor Signature: _____ Date: _____

Title: _____

Additional Comments: