

YEAR 2000 TESTING FOR APPLICATION SOFTWARE

Scope — Testing Strategy

Version 1.2



Expert Software Services Pty Ltd

A.C.N. 054 141 365

2nd Floor • 326 William Street
Melbourne Victoria 3000 • Australia

WWW: <http://www.expert.com.au>

Email: info@expert.com.au

Telephone: +61 3 9320 6100

Facsimile: +61 3 9320 6111

Document Authorisation

Document Title:	Year 2000 Compliance Testing for Applications Scope — Testing Strategy
Document ID:	Y2K Testing.doc
Document Version:	1.0
Version Date:	23 September 1998
Print Date:	7 October 2025
Distribution:	Lloyd Borrett, Glenn Charlesworth, Gary Ebeyan, Clients

	Name	Signature	Date
Prepared by	Gary Ebeyan		7 Sep 1998
Reviewed by	Glenn Charlesworth		7 Sep 1998
Authorised by:	Gary Ebeyan		7 Sep 1998

Copyright © 1998 Expert Software Services Pty. Ltd. A.C.N. 054 141 365

All Rights Reserved

Document Revision History

Version	Revision Date	Revised By	Description
1.0	10 Feb 1998	Gary Ebeyan	Original Version
1.1	24 Apr 1998	Gary Ebeyan	First Update
1.2	7 Sep 1998	Lloyd Borrett	Makeover and Update
1.3	21 Sep 1998	John Campbell	Add and Merge Revisions

CONTENTS

1. EXECUTIVE SUMMARY	6
2. GETTING IT RIGHT FOR YEAR 2000	7
2.1 Expert Software Services' Year 2000 Compliance	7
2.2 Getting IT Right for Year 2000 Programme	7
3. SCOPE	8
3.1 Problem Statement	8
3.2 Proposed Solution	8
3.3 Definition — Year 2000 Compliance	8
3.4 Limitations	9
3.5 Constraints	9
3.6 Assumptions	9
3.7 Key Milestones	10
4. TEST STRATEGY	11
4.1 Objectives	11
4.2 Test Plan Preparation	11
4.2.1 Overall Strategy	11
4.2.2 Testing Steps.....	11
4.2.3 User Interfaces	12
4.2.4 System Interfaces	13
4.2.5 Programs using Current System Date	13
4.2.6 Reports.....	14
4.2.7 Business Use Cases.....	14
5. TEST ASSESSMENT CRITERIA	15
5.1 "Bug" Classification	15
5.2 Business Impact	16
6. TEST PLAN TEMPLATES	17

System Test Form	18
Test Environment Build Procedure	19
User Interface Testing Form	21
System Interface Testing Form	22
Program Testing Form	23
Report Testing Form	24
Business Case Testing Form	25
Testing Sign Off	25
Business Case 1 — <Name>	26
Test Failure Assessment Form	27

1. EXECUTIVE SUMMARY

Expert Software Services recognises that Year 2000 date change issues are a matter of considerable importance for our clients, especially in regard to the integrity of their ongoing operations. We are applying our best endeavours to ensure that the Year 2000 transition causes minimal disruption for clients using applications provided and supported by Expert Software Services.

Expert Software Services has established a “Getting IT Right for Year 2000” programme, the aim of which is to scope and define the required Year 2000 changes within a client’s environment and to conduct a controlled programme of implementation to achieve a smooth transition.

This programme includes all aspects of the Year 2000 date change for clients. The emphasis is on preserving operational integrity and business continuity. We hope to assist our clients with their Year 2000 transition requirements by using our best efforts to provide:

- ☐ New applications that are Year 2000 compliant; and
- ☐ Services to assist clients:
 - ☐ Conduct inventory and risk analysis of applications and processes;
 - ☐ Conduct remedial activity and testing programmes; and
 - ☐ Develop contingency plans.

This document describes the necessary tests that a software application needs to pass to ensure that it is ‘Year 2000 Compliant’¹. This document does not deal with the process of risk analysis, system correction or change management.

¹ See ‘Year 2000 Compliance’ definition later in this document.

2. GETTING IT RIGHT FOR YEAR 2000

As the Year 2000 approaches, all businesses need to take action to address the transition to the year 2000. Technology in the form of business applications, processes, products, hardware and software must be able to function correctly before, during and after 1st January 2000.

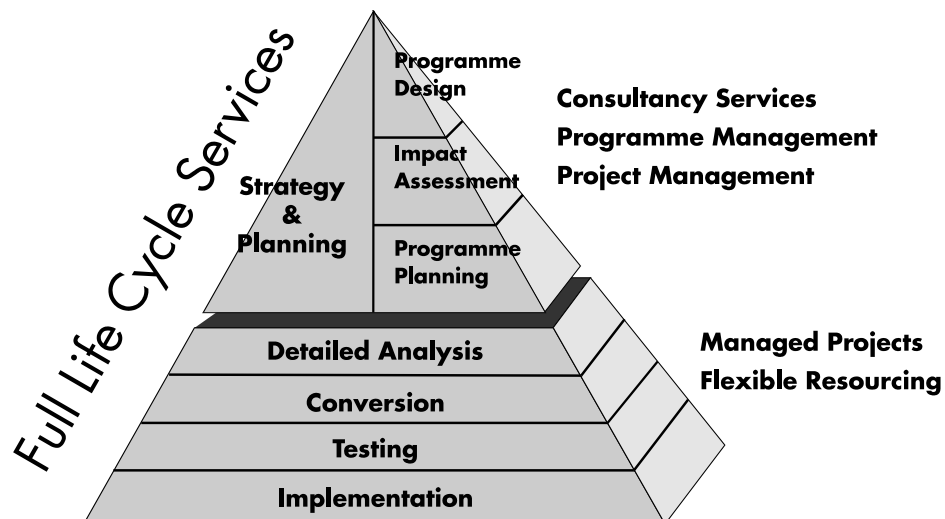
2.1 Expert Software Services' Year 2000 Compliance

Expert Software Services warrants that the code we produce and supply to our clients has been Year 2000 Compliance tested as per PD2000-1, the British Standards Institute definition of Year 2000 conformity. This conformance testing is incorporated in our test plans for all projects. If the client prefers, we are prepared to consider testing to a Year 2000 Conformance definition supplied by them.

Expert Software Services takes no responsibility for the compliance of the third-party software tools, development tools, database systems, operating systems etc. that are used as part of the development or operational platform or damages caused thereof. If requested, we will fully test the solutions we build for Year 2000 compliance and provide a comprehensive compliance report.

2.2 Getting IT Right for Year 2000 Program

Expert Software Services' "Getting IT Right for Year 2000" program is designed to offer a step by step approach to the understanding and remediation of the Year 2000 problem. These steps involve qualifying the impact, agreeing to the best way forward for you and your business, and implementing the agreed solution efficiently and cost-effectively.



3. SCOPE

3.1 Problem Statement

The Year 2000 problem poses one of the most significant challenges ever faced by business. The requirement for corrective action arises because some computer chip dependent equipment and software stores or processes dates with only two digits representing a calendar year (e.g. DD/MM/YY instead of DD/MM/YYYY).

The use of two digit recording was originally intended to save valuable storage space and data entry time. It has the effect that some computers are unable to distinguish years as either 19XX or 20XX. As a result, many forms of technology may either stop working or produce erroneous results when faced with the Year 2000 (i.e. "00") by interpreting it as 1900 or something else.

Some equipment or software may also fail to recognise the Year 2000 as a Leap Year or having 29 days in February (if the century is divisible by 400 then it is a leap year).

3.2 Proposed Solution

In order to certify software applications as Year 2000 Compliant, certain tests must be carried out. A test plan must be created for each software application. The test plan must minimally meet the test criteria set out in this document. The test criteria itself has been created to meet the conditions of the Year 2000 (Y2K) Compliance Definition (see below). Upon the successful completion of all test cases, the system may be considered to be Year 2000 Compliant.

3.3 Definition — Year 2000 Compliance

The following definition is extracted from the Expert Software Services Millennium Date Standards document.

The following compliance definition is to be applied to all applications. In the event that it is found to be impractical or unfeasible for a specific application to meet this Standard, a Concession or Waiver of all or part of the Standard may be authorised by the Compliance Certification Committee.

To be classed as Year 2000 Compliant by Expert Software Services, each application must:

- ☐ *Not use date based logic flags (e.g.: 00 or 99 in year fields) to indicate meanings other than a year.*
- ☐ *Not fail when the year changes to 1999.*
- ☐ *Not fail when the year changes to 2000.*

- ☐ *Treat the year 2000 as the year immediately following the year 1999.*
- ☐ *Recognise that the year 2000 is a leap year.*
- ☐ *Treat dates and times in the twenty-first century in the same way it treats dates and times in the twentieth century.*
- ☐ *Calculate date and time ranges correctly in the twenty-first century, the twentieth century, or spanning the twentieth and twenty-first centuries.*
- ☐ *Sort all date based data correctly in the twenty-first century, the twentieth century, or spanning the twentieth and twenty-first centuries.*
- ☐ *Accept dates input into the system so that their century is unambiguous and correct.*
- ☐ *Display and print dates in an unambiguous way, such that the century is clearly understood.*
- ☐ *Exchange data with other entities or systems with fully expanded 4 digit year fields.*

This compliance definition is derived from PD2000-1, the British Standards Institution definition of Year 2000 conformity.

In addition, business acceptance of the Y2K test results must be obtained before the Y2K compliance can be given. This is particularly important if the specific application has additional data or date ranges requiring testing.

3.4 Limitations

The Y2K Compliance is limited to the Y2K Compliance defined in this document. As some applications may use specific date requirements, the test strategy in this document may need to be modified to meet those specific needs.

3.5 Constraints

In order to allow for sufficient Y2K Compliance testing of software applications, a test plan must be developed for each application using the test strategy defined in this document.

3.6 Assumptions

It is assumed that Clients will determine that the necessary infrastructure required for the software application is Y2K Compliant, including but not limited to the Hardware, Operating System and Database Management System.

3.7 Key Milestones

Each system critical to business continuity must be tested, modified and retested before dates past the year 1999 are used in that system, in a timeframe defined by the millennium team.

4. TEST STRATEGY

4.1 Objectives

- ☐ Test each application system for Y2K Compliance.
- ☐ Identify all cases where Y2K related 'Key Dates' have not been successfully processed.
- ☐ Provide an audit trail of test history.
- ☐ Provide an audit trail of test completeness.

4.2 Test Plan Preparation

4.2.1 Overall Strategy

The overall system testing will need to be broken down into the following 'Test Classifications':

1. User Interfaces.
2. System Interfaces.
3. Programs.
4. Reports.
5. Business Use Cases.

There are templates for each classification in the Appendices to assist in the test plan preparation. An overall System Test form (see Section 6, Test Plan Templates) needs to be prepared for each system.

4.2.2 Testing Steps

A three tier testing process will be carried out.

1. A unit test of each application in a dedicated test environment.
2. A system test of each application and its interface in a dedicated test environment.
3. Staged roll out testing into the production environment.
- .

4.2.3 User Interfaces

Identify all User Interface fields used for date input and/or output.

1. Test each date field with at least the following 'Key Dates':
 - ☐ 01/06/1998 Ensure 20th century dates are still accepted.
 - ☐ 01/01/1999 Ensure that 99 is not treated as end-of-file.
 - ☐ 09/09/1999 Trap special conditions.
 - ☐ 31/12/1999 Highest date value in the 20th century.
 - ☐ 01/01/2000 Ensure that the year 2000 can be input/output.
 - ☐ 04/01/2000 First working day in 2000.
 - ☐ 05/01/2000 First working day change in 2000.
 - ☐ 08/01/2000 First end of week in 2000.
 - ☐ 10/10/2000 Ensure that a 6 digit date field can be input/output.
 - ☐ 01/02/2000 First end of month in 2000.
 - ☐ 29/02/2000 Ensure that year 2000 is a leap year.
 - ☐ 01/04/2000 First quarter end in 2000.
 - ☐ 01/07/2000 First end of financial year for ANZ.
 - ☐ 01/01/2001 First end of financial year.
 - ☐ 29/02/2001 Reject this date to ensure year 2001 is not a leap year.
 - ☐ 29/02/2004 Ensure that the leap year date can be correctly processed.
2. Ensure that all of the above dates are stored correctly in the data store.
3. Ensure that all of the above dates are correctly retrieved and displayed.
4. Ensure that dates are correctly sorted in grids (where appropriate) across centuries.

4.2.4 System Interfaces

4 Digit Years

Identify all System Interfaces with date inputs and/or outputs using 4 digit years.

1. Test each date field with the following 'Key Dates':
 - ☐ 01/01/1999 Ensure that 99 is not treated as end-of-file.
 - ☐ 01/01/2000 Ensure that the year 2000 can be input/output.
 - ☐ 10/10/2000 Ensure that a 6 digit date field can be input/output.
 - ☐ 29/02/2000 Ensure that year 2000 is a leap year.
 - ☐ 29/02/2001 Ensure that year 2001 is not a leap year. Reject this date.
 - ☐ 29/02/2004 Ensure that the leap year date can be correctly processed.
 - ☐ 01/01/0000 Ensure that interfaces with Y2K errors are trapped.
 - ☐ 31/12/0000 Ensure that interfaces with Y2K errors are trapped.
 - ☐ 01/01/1900 Ensure that interfaces with Y2K errors are trapped.
 - ☐ 31/12/1909 Ensure that interfaces with Y2K errors are trapped.
2. Ensure that all of the above dates are correctly stored and retrieved in the data store.
3. Ensure that invalid and inconsistent dates are trapped and rejected.

2 Digit Years

Identify all System Interfaces with date inputs and/or outputs using 2 digit years.

1. Ensure that dates prior to 01/01/50, are treated as 21st century dates.
2. Ensure that dates on and after 01/01/50, are treated as 20th century dates.
3. Ensure that all of the 'Key Dates' above are tested once the correct century has been assigned.

4.2.5 Programs using Current System Date

Identify all Programs using current system date.

1. Test each program with the following system dates:
 - ☐ 01/01/1999 Ensure that 99 is not treated as end-of-file.
 - ☐ 01/01/2000 Ensure that the year 2000 can be processed.
 - ☐ 10/10/2000 Ensure that a 6-digit date field can be processed.
 - ☐ 29/02/2000 Ensure that year 2000 is a leap year.

and ensure that all of the above dates are correctly processed.

4.2.6 Reports

Identify all Reports using date fields.

1. Test each report with at least the following dates in the data store:
 - ☐ 01/01/1999 Ensure that 99 is not treated as end-of-file.
 - ☐ 01/01/2000 Ensure that the year 2000 can be unambiguously output.
 - ☐ 10/10/2000 Ensure that a 6 digit date field can be unambiguously output.
 - ☐ 29/02/2000 Ensure that the leap year date can be unambiguously output.
2. Ensure that all of the above dates are correctly reported.
3. Ensure correct sorting of dates where applicable.

4.2.7 Business Use Cases

Identify all Business Cases using date calculations (eg. recurring orders, due date calculations, etc.).

1. For each Business Case, use the following dates in calculations:

Start Date	End Date	Condition
=< 31/12/1999	>=1/1/2000	Ensure correct date processing.
>=1/1/2000	>=1/1/2000	Ensure correct date processing.
>=1/1/1998	<=31/12/1999	Ensure correct date processing.
2. Ensure that all of the above dates are correctly processed.

5. TEST ASSESSMENT CRITERIA

Any errors and inconsistencies found during the testing phase need to be categorised and evaluated to determine their potential impact.

Impact should be measured both by **technical impact** (the ability of the application to continue operating) and by **business impact** (application's ability to continue to support business processes).

5.1 "Bug" Classification

The following measure provides a means of assessing the severity of different "bug" classifications from a technical viewpoint.

ID	Category	Technical Impact
1.	Ambiguous Display of Date Information	Mild
2.	Ambiguous Printing of Date Information	Mild
3.	Incorrect Display of Date Information	Serious
4.	Incorrect Printing of Date Information	Serious
5.	Use of Date Formats for Special Meanings	Serious
6.	Ambiguous Date Storage Formats	Serious
7.	Unexpected Behaviour	Serious
8.	Ambiguous and incorrect date input	Serious
9.	Inability to store and recognise dates	Critical
10.	Incorrect or invalid date processing including date manipulations and calculations	Critical
11.	Incorrect determination of leap years	Critical
12.	Incorrect or invalid date transmission	Critical
13.	Corruption, loss or unintentional deletion of data	Catastrophic
14.	System shutdown	Catastrophic

Table 1: "Bug" Classifications with technical impact.

5.2 Business Impact

The impact of a “bug” needs to be assessed from a business process viewpoint to determine the potential impact the “bug” may have on an Application’s ability to perform the required business functionality and support the necessary business process.

The severity of the business impact may differ from the severity of the technical impact. For example, the technical impact of printing ambiguous dates is mild but the business impact would be critical for an application that prints dates on cheques.

The following measure may be used to assess the business impact of a bug:

High	Application cannot perform necessary business functionality. Business processes cannot be supported.
Medium	Business processes and functionality are impacted but manual workarounds are known and are available and in use.
Low	Does not impede business processes or functionality.

Table 2: Business Impact Ratings

6. TEST PLAN TEMPLATES

The following templates are presented to assist in the preparation of the test plan:

- ☐ System Test Template
- ☐ Test Environment Build Procedure
- ☐ User Interface Testing Template
- ☐ System Interfaces Testing Template
- ☐ Program Testing Template
- ☐ Report Testing Template
- ☐ Business Case Testing Template
- ☐ Test Failure Assessment Template

System Test Form

System/Application :	
IT Owner:	
Business Owner:	
Test Machine:	
Estimated Days:	

Importance:	
IT Service Provider:	
Business Unit:	
Test Database:	

External Interfaces

Service Provider

Network Requirements

Operational Requirements

Business Involvement

Test Environment Build Procedure

Procedure for Creating Acceptance Data

Type of data load (Binary or raw data)

Procedure for creation of test database or raw files

Script files for building test database (Name, Location and Function)

Procedure for Creating Test Application Structure

Location of Test Application Structure

Steps used to create Test Application Structure

User Configuration on Server

Logon Server requirements

Not Applicable

Database Setup Procedure

UserID and Password to be used to administer database instance

What needs to be copied to server

Database Login Script

Database Setup Script

Database Schema Script

Database Load Script

Permission Setting Script

Server Processing Configuration

UserID and Password to be used to administer environment

What files needs to be copied to server

Server Configuration to run process

Setting up Tasks

COE PC Configuration

List of dependencies required

Installation Script or Procedure (if available)

ODBC settings

Other Settings

Application Setup Issues

General Comments

(Include any special testing constraints and conditions)

IT Approval of the Test Plan

Approved by:	
Signature:	

Date:	
-------	--

Business Approval of the Test Plan

Approved by:	
Signature:	

Date:	
-------	--

IT Acceptance of the Test Results

Accepted by:	
Signature:	

Date:	
-------	--

Business Acceptance of the Test Results

Accepted by:	
Signature:	

Date:	
-------	--

User Interface Testing Form

Estimated Duration: _____ hours/days

Form Id	Field	Pass	Fail

Prepared by:	
Signature:	
Date:	

Tested by:	
Signature:	
Date:	

System Interface Testing Form

Estimated Duration: _____ hours/days

File Id	Field	Pass	Fail

Prepared by:	
Signature:	
Date:	

Tested by:	
Signature:	
Date:	

Program Testing Form

Estimated Duration: _____ hours/days

Program Id	Condition	Pass	Fail

Prepared by:	
Signature:	
Date:	

Tested by:	
Signature:	
Date:	

Report Testing Form

Estimated Duration: _____ hours/days

Report Id	Field	Pass	Fail

Prepared by:	
Signature:	
Date:	

Tested by:	
Signature:	
Date:	

Business Case Testing Form

ID	Description	Estimated Duration (hours/days)

Testing Sign Off

ID	Prepared By	Signature	Date	Tested by	Signature	Date

Business Case 1 — <Name>

[illegible]

Test Failure Assessment Form

System/Application: _____

Test Classification: _____

Business Case ID: _____

Test ID: _____

Bug Classification: Mild ☐ Serious ☐ Critical ☐ Catastrophic ☐

Business Impact: Low ☐ Medium ☐ High ☐

Reported by: _____

Date: ____ / ____ / ____

Estimated Remediation: _____ days.

Failed Condition

Suggested Remediation

Business Acceptance

Accepted by:	
Signature:	

Date:	
-------	--