

Sample System Integration Test Plan

for

XYZ Remote Office Payroll System
Version 1.0

June 5, XXXX

Sample Integration Test Plan

Approvals

Table of Contents

1. INTRODUCTION 4

- 1.1 TEST OBJECTIVES
- 1.2 SCOPE OF TESTING
- 1.3 SYSTEM OVERVIEW
- 1.4 DEFINITIONS/ACRONYMS
- 1.5 REFERENCES

2. APPROACH

- 2.1 ASSUMPTIONS/CONSTRAINTS
 - 2.1.1 *Assumptions*
 - 2.1.2 *Constraints*
- 2.2 COVERAGE
 - 2.2.1 *Software Components*
 - 2.2.2 *Requirements*
 - 2.2.3 *Business Processes*
- 2.3 TEST TOOLS
- 2.4 TEST TYPE (REGRESSION, CONVERSION, ETC.)
- 2.5 TEST DATA

3. PLAN

- 3.1 TEST TEAM
- 3.2 TEAM REVIEWS
- 3.3 MAJOR TASKS AND DELIVERABLES
- 3.4 ENVIRONMENTAL NEEDS
 - 3.4.1 *Test Environment*
 - 3.4.2 *Test Lab*
- 3.5 TRAINING

4. FEATURES TO BE TESTED

- 4.1 BUILD 1
 - 4.1.1 *Table Maintenance*
 - 4.1.2 *Create Timesheets*
 - 4.1.3 *Employee Time Entry*
 - 4.1.4 *System Ease of Use*
 - 4.1.5 *System Performance*
- 4.2 BUILD 2
 - 4.2.1 *Payroll Tax Calculation*
 - 4.2.2 *Create Paychecks*
 - 4.2.3 *Accounting Reports and General Ledger Interface (GreenTree Accounting software)*
 - 4.2.4 *System Ease of Use*
 - 4.2.5 *System Performance*
- 4.3 BUILD 3
 - 4.3.1 *Direct Deposit*
 - 4.3.2 *Submit Payroll Withholding Reports to IRS*
 - 4.3.3 *System Ease of Use*
 - 4.3.4 *System Performance*
 - 4.3.5 *Data Recovery*

5. FEATURES NOT TO BE TESTED

5.1 SYSTEM ADMINISTRATION FUNCTIONS

6. TESTING PROCEDURES

6.1 TEST EXECUTION

6.1.1 *Test Cases*

6.1.2 *Order of Testing*

6.2 PASS/FAIL CRITERIA

6.3 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

6.3.1 *Normal Criteria*

6.3.2 *Abnormal Criteria*

6.4 DEFECT MANAGEMENT

7. RISKS AND CONTINGENCIES

8. APPENDIX

8.1 APPENDIX A: WORK BREAKDOWN STRUCTURE

1.

Introduction

1.1 Test Objectives

The system integration test of the XYZ system should validate from both the requirements perspective and business perspective that:

- All payroll business processes are supported.
- All timekeeping functions work correctly.
- Direct deposit functions work correctly.
- Payroll functions work correctly.
- The system is easy to use by the end-users.
- Payroll policies and procedures are supported by the system.
- The system can be customized by remote offices to handle localized payroll processing needs.
- The system complies with all government payroll tax reporting format requirements.
- Financial controls are adequate to prevent fraudulent transactions.
- Security controls are in place to prevent unauthorized system access.
- All financial calculations are correct.
- All points of integration within the system work as defined in requirements.
- All points of integration with other systems work as defined in requirements.
- Recovery procedures are correct and can be performed by users.

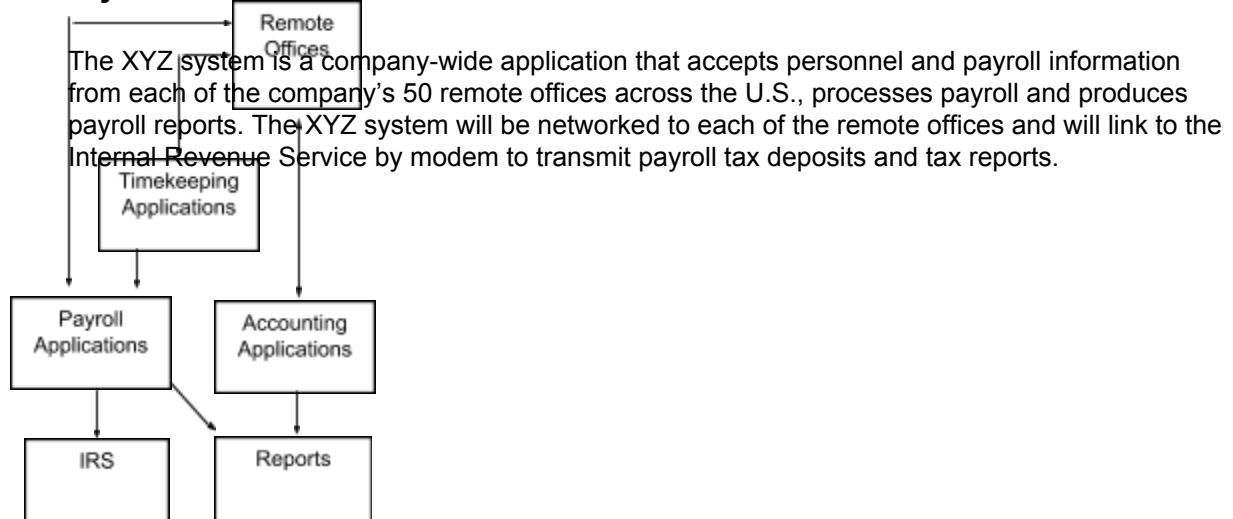
The objective of system integration testing is to validate the system operation as a whole and with other systems. At the conclusion of testing, the project team and the test team will have a high level of confidence that the system will work according to user requirements and will meet business needs.

1.2 Scope of Testing

The system integration test of the XYZ system will include payroll, accounting, and timekeeping applications. In addition, the interfaces to remote offices and the Internal Revenue Service will be tested.

The system integration test of the XYZ system will not include system administration functions.

1.3 System Overview



1.4 Definitions/Acronyms

1.4.1 Definitions

Build	A functionally independent piece of software that supports a well-defined logical subset of a system. A build can be tested independently and then integrated with other builds. Builds can be migrated from one level of testing to another and possibly installed independently of the rest of the system.
Critical Processing Unit	A program, module or unit that is critical to the correct functioning of the system. A critical processing unit carries with it a high impact of failure.
Model Office	A validation of the implementation, operation and training of the system in a simulated office environment.
Prototype	A working model of the software to be built. Demonstrates look and feel of the software, but does not support all features and functions.
Regression Testing	Testing to ensure that unchanged parts of the software work the same as before a change was made.
Requirement	Something that the system should do or be. May be based on user, business, or technical needs.
Static Test	A verification performed without execution on a computer. For example, reviewing a document for accuracy.
System Integration Testing	A level of testing that validates both internal and external system interfaces, ensuring that the system works as a cohesive whole. The purpose of systems integration testing is to perform test cases that validate the system was built according to requirements.
Test Tool	Any vehicle that assists in testing.
Unit Testing	A level of testing in which the smallest units of a system (i.e., modules) are separately tested.
Unit-to-Unit Testing	A level of testing that validates the integration between groups of related modules or units.

1.4.2 Acronyms

FICA	Federal Income Tax
FUTA	Federal Unemployment Tax
IRS	Internal Revenue Service
SSN	Social Security Number

1.5 References

- Requirements Specification Document for the XYZ System
- Test Standards
- Test Procedures
- Test Plan Notebook
- Payroll Policy and Procedures Notebook

2. Approach

2.1 Assumptions/Constraints

2.1.1 Assumptions

- The first build of the XYZ system will be ready for system integration testing on July 1, XXXX.
- Each build of the XYZ system will have passed unit and unit-to-unit testing before it is transferred to the system integration testing environment.

2.1.2 Constraints

- Six weeks might not be enough time to test the entire system and then retest the system to find new defects due to fixes.

2.2 Coverage

Test coverage will be measured by:

- A completed matrix of testable requirements and test cases.
- A completed matrix of business processes and business test cases.

In the event that coverage levels are not met, the QA manager will determine if the actual levels are adequate in light of the system risks.

2.2.1 Software Components

All software modules in the payroll, timekeeping, and accounting sub-systems will be tested.

2.2.2 Requirements

All user requirements as specified in the Requirements Specification Document will be tested.

2.2.3 Business Processes

All critical business processes will be validated completely. Critical business processes are:

- Employee Time Entry
- Payroll Tax Calculation
- Create Paychecks
- Direct Deposit
- Submit Payroll Withholding Reports to the IRS.

2.3 Test Tools

- Capture/Playback
- Test Manager

- Defect Tracker

2.4 Test Type (Regression, Conversion, etc.)

The following types of testing will be performed during system integration testing:

- Functional testing, by performing test cases based on testable requirements
- Functional testing, by performing test cases based on business functions
- Compliance testing, by evaluating system performance against company policies and procedures
- Security testing, by testing each end-user's security access levels
- Controls testing, by testing all financial controls
- Regression testing, to ensure that a change to the system does not introduce new defects.
- Recovery testing, to ensure the system can be restarted by the end user in the event of a system failure.

2.5 Test Data

To perform system integration testing, test data will be supplied from two sources:

- Data created specifically for the system integration test and
- Data obtained from past payroll periods.

The order of test execution allows for test data to be created before it is needed in payroll processing and payroll reporting.

The following test data sources will be located on the central server in the test environment:

- Employee data table (EMPLOYEE) - converted from existing sequential files and supplemented with specific test data that will execute test cases.
- Employee time data (EMPTIME) - entered during the test and converted from existing sequential files
- Tax table for current year and next year (TAXTABLE) - obtained in electronic format from the IRS.

3. Plan

3.1 Test Team

The following people will be on the system integration test team:

Name	Title	Level of involvement	Responsibilities
Joe Johnson	Team Leader - Independent Test Team	40 hrs/wk	Lead all testing activities, including test planning, test execution, and status reporting.
Mary Anderson	Assistant Team Leader - Independent Test Team	40 hrs/wk	Fill in during any absence of team leader. Design and execute test cases, create test data, write test summary report
Pete Wilson	End user - Payroll Dept.	25 hrs/wk	Design and execute test cases for payroll processing.

Tom Jones	End user - Internal Audit Dept.	40 hrs/wk	Design and execute test cases to validate financial controls
Jane Peterson	End user - Personnel Dept.	30 hrs/wk	Design and execute test cases, build employee test tables
Doug Thompson	Independent Tester	40 hrs/wk	Design and execute test cases for time reporting.
Dot Wong	Independent Tester	40 hrs/wk	Design and execute test cases for payroll direct deposit.
Renee Roberts	Independent Tester	40 hrs/wk	Design test cases for payroll reporting to IRS and financial sub-system.
Gary Moore	Developer	40 hrs/wk	Technical assistance as needed during the test

3.2 Team Reviews

The following reviews will be conducted by the entire test team and a representative from the QA department. Refer to the work schedule for the planned review dates.

- Test plan review
- Test case review
- Test progress review
- Post-test review

3.3 Major Tasks and Deliverables

Task	Start	Stop	Deliverable(s)
System integration test case design	5/1/XXXX	6/1/XXXX	System integration test cases
Build system integration test environment	5/15/XXXX	6/15/XXXX	Test environment ready for test data population
Build system integration test data	6/2/XXXX	6/15/XXXX	Employee data table, Employee time data, Tax table for current year and next year.
Train test team	6/15/XXXX	6/17/XXXX	Trained testers
Build 1 delivered for system integration testing		6/29/XXXX	Build 1 ready for system integration testing.
Build 1 test execution	7/1/XXXX	7/15/XXXX	Build 1 tested
Build 1 test summary report due		7/17/XXXX	Build 1 test summary report
Build 2 test execution	7/18/XXXX	7/26/XXXX	Build 2 tested
Build 2 test summary report due		7/28/XXXX	Build 2 test summary report
Build 3 test execution	8/1/XXXX	8/15/XXXX	Build 3 tested
Build 3 test summary report due		8/17/XXXX	Build 3 test summary report
System migration to model office environment	8/20/XXXX	8/22/XXXX	Installed system

3.4 Environmental Needs

3.4.1 Test Environment

Hardware

All test cases will be executed on the Development Server in the QA database environment.

- One (1) networked HP Laser Jet printer.

(2) Asus

Intel Core i5-4460S

8GB DDR3 RAM:

24X Super-Multi DVD/RW drive

(2) Dell Inspiron 3000

Intel® Pentium® G3240 dual-core 3.1 GHz 3MB cache processor

4GB RAM DDR3 1600 SDRAM memory

1TB 7200 RPM serial ATA/600 hard drive, controller type: serial ATA

(2) iMAC

1.4GHz

dual-core Intel Core i5 processor (Turbo Boost up to 2.7GHz) with 3MB shared L3 cache

8GB of 1600MHz LPDDR3 onboard memory

500GB (5400-rpm) hard drive

Network

- LAN
 - Synoptic 810 10Base-T Ethernet Concentrator
 - Category 5 cables to meet 10Base-T specifications

Software

- XYZ application software
- Server
 - GreenTree Accounting version 3.0
 - MS Windows 7 Pro operating system
 - MS Windows 8 Pro operating system

3.4.2 Test Lab

The following items will be needed full-time by the system integration test team:

- 1 whiteboard (large) with markers and erasers
- Clerical/organizational material - file cabinet, storage boxes, folders, notebooks

NOTE: The aforementioned items are in addition to the hardware and software items detailed in Section 3.4.1.

3.5 Training

Test team members who have not been trained in the testing process will be trained in testing techniques by the QA staff. The training will be three days in length and will be conducted at the corporate training facility the dates of 6/15/XXXX - 6/17/XXXX.

4. Features to be Tested

4.1 Build 1

4.1.1 Table Maintenance

- Update project code table entries
- Update department code entries
- Add/Update office codes
- Security (authorization levels for table maintenance)
- Security (authorization and access for system users)

4.1.2 Create Timesheets

- First timesheet
- Last timesheet
- Partial period timesheet
- Security
- Controls

4.1.3 Employee Time Entry

- Overtime entry
- Incomplete entry of time worked
- Normal entry of time worked
- Security
- Controls

4.1.4 System Ease of Use

- Navigation
- Help functions
- Order of processing

4.1.5 System Performance

- Response time for employee time entry
- Transaction throughput time for timesheet creation

4.2 Build 2

4.2.1 Payroll Tax Calculation

- FICA
- Medicare
- Unemployment (FUTA)
- State Income Tax
- Financial controls

4.2.2 Create Paychecks

- Withholding reconciliation
- YTD totals
- Calculations
- Financial controls
- Security

4.2.3 Accounting Reports and General Ledger Interface (GreenTree Accounting software)

- Gross pay and withholdings
- Payroll reconciliation
- Journal listing
- GL account listing
- GL report
- GL transaction reconciliation
- Security
- Financial controls

4.2.4 System Ease of Use

- Navigation
- Help functions
- Order of processing

4.2.5 System Performance

- Response time for GreenTree accounting functions
- Transaction throughput time for paycheck creation

4.3 Build 3

4.3.1 Direct Deposit

- Update employee direct deposit information
- Transmit transactions
- Reconcile transmission report
- Security
- Financial controls

4.3.2 Submit Payroll Withholding Reports to IRS

- Transfer protocol correct
- Calculations correct
- Transmit payroll reports to the IRS
- Transmit weekly payroll tax deposit
- Security
- Financial controls

4.3.3 System Ease of Use

- Navigation
- Help functions
- Order of processing

4.3.4 System Performance

- Transaction throughput time for direct deposit and payroll reporting

4.3.5 Data Recovery

- Recovery from remote transmission errors
- Recovery from interruptions in batch processing
- Recovery from interruptions in online processing
- Recovery from General Protection Faults (GPFs)

5. Features Not to be Tested

5.1 System Administration Functions

- User Password Administration
- File Security Procedures

6. Testing Procedures

6.1 Test Execution

6.1.1 Test Cases

For each requirement, business process, or system feature to be tested, the tester will execute a set of pre-defined test cases. Each test case will have a series of actions and expected results. As each action is performed, the results are evaluated. If the observed results are equal to the expected results, a checkmark is placed in the “pass” column. If the observed results are not equal to the expected results, a checkmark is placed in the “fail” column.

6.1.2 Order of Testing

The order of testing will be driven primarily by the build order. Within each build, the following order of testing will be followed:

Build 1

1. Table Maintenance
2. Create Timesheets
3. Employee Time Entry
4. System Ease of Use
5. System Performance

Build 2

1. Payroll Tax Calculation
2. Create Paychecks
3. Accounting Reports and General Ledger Interface (GreenTree Accounting software)
4. System Ease of Use
5. System Performance

Build 3

1. Direct Deposit
2. Submit Payroll Withholding Reports to IRS
3. System Ease of Use
4. System Performance
5. Data Recovery

6.2 **Pass/Fail Criteria**

To pass the system integration test, the following criteria must be met:

- All payroll business processes are supported.
- All timekeeping functions work correctly.
- Direct deposit functions work correctly.
- Payroll functions work correctly.
- The system is easy to use by the end-users.
- Payroll policies and procedures are supported by the system
- The system can be customized by remote offices to handle localized payroll processing needs.
- The system complies with all government payroll tax reporting format requirements.
- Financial controls are adequate to prevent fraudulent transactions.
- Security controls are in place to prevent unauthorized system access.
- All financial calculations are correct.
- All points of integration within the system work as defined in requirements.
- All points of integration with other systems work as defined in requirements.
- The system can be recovered by a user following documented recovery procedures in the event of a system failure.

6.3 **Suspension Criteria and Resumption Requirements**

6.3.1 **Normal Criteria**

At the end of each day (5:00 p.m.) testing will be suspended. At that time, all test cases executed during the day should be marked as such. The system integration test team will initiate a backup routine to save the day's updated test files.

When all test cases have been executed, the test will be suspended and the results documented for the System Integration Test Summary Report.

6.3.2 Abnormal Criteria

As a general guideline, if the defect backlog continually increases over a two week period, testing should be suspended. This will allow the developers time to fix existing defects without the pressure and confusion of new defects being added to the backlog. **When a change is being migrated to the test environment, the system integration test team leader must be notified in advance to schedule a time for the move.** After the move has been completed, a retest of previously tested functions should be performed.

If a critical processing unit is found to have severe defects (as defined by the defect reporting process), testing should be suspended until the defects have been fixed. When the fixed unit is moved back into the test environment, any previously performed tests that affect the unit should be performed again to ensure new defects were not created as a result of the fix.

6.4 Defect Management

It is the intention of the System Engineering Test team to use PVCS Tracker for reporting, maintaining, tracking and overall management of the defects on the XYZ Payroll System. Change management procedures have been developed and have been described in the Project Test Plan.

The assignment and description of defect severity levels will be as follows:

- | | |
|--------------|--|
| 1 - Critical | Business objectives or completion of test case are impacted. |
| 2 - High | Defects which prove to be detrimental to the system. Testing should not progress to the next build until corrective measures have been taken. |
| 3 - Medium | Defects which provide invalid/incorrect information. An example of a priority 3 defect could be a miscalculation of overtime pay, or a numeric entry is allowed in an alpha only field - which corrupts other database information. |
| 4 - Low | Defects are esthetic in nature. An example of a priority 4 defect could be the misplacement of an entry button on the left side of the screen when the user requirements stated it should be on the right side of the screen. Functionality is NOT impacted. |
| 5 - Info | An item observed during testing that may require further information. This type of priority could be assigned to a work order for an item encountered that is not clear in the requirements. |

7. Risks and Contingencies

This section describes the system or project risks and the contingency plans that should take effect if the project experiences problems.

- Timesheet Creation - Risk level low to moderate. Should a problem occur, new timesheets can be created and re-distributed with a low level of expense.
- Employee Time Entry - Risk level moderate to high. Should a problem occur, employees' time records and pay could be affected. Correcting a major defect in production could be very expensive.
- Payroll Processing - Risk level moderate to high. Should a problem occur, employees' pay could be affected. Correcting a major defect in production could be very expensive.
- Payroll Reporting - Risk level moderate to high. Should a problem occur, employees' tax withholding and pay could be affected. If the withholding tax reports sent to the IRS are incorrect, fines and back interest may be assessed. Correcting a major defect in production could be very expensive.

8. Appendix

8.1 Appendix A: Work Breakdown Structure