

---

**KTH Ingenjörsskolan Kista**

---

**Knowledge Management Intranet System  
Test Plan**

**Version 1.0**

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

## Revision History

Date	Version	Description	Author
2000-12-05	1.0	Initial Release – Prototype Test Plan	Madde and Lennart

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

## Table of Contents

1.	Introduction	4
1.1	Objectives	4
1.2	Scope	4
1.3	Project Identification	5
2.	Test Requirements	6
3.	Test Strategy	7
3.1	Testing Types	7
3.1.1	Data and Database Integrity Testing	7
3.1.2	Function Testing	8
3.1.3	Business Cycle Testing	9
3.1.4	User Interface Testing	9
3.1.5	Performance Testing	10
3.1.6	Load Testing	11
3.1.7	Stress Testing	11
3.1.8	Volume Testing	11
3.1.9	Security and Access Control Testing	12
3.1.10	Failover and Recovery Testing	13
3.1.11	Configuration Testing	13
3.1.12	Installation Testing	13
3.2	Tools	14
4.	Resources	14
4.1	Workers	14
4.2	System	16
4.3	Test Model	16
4.4	Test Logs	16
4.5	Defect Reports	16
5.	Project Tasks	17

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

# Test Plan

## 1. Introduction

### 1.1 Objectives

This document describes the plan for testing the architectural prototype of the Knowledge management system. This Test Plan document supports the following objectives:

- ?? Identify existing project information and the software that should be tested.
- ?? List the recommended test requirements (high level).
- ?? Recommend and describe the testing strategies to be employed.
- ?? Identify the required resources and provide an estimate of the test efforts.
- ?? List the deliverable elements of the test activities.

### 1.2 Scope

This Test Plan describes the integration and system tests that will be conducted on the architectural prototype.

The testing will include testing of source code, and all module interfaces.

The purpose of assembling the architectural prototype was to test feasibility and performance of the selected architecture. It is critical that all system and subsystem interfaces be tested as well as system performance at this early stage. Testing of system functionality and features will not be conducted on the prototype.

The interfaces between the following subsystems will be tested:

1. Login
2. Administrator interface
3. Knowledge management module

The external interfaces to the following devices will be tested:

1. Local PCs
2. Remote PCs.

The most critical performance measures to test are:

1. Response time for remote login to the system.
2. Response time to access the knowledge management module.
3. Response time to access the administrator interface
4. System response time when system loaded with 100 simultaneous users.
5. User responsetime when 45 simultaneous accesses to the Knowledge management database.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

### 1.3 Project Identification

The table below identifies the documentation and availability used for developing the *test plan*:

Document (and version / date)	Created or Available	Received or Reviewed	Author or Resource	Notes
Use-Case Reports	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No		
Project Plan	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No		
Prototype	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No		
User's Manuals	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No		
Business Model or Flow	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No		
Data Model or Flow	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No		
Risk List	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No		

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

## 2. Test Requirements

The listing below identifies those items (use cases, functional requirements, non-functional requirements) that have been identified as targets for testing. This list represents *what* will be tested.

### Data and Database Integrity Testing

Verify access to Knowledge Management Database.

Verify simultaneous read and write accesses.

Verify correct retrieval of update of database data.

### System Testing

The system shall interface with existing systems at Flow Interactive.

The client component of the system shall operate on any personal computer regardless of operating system.

The server component of the system shall run under the Windows 2000 Operating System.

The client component of the system shall operate on any personal computer with Netscape Navigator 4.0 and Microsoft Internet Explorer 4.0 or better.

### Business Cycle Testing

None.

### User Interface Testing

Verify ease of navigation through a sample set of screens.

Verify sample screens conform to GUI standards.

The System shall be easy-to-use and shall be appropriate for the employees at Flow Interactive.

### Performance Testing

Verify response time for login.

Verify response time to access Knowledge Management database.

Verify response time for remote login.

Verify response time for remote access of Knowledge Management database.

The system shall provide access to the Knowledge Management database with no more than a 10 second latency.

### Load Testing

Verify system response when loaded with 100 logged on users.

Verify system response when 45 simultaneous users accesses Knowledge management database.

### Stress Testing

None.

### Volume Testing

None.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

### Security and Access Control Testing

Verify Logon from a local PC.

Verify Logon from a remote PC.

Verify Logon security through user name and password mechanisms.

### Failover / Recovery Testing

None.

### Configuration Testing

The web-based interface shall be compatible with the Java 1.2 VM runtime environment.

### Installation Testing

None.

## 3. Test Strategy

The Test Strategy presents the recommended approach to the testing of the software applications. The previous section on Test Requirements described *what* will be tested; this describes *how* it will be tested.

The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.

In addition to the considerations provided for each test below, testing should only be executed using known, controlled databases, in secured environments.

The following test strategy is generic in nature and is meant to apply to the requirements listed in Section 4 of this document.

### 3.1 Testing Types

#### 3.1.1 Data and Database Integrity Testing

The databases and the database processes should be tested as separate systems. These systems should be tested without the applications (as the interface to the data). Additional research into the DBMS needs to be performed to identify the tools / techniques that may exist to support the testing identified below.

Test Objective:	Ensure Database access methods and processes function properly and without data corruption.
Technique:	<p>?? Invoke each database access method and process, seeding each with valid and invalid data (or requests for data).</p> <p>?? Inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved (for the correct reasons)</p>
Completion Criteria:	All database access methods and processes function as designed and without any data corruption.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

- Special Considerations:
- ?? Testing may require a DBMS development environment or drivers to enter or modify data directly in the databases.
  - ?? Processes should be invoked manually.
  - ?? Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events.

### 3.1.2 *Function Testing*

Testing of the application should focus on any target requirements that can be traced directly to use cases and rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques, that is, verifying the application (and its internal processes) by interacting with the application via the GUI and analysing the output (results). Identified below is an outline of the testing recommended for each application:

- Test Objective: Ensure proper application navigation, data entry, processing, and retrieval.
- Technique:
- ?? Execute each use case, use case flow, or function, using valid and invalid data, to verify the following:
  - ?? The expected results occur when valid data is used.
  - ?? The appropriate error / warning messages are displayed when invalid data is used.
  - ?? Each rule is properly applied.
- Completion Criteria:
- ?? All planned tests have been executed.
  - ?? All identified defects have been addressed.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

### 3.1.3 Business Cycle Testing

This section is not applicable to test of the architectural prototype.

### 3.1.4 User Interface Testing

User Interface testing verifies a user's interaction with the software. The goal of UI Testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the applications. In addition, UI Testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

Test Objective:	Verify the following: <ul style="list-style-type: none"> <li>?? Navigation through the application properly reflects functions and requirements, including window to window, field to field, and use of access methods (tab keys, mouse movements, accelerator keys)</li> <li>?? Window objects and characteristics, such as menus, size, position, state, and focus conform to standards.</li> </ul>
Technique:	?? Create / modify tests for each window to verify proper navigation and object states for each application window and objects.
Completion Criteria:	Each window successfully verified to remain consistent with benchmark version or within acceptable standard
Special Considerations:	?? Not all properties for custom and third party objects can be accessed.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

### 3.1.5 Performance Testing

Performance testing measures response times, transaction rates, and other time sensitive requirements. The goal of Performance testing is to verify and validate the performance requirements have been achieved. Performance testing is usually executed several times, each using a different “background load” on the system. The initial test should be performed with a “nominal” load, similar to the normal load experienced (or anticipated) on the target system. A second performance test is run using a peak load.

Additionally, Performance tests can be used to profile and tune a system’s performance as a function of conditions such as workload or hardware configurations.

Test Objective:	Validate System Response time for designated operations under the following two conditions: <ul style="list-style-type: none"> <li>- normal anticipated workload</li> <li>- anticipated worst case workload</li> </ul>
Technique:	<ul style="list-style-type: none"> <li>?? Use Test Procedures developed for Business Model Testing (System Testing).</li> <li>?? Modify data files or modify scripts to increase the number of iterations of each operation.</li> <li>?? Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual, <i>see special considerations below</i>).</li> </ul>
Completion Criteria:	<ul style="list-style-type: none"> <li>?? Single user: Successful completion of the test scripts without any failures and within the expected / required time allocation</li> <li>?? Multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation.</li> </ul>
Special Considerations:	<ul style="list-style-type: none"> <li>?? Comprehensive performance testing includes having a “background” load on the server. There are several methods that can be used to perform this, including:</li> <li>?? “Drive transactions” directly to the server, usually in the form of SQL calls.</li> <li>?? Create “virtual” user load to simulate many (usually several hundred) clients. Remote Terminal Emulation tools are used to accomplish this load. This technique can also be used to load the network with “traffic.”</li> <li>?? Use multiple physical clients, each running test scripts to place a load on the system.</li> <li>?? Performance testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.</li> <li>?? The databases used for Performance testing should be either actual size, or scaled equally.</li> </ul>

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

### 3.1.6 Load Testing

Load testing measures subjects the system-under-test to varying workloads to evaluate the system's ability to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics (response times, transaction rates, and other time sensitive issues).

Test Objective:	Verify System Response time for designated operations under varying workload conditions.
Technique:	?? Use tests developed for testing. ?? Modify data files (to increase the number of operations) or the tests to increase the number of times each operation occurs.
Completion Criteria:	?? Multiple users: Successful completion of the tests without any failures and within acceptable time allocation.
Special Considerations:	?? Load testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement. ?? The databases used for load testing should be either actual size, or scaled equally.

### 3.1.7 Stress Testing

This section is not applicable to test of the architectural prototype.

### 3.1.8 Volume Testing

This section is not applicable to test of the architectural prototype.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

### 3.1.9 Security and Access Control Testing

Security and Access Control Testing focus on two key areas of security:

- Application security, including access to the Data or Business Functions, and
- System Security, including logging into / remote access to the system.

Application security ensures that, based upon the desired security, users are restricted to specific functions or are limited in the data that is available to them. For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that user "type" one can see all customer information, including financial data, however, user two only sees the demographic data for the same client.

System security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways.

Test Objective:	<p>Function / Data Security: Verify that user can access only those functions / data for which their user type is provided permissions.</p> <p>System Security: Verify that only those users with access to the system and application(s) are permitted to access them.</p>
Technique:	<p>?? Function / Data Security: Identify and list each user type and the functions / data each type has permissions for.</p> <p>?? Create tests for each user type and verify permission by creating transactions specific to each user type.</p> <p>?? Modify user type and re-run tests for same users. In each case verify those additional functions / data are correctly available or denied.</p> <p>?? System Access (see special considerations below)</p>
Completion Criteria:	For each known user type the appropriate function / data are available and all transactions function as expected and run in prior Application Function tests
Special Considerations:	?? Access to the system must be reviewed / discussed with the appropriate network or systems administrator. This testing may not be required as it maybe a function of network or systems administration.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

### 3.1.10 *Failover and Recovery Testing*

This section is not applicable to test of the architectural prototype.

### 3.1.11 *Configuration Testing*

Configuration testing verifies operation of the software on different software and hardware configurations. In most production environments, the particular hardware specifications for the client workstations, network connections and database servers vary. Client workstations may have different software loaded (e.g. applications, drivers, etc.) and at any one time many different combinations may be active and using different resources.

Test Objective:	Validate and verify that the client Applications function properly on the prescribed client workstations.
Technique:	<ul style="list-style-type: none"> <li>?? Use Integration and System Test scripts</li> <li>?? Open / close various PC applications, either as part of the test or prior to the start of the test.</li> <li>?? Execute selected transactions to simulate user activities into and out of various PC applications.</li> <li>?? Repeat the above process, minimizing the available conventional memory on the client.</li> </ul>
Completion Criteria:	For each combination of the Prototype and PC application, transactions are successfully completed without failure.
Special Considerations:	<ul style="list-style-type: none"> <li>?? What PC Applications are available, accessible on the clients?</li> <li>?? What applications are typically used?</li> <li>?? What data are the applications running (i.e. large spreadsheet opened in Excel, 100 page document in Word).</li> <li>?? The entire systems, network servers, databases, etc. should also be documented as part of this test.</li> </ul>

### 3.1.12 *Installation Testing*

This section is not applicable to test of the architectural prototype.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

### 3.2 Tools

The following tools will be employed for testing of the architectural prototype:

	Tool	Vendor/In-house	Version
Test Management	Requisite Pro Rational Unified Process	Rational Rational	TBD
Test Design	Rose	Rational	TBD
Defect Tracking	ClearQuest	Rational	TBD
Functional Testing	Rational Robot	Rational	TBD
Performance Testing	Visual Quantify	Rational	TBD
Test Coverage Monitor or Profiler	Visual Pure Coverage	Rational	TBD
Other Test Tools	Purify TestFactory	Rational Rational	TBD
Project Management	Project Word Excel	Microsoft Microsoft Microsoft	TBD
DBMS tools	TBD	TBD	TBD

## 4. Resources

This section presents the recommended resources for the Knowledge Management Intranet System project, their main responsibilities, and their knowledge or skill set.

### 4.1 Workers

This table shows the staffing assumptions for the test of the Prototype.

Human Resources		
Worker	Minimum Resources Recommended  (number of workers allocated full-time)	Specific Responsibilities/Comments
Test Manager	1– Madelene Ödquist 2– Lennart Henriksson	Provides management oversight Responsibilities: ?? Provide technical direction ?? Acquire appropriate resources ?? Management reporting

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

Test Designer	Madelene Ödquist Lennart Henriksson	Identifies, prioritises, and implements test cases Responsibilities: ?? Generate test plan ?? Generate test model ?? Evaluate effectiveness of test effort
System Tester	Helena Schuber	Executes the tests Responsibilities: ?? Execute tests ?? Log results ?? Recover from errors ?? Document defects
Test System Administrator	Kjell Holmlin	Ensures test environment and assets are managed and maintained. Responsibilities: ?? Administer test management system ?? Install / manage worker access to test systems
Database Administration / Database Manager	Madelene Ödquist Lennart Henriksson Helena Schuber Kjell Holmlin	Ensures test data (database) environment and assets are managed and maintained. Responsibilities: ?? Administer test data (database)
Designer	Peter Eriksson	Identifies and defines the operations, attributes, and associations of the test classes Responsibilities: ?? Identifies and defines the test class(es) ?? Identifies and defines the test packages
Implementer	All programmers are responsible for creating test classes to their own classes	Implements and unit tests the test classes and test packages Responsibilities: ?? Creates the test classes and packages implemented in the test model.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

#### 4.2 System

The following table sets forth the system resources for the testing

System Resources	
Resource	Name / Type / Serial No.
PDC Server	IBM PC 300 GL: A7
Client Test PC's	
Remote PCs (with internet access)	Serial No: Serial No: Serial No:
Local PCs (connected via LAN)	Serial No: Serial No: Serial No:
Test Development PC's - 4	IBM PC 300 GL: Bone IBM PC 300 GL: Pluto IBM PC 300 GL: Tellus DELL Optiplex GX110: X??

#### 4.3 Test Model

The Test Model will define all the test cases and will reference the test procedures and test scripts which are associated with each test case.

#### 4.4 Test Logs

It is planned to use Requisite Pro to identify the test cases and to track the status of each test case. The test results will be summarized in Requisite Pro as untested, passed, conditional pass, or failed. In summary, Requisite Pro will be setup to support the following attributes for each test case.

- ?? Test status
- ?? Build Number
- ?? Tested By
- ?? Date Tested
- ?? Test Notes

It will be the responsibility of the System Tester to update the test status in Requisite Pro.

#### 4.5 Defect Reports

ClearQuest will be used for logging and tracking individual defects.

Knowledge Management <i>Intranet System</i>	Version: 1.0
Test Plan	Date: 2000-12-05
test_plan.TPL	

## 5. Project Tasks

Below are the test related tasks for testing the Knowledge Management Architectural Prototype:

### Plan Test

- Identify Requirements for Test
- Assess Risk
- Develop Test Strategy
- Identify Test Resources
- Create Schedule
- Generate Test Plan

### Design Test

- Workload Analysis (not applicable for Prototype)
- Develop Test Model
- Identify and Describe Test Cases
- Identify and Structure Test Procedures
- Review and Access Test Coverage

### Implement Test

- Setup Test Environment
- Record or Program Test Scripts
- Develop Test Stubs and Drivers
- Identify Test-Specific functionality in the design and implementation model
- Establish External Data sets

### Execute Test

- Execute Test Procedures
- Evaluate Execution of Test
- Recover from Halted Test
- Verify the results
- Investigate Unexpected Results
- Log Defects

### Evaluate Test

- Evaluate Test-Case Coverage
- Evaluate Code Coverage
- Analyze Defects
- Determine if Test Completion Criteria and Success Criteria have been achieved
- Create Test Evaluation Report