NIMBLE

TEST

MANAGEMENT

Contents

Nimble Test Management

What we're going to cover

Testing is an essential aspect of software development in bug and issue tracking. Gemini's approach to testing is to base it on the combined Agile principles of valuing simplicity and supporting multi-disciplinary teams. Our Testing platform is simple so that everybody in the software development team gets it, which is why we call our flavour of test management **Nimble Testing**.
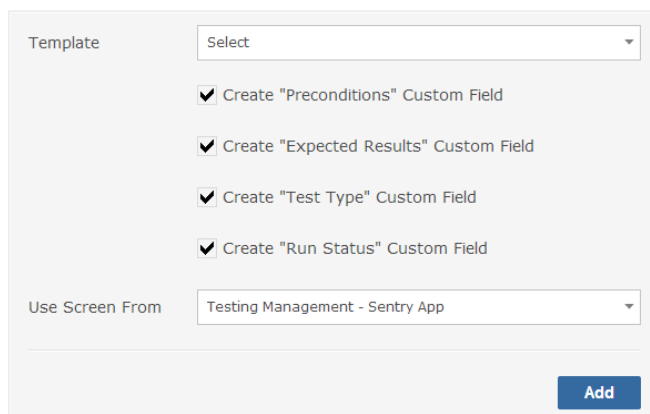
Testing functionality is contained largely, but not exclusively, within an app that comes out-of-the-box, called Sentry. Sentry is not enabled by default but to switch it on is a trivial task for someone with the Gemini Administrator privilege. Once enabled, Sentry will give you access to Test Plans, which implements the Test Runner, and Test Cases, which contain Test Steps. This guide explains the functionality behind these four components of Sentry.

Switching Sentry on

We said that testing functionality is largely but not exclusively in Sentry and the reason for this is Gemini's approach to delivering great, flexible functionality to any project, through Project Templates. Project Templates define the fields, field-level permissions, workflow, workflow permissions and apps that a project built from the Template will use.

Simply adding Sentry apps and Custom Fields to a Project Template instantly enables Nimble Testing on any projects based on that Template, whether the projects are used exclusively for testing or not. But how will you know which Sentry apps and Custom Fields to deploy? Gemini makes it easy, a Gemini Administrator simply needs to navigate to: Customize…Apps and in the list of deployable apps in the left hand toolbar select Sentry. The popup window that is shown allows you to add Sentry functionality to any pre-existing Template and to create the standard Custom Fields to store test Preconditions and Expected Results (rich textboxes). The functionality also lets you create two Custom Fields with default lookup values for Test Types (e.g. black box, regression etc.) and Run Statuses. Finally, because you may be placing Sentry meta data on a Template designed just for testing, you can clone the field list and field permissions from an existing screen on a different Template.

» Add testing capabilities to a template by adding test plans and cases.

Template	Select ▾
	☑ Create "Preconditions" Custom Field
	☑ Create "Expected Results" Custom Field
	☑ Create "Test Type" Custom Field
	☑ Create "Run Status" Custom Field
Use Screen From	Testing Management - Sentry App ▾
	Add

Fig 1.0 Adding Sentry functionality to any Gemini Template

Test Plans

Test Plans are the top level holding structure of all your Nimble Testing. Because Test Plans are just a Type, with [Screens](#) and [Workflow](#) in a Project Template, you can customize the fields and field-level permissions that are held at that level. If, for example, at the Test Plan level you wished to record the application that was being tested then you could simply create a Custom Field of type dropdown list, place your application values in it and add the field to the Test Plan Screens (making it mandatory if that is a requirement). You can also change the Workflow to match your needs by adding/deleting Status values, or simply altering the transition flow or permissions.

If you add the Type Test Plan to your Project Template using the Sentry app in Customize…Apps then the apps that Test Plan needs will be automatically added for you. If you add Test Plan manually then you should know that Test Plan needs the following apps in the Content area of Screens (scroll to the very bottom of the Screens layout to react the Content section):

- Test Cases
- Plan Test Items - an optional list of items in Gemini to be tested. This list is not populated on the Test Plan but rather is view only at that level as it is a summary of items to be tested on all the Test Cases that belong to the Test Plan.
- Test Runner – the wizard that executes tests
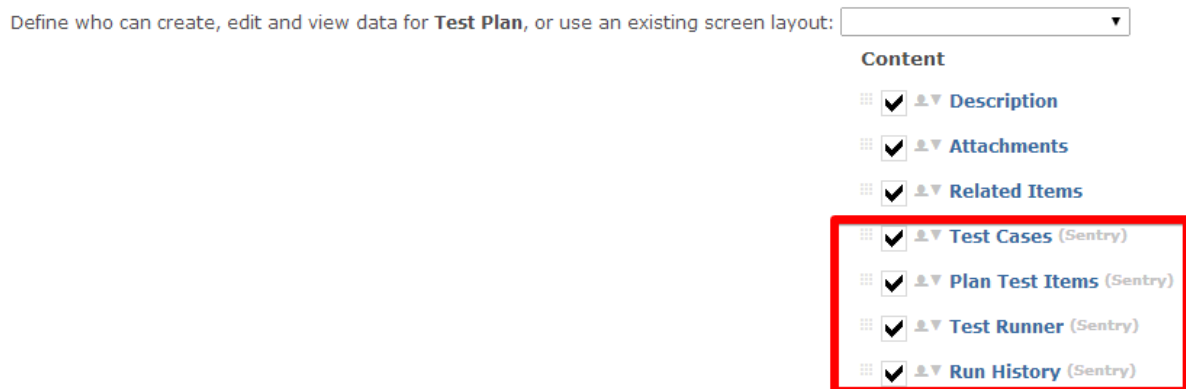- Run History – a sophisticated history of all executed Test Runs

Fig 2.0 The Content area in the Screens section of the Test Plan Type

In Nimble Testing one creates Test Cases and appends them to one or more Test Plans, which must contain at least one Test Case. The Test Runner belongs to the Test Plan, not a Test Case and therefore this is the level at which tests are executed and results are stored.

Note: It is not uncommon for users to create a generic Test Plan that holds a generic Test Case as a reusable structure for ad hoc testing. You can execute as many Test Runs as you like against a single Test Plan and you can vary the Test Cases against a Test Plan as you need. Some organizations have Test Plans that belong to individual Testers, who simply vary the Test Cases on their Test Plans as needed. The modular Test Plan/Test Case structure and the fact that the Test Runner will track the history of every Test for you makes testing in Gemini very flexible.

Test Cases

Test Cases are the level at which the tests themselves are defined and broken down into a manageable number of logical steps, each of which should have identified preconditions and an expected result. If you have used the Sentry App in customize to populate a Project Template with Sentry then you will have Preconditions, Expected Result and Test Type as three fields on your Test Case. Test Type is not to be confused with Type on its own. Test Case is a Type, to which Screens, Fields and Workflow etc. belong. Test Type is simply a lookup field with pre-populated values.

If you add the Test Case Type to your Project Template using the Sentry app in Customize…Apps then the apps that Test Case needs will be automatically added for you. If you add Test Case manually then you should know that Test Case needs the following apps in the Content area of Screens (scroll to the very bottom of the Screens layout to react the Content section):

- Test Steps
- Items Tested – these will amalgamate on the Test Plan as the view only Plan Test Items - an optional list of items in Gemini to be tested.
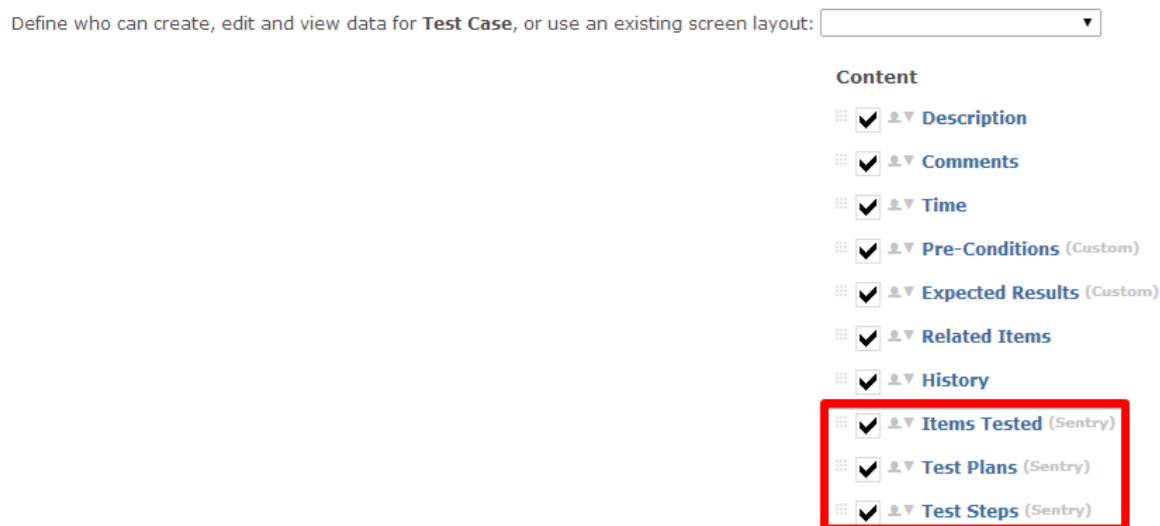- Test Plans – the reciprocal control that lets you find the Plans on which a Test Case exists

Fig 2.1 The Content area in the Screens section of the Test Case Type

Note: *A Test Case can be reused across any number of Test Plans, since it the Plan that executes the test and stores the result. Try therefore to make Test Cases as generic and reusable as possible.*

Test Steps

You cannot put a Test Case on a Test Plan unless it has at least one Test Step. A common mistake that people make when setting up Sentry is to create a dummy Test Case, forget to give it any steps and then try to add it to a Plan – it won't work, there is nothing to do, the Test Plan won't find any such Test Case because it is invalid. True to the principles of Nimble Testing, Gemini's Test Steps are very simple. You describe the step the Tester is meant to execute, the tangible result that should indicate success or failure and that's it.

The following image lists the steps of a single Test Case, in the order in which they should be executed and what the Tester will see if the step is successful. The shaded box on the extreme left is

a drag-handle when you hover over it, to allow you to drag-drop the steps into whatever order is needed.

Fig 3.0 Sample Test Steps of a Test Case

Items Tested

Whether you use Items Tested or not very much depends on the overall use of Gemini. If you are using Gemini for Bug and Issue Tracking then it makes sense that when a specific bug/defect/issue is being tested it is linked to the Test Case that relates to it. Since that item might hold the description of the problem, the contact details of the user who reported it and other information, this could be a useful connection for a Tester. However, it isn't a requirement of the system and a number of organizations that use Sentry do not use this field, preferring instead to focus on the fact that a Test Case relates to an area of functionality, all of which will be tested in its entirety regardless of individual problems.

Test Runner

Once the Test Steps are defined for a Test Case and the Test Case(s) linked to a Test Plan, testing can begin. The Test Runner is a wizard that walks the Tester through each of the steps of the test in turn, letting that person record whether a given step has passed or failed (there is no other option).
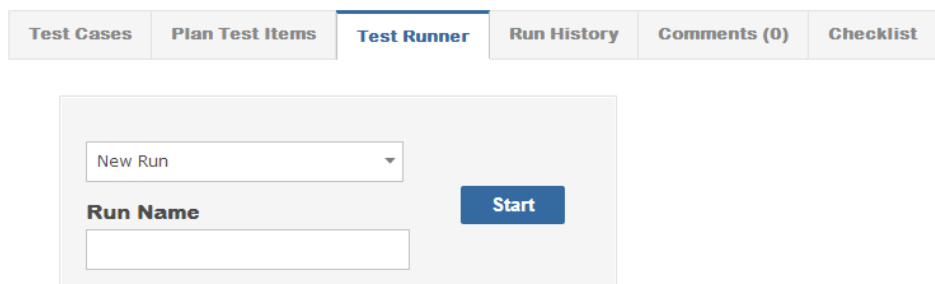

Figure 4.0 The Test Runner

Starting Tests

The Test runner will let you start a new Test Run or restart an incomplete Test Run at any point in time. The dropdown above the Run Name (see fig 4.1 below) will show all previous Test Runs in an incomplete status. If you select an incomplete Test Run and click on Start then you will be taken to the step immediately following the last executed Step in that Run.

If you wish to start a new Teat Run, irrespective of whether or not there are previous incomplete Runs, simply type in a name in the Run Name textbox and click on Start.

Note: *You may adopt whatever naming convention you wish for your Test Runs but we recommend you include a Tester's initials, a date and a sequence number e.g. DS-Aug-01-2014-001.*

Test Cases	Plan Test Items	**Test Runner**	Run History	Comments (0)	Checklist

New Run ▾

Run Name

Start

Passing Tests

To pass a test the Tester has only to click on the Pass button (see fig 4.0 above). The Test Runner will automatically advance to the next Test Step, displaying the preconditions and expected results of each step. When all Test Steps of a Test Case are complete the wizard will advance to the next Test Case and when all Test Cases are complete, if there have been no failed Test Steps the run will be marked as Complete and Passed.

Failing Tests

If a Test Step should fail (not deliver the expected result), the Tester has two choices:

- Fail the Step
- Fail & Log (Fail the Step and create a new defect/bug/issue)
  - This new item can be of any Type in any Project that the Tester has access to
  - The Source field of this new item will link back to the Test Plan and the failed run. The Source field will display as a hyperlink and when clicked it will drill back to the precise failed step using JavaScript (that must therefore be supported in the browser)

The Actual Results description field cannot be left blank on a failed step.

When a Test Step fails an additional option is shown to the Tester allowing him or her to fail the entire Test Run. Gemini will let you fail a step and continue testing or abandon the run at any failure point.

Note: Whether passing or failing a step, Gemini allows you to:

- Enter Comments in the Actual Results field

- Attach files to the Test Step results (screenshots etc). Simply click on the Add hyperlink on the line that says Attachments and browse for your attachment(s).
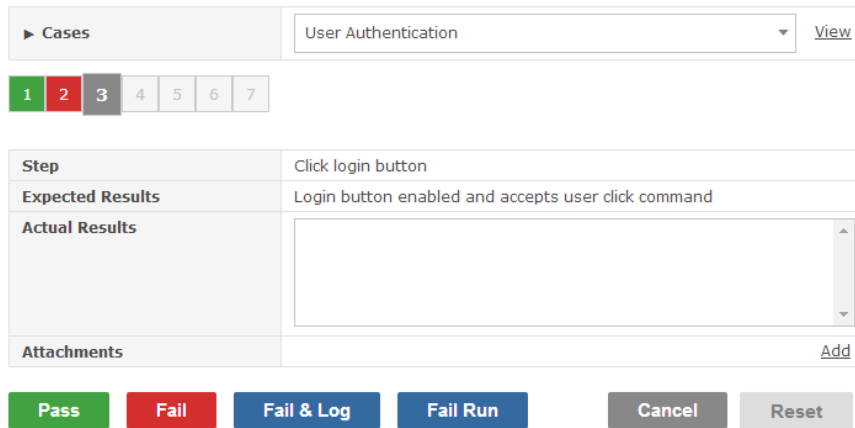

Figure 4.2 The Test Runner with the Fail Run option (step 2 has been failed)

Restarting Steps and Cases

To restart a Test Run from a previous Test Case:

1. Click on the dropdown list of Test Cases, if it is not currently in view and then click on the Test Step number in the top left. Gemini will display the details so you can be sure you are at the correct point.
2. Click Reset. You will be shown the following warning:

If you select Yes, to continue the reset, then as the message states the step you are on and all subsequent steps will be reset as if they had never been tested. Gemini will not let you reset a step in the middle of a test sequence, you must start again from a given point in time. Any attachments that had been added to the Test Step results for all reset steps will be deleted.