



Functional Test 작성

목차

Functional Test 작성	1	연습 5: 검증 포인트 및 오브젝트 맵 보기	8
소개: Functional Test 작성	1	검증 포인트 보기	8
연습 1: Rational Functional Tester 설정	2	오브젝트 맵 보기	9
로깅 옵션 설정	2	연습 6: 회귀 테스트 수행	10
Functional Tester 프로젝트 작성	3	연습 7: 비교기를 사용하여 검증 포인트 갱신	11
연습 2: 스크립트 레코딩	3	연습 8: 오브젝트 맵 갱신	13
레코딩 시작	3	오브젝트 맵에서 오브젝트 인식 특성 보기	13
응용프로그램 시작	4	맵에 새 오브젝트 추가	14
조치 레코딩	4	오브젝트 통합	14
연습 3: 검증 포인트 작성	4	스크립트 다시 재생	15
데이터 검증 포인트 작성	4	연습 9: 인식 환경 설정 변경	16
이미지 검증 포인트 작성	5	연습 10: 정규식 사용	16
특성 검증 포인트 작성	5	오브젝트 맵 열기 및 오브젝트 통합	17
암호 필드 테스트	6	특성 값을 정규식으로 변환	17
연습 4: 스크립트 재생	7	요약: Functional Test 작성	18

Functional Test 작성

이 Rational® Functional Tester 학습서를 활용하여 Functional Test를 작성하고 재생하는 데 필요한 주요 유스 케이스를 학습할 수 있습니다. 이 종합 학습서는 제품과 함께 설치되어 있는 샘플 Java™ 응용프로그램을 사용합니다.

학습 목표

이 학습을 완료하고 나면 다음을 수행할 수 있습니다.

- Functional Test 프로젝트 작성 및 스크립트 레코딩
- 검증 포인트, 오브젝트 맵 및 정규식으로 작업
- 비교기를 사용하여 검증 포인트 갱신
- 스크립트 재생
- 회귀 테스트 수행

소요 시간

45분

관련 정보

PDF 버전 보기

학습: 데이터 기반 Functional Test 작성

학습: 키워드에 기초한 Manual Test 자동화

샘플: Functional Test 프로젝트

소개: Functional Test 작성

이 학습서를 활용하여 Functional Tester 사용을 시작할 수 있으며 기본 조작을 테스트하고 수행하는 데 필요한 주요 유스 케이스를 학습할 수 있습니다. 이 학습서에서는 Functional Tester에서 제공되는 샘플 응용프로그램을 사용하여 모든 작업을 수행할 수 있습니다.

Functional Tester 학습서는 열 개의 연습으로 이루어져 있으며 학습서를 제대로 작동하기 위해서는 이를 순차적으로 완료해야 합니다.

학습 목표

이 학습을 완료하고 나면 다음을 수행할 수 있습니다.

- Functional Test 프로젝트 작성 및 스크립트 레코딩
- 검증 포인트, 오브젝트 맵 및 정규식으로 작업

- 비교기를 사용하여 검증 포인트 갱신
- 스크립트 재생
- 회귀 테스트 수행

주: 시작하기 전에 학습서를 인쇄하여 연습 시에 인쇄 사본을 활용하십시오. 학습서의 PDF 버전을 인쇄하거나 각 주제 항목 안에서 마우스 오른쪽을 클릭한 다음 인쇄를 클릭하여 개별 연습 내용을 인쇄할 수 있습니다.

소요 시간

이 학습서를 완료하려면 약 45분이 소요됩니다. 이 학습서와 관련된 다른 개념을 찾아 볼 경우, 시간이 더 오래 걸릴 수 있습니다.

전제조건

이는 기초 학습서입니다. Rational Functional Tester 사용 경험이 없는 경우에도 타스크를 수행할 수 있습니다.

연습 1: Rational Functional Tester 설정

IBM®은 Java 응용프로그램의 테스트를 위해 설치 및 사용되는 JRE(Java Runtime Environment)를 제공합니다. 학습서와 관련하여 이 JRE를 사용하십시오. 사용자 고유의 Java 또는 HTML 응용프로그램을 테스트하려는 경우, 인에이블러를 실행하고 사용자 환경 및 응용프로그램을 구성해야 합니다. 이 설정 타스크에 대한 자세한 정보는 제품 환영 페이지의 첫 단계 섹션에서 Functional Tester 시작하기 마법사를 참조하십시오. 이 제는 사전 구성된 JRE를 계속해서 사용하기 위해 아무런 작업도 수행할 필요가 없습니다.

Rational Functional Tester를 시작한 다음 첫 번째 테스트 스크립트를 레코딩하기 전에 다음 타스크를 수행하십시오.

로깅 옵션 설정

Rational Functional Tester는 몇 가지 로깅 옵션을 제공합니다. 여기서는 HTML 로그를 사용합니다.

1. 창 → **Perspective** 열기 → 기타를 클릭하여 Functional Test Perspective를 여십시오. Perspective 열기 대화 상자에서 **Functional Test** 선택사항을 선택하십시오.
2. HTML 로깅이 설정되었음을 확인하려면 창 → 환경 설정을 클릭하십시오.
3. 환경 설정 창의 왼쪽 분할창에서 **Functional Test**를 펼친 다음 재생을 펼치고 로깅을 클릭하십시오.
4. 기본값 사용 선택란이 선택되어 있으며 로그 유형 필드에 **html**이 표시되는지 확인하십시오.
5. 확인을 클릭하십시오.

이 설정은 스크립트 재생 후에 HTML 로그를 자동으로 엽니다.

Functional Tester 프로젝트 작성

레코딩을 시작하기 전에 Functional Tester 프로젝트를 작성해야 합니다.

1. Functional Tester 메뉴에서 파일 → 새로 작성 → **Functional Test** 프로젝트를 클릭하십시오.
2. 프로젝트 이름 아래에 FTtutorial을 입력하십시오(간격 없음).
3. 프로젝트 위치 아래에 C:\WFTproject를 입력하십시오. Functional Tester가 해당 디렉토리를 작성합니다.
4. 소스 제어 옵션이 사용 가능한 경우, 소스 제어에 프로젝트 추가를 선택하지 마십시오.
5. 프로젝트 연관 옵션이 사용 가능한 경우, 현재 **Rational** 프로젝트와 **Functional Test** 프로젝트 연관을 선택하지 마십시오.
6. 완료를 클릭하십시오.

이제 FTtutorial 프로젝트가 Functional Test 프로젝트 보기에 나타납니다. 이는 Functional Test Perspective의 왼쪽 분할창에 있습니다.

연습 2: 스크립트 레코딩

이 연습에서는 Functional Tester 레코딩 모니터를 사용하여 스크립트를 레코딩합니다.

레코딩 시작

이제 레코딩을 시작할 준비가 되었습니다.

1. 레코딩을 시작하려면 Functional Test 도구 모음에서 **Functional Test** 스크립트 레코딩 단추(●)를 클릭하십시오.
2. 방금 작성한 FTtutorial 프로젝트를 선택하십시오.
3. 스크립트 이름 필드에 Classics(사용할 응용프로그램의 이름)를 입력하십시오.
4. 소스 제어에 스크립트 추가 옵션이 사용 가능한 경우, 이를 선택하지 마십시오.
5. 완료를 클릭하십시오.

Functional Tester 창이 자동으로 최소화되며 레코딩 모니터가 표시됩니다.


레코딩 모니터에 대한 추가 학습: Functional Tester 레코딩 모니터는 레코딩을 시작할 때마다 표시됩니다. 화면에 표시되지 않도록 모니터를 최소화할 수 있으며 해당 크기를 조정할 수도 있습니다. 도구 모음만 표시 단추(☐)를 클릭하면 레코딩 모니터가 숨겨지며 도구 모음만 표시됩니다. 모니터 표시 단추(▣)를 클릭하면 다시 모니터가 표시됩니다. 이 학습서에서는 모니터를 표시된 상태로 두십시오. 모니터는 레코딩 세션 중에 수행되는 모든 조치에 대한 메시지를 표시합니다(예: 레코딩 시작 및 일시정지, 응용프로그램 또는 브라우저 시작, 응용프로그램내의 클릭, 검증 포인트 삽입, 스크립트에 기타 항목 삽입).

6. 모니터 메시지 환경 설정 도구 모음 단추(⚙)를 클릭하십시오. 이 옵션을 사용하면 언제든지 모니터에서 텍스트의 형태를 제어할 수 있습니다.
7. 취소를 클릭하십시오.
8. 스크립트 지원 명령 삽입 도구 모음 단추(🔗)를 클릭하십시오.

스크립트 지원 기능 창이 열립니다. 이 창을 사용하여 다른 스크립트 호출, 로그 항목 삽입, 타이머 삽입, 일시 정지(sleep) 명령(지연) 삽입 또는 스크립트에 주석 삽입을 수행할 수 있습니다.

9. 닫기를 클릭하십시오.

응용프로그램 시작

1. 테스트 응용프로그램을 시작하려면 **응용프로그램 시작 도구 모음 단추**()를 클릭하십시오.
2. 응용프로그램 시작 창에서 **ClassicsJavaA**를 선택한 다음 **확인**을 클릭하십시오.

Functional Tester 학습서 샘플 응용프로그램인 ClassicsCD가 열립니다. 레코딩 모니터가 응용프로그램 전면에 있으면 이를 화면의 오른쪽 하단 모서리로 끌어 놓을 수 있습니다.

조치 레코딩

이 응용프로그램에서 주문하기를 레코딩하려고 합니다.

1. **Haydn** 다음의 **+**를 클릭하여 **Composers** 트리의 폴더를 펼치십시오.
2. 목록에서 **Symphonies Nos. 94 & 98**을 클릭하십시오.
3. 주문하기 단추를 클릭하십시오.
4. 구성원 로그인 창에서 **Existing Customer** 및 **Trent Culpito**의 기본 설정을 유지하십시오. 이 시점에서 암호 필드를 클릭하지 마십시오.
5. **확인**을 클릭하십시오.
6. 카드 번호 필드에 신용카드 번호를 입력하십시오. 네 자리 숫자의 4 세트로 구성된 올바른 형식을 사용해야 합니다(예: 7777 7777 7777 7777).
7. 만기 날짜 필드에 올바른 형식의 만기 날짜를 입력하십시오(예: 07/07).
8. 주문하기를 클릭하십시오.
9. 주문 확인 메시지 창에서 **확인**을 클릭하십시오.


연습 3: 검증 포인트 작성


이 연습에서는 검증 포인트를 레코딩하여 오브젝트를 테스트합니다. 검증 포인트는 특정 조치의 발생을 검증하거나 오브젝트의 상태를 검증합니다.

특성 검증 포인트, 이미지 검증 포인트 또는 여섯 가지 유형의 데이터 검증 포인트를 작성할 수 있습니다. 검증 포인트를 작성하면 응용프로그램에서 오브젝트 정보를 캡처하여 재생 도중 비교에 필요한 기준선 정보를 설정합니다.

데이터 검증 포인트 작성



데이터 검증 포인트를 레코딩하여 Composers 트리를 캡처합니다.

1. 레코딩 모니터에서 검증 포인트 또는 조치 명령 삽입 단추()를 클릭하십시오.

2. 검증 포인트 및 조치 마법사의 오브젝트 선택 페이지에서 오브젝트를 선택한 후 다음 페이지로 진행 옵션이 선택되어 있는 경우, 이를 지우십시오.
3. 오브젝트 파인더()를 사용하여 응용프로그램에서 Composers 트리를 선택하십시오. 오브젝트 파인더를 클릭하고 이를 트리 위로 끌어 놓으십시오. 마우스 단추를 누르고 있으면 전체 트리가 빨간색 경계로 아웃라인되며 오브젝트 이름이 빨간색 경계 다음의 화면 팁에 표시됩니다(javax.swing.JTree). 마우스 단추를 해제하여 선택을 실행하면 오브젝트에 대한 인식 특성이 오브젝트 선택 페이지 맨 아래의 눈금에 나열됩니다.
4. 다음을 클릭하십시오.
5. 조치 선택 페이지에서 데이터 검증 포인트 수행을 선택하고 다음을 클릭하십시오.
6. 검증 포인트 데이터 명령 삽입 페이지에서 데이터 값 필드의 트리 계층 구조 테스트를 선택하십시오. 이 테스트는 전체 트리 계층 구조에 대한 정보를 캡처합니다.
7. 검증 포인트 이름 필드에서 Classics_tree를 입력하고 다음을 클릭하십시오.
8. 검증 포인트 데이터 페이지가 오른쪽 분할창의 눈금에 캡처된 데이터를 표시합니다. 항목 옆의 상자에 선택 표시가 나타나면 해당 항목이 테스트됩니다. 기본적으로 모든 항목이 선택되어 있습니다. 선택된 상태로 두십시오. 선택되지 않았으면 모두 선택 단추를 클릭하십시오.
9. 완료를 클릭하십시오.

이미지 검증 포인트 작성

선택된 CD에 적절한 앨범이 표시되는지 확인하기 위해 이미지 검증 포인트를 삽입할 수 있습니다.

1. 레코딩 모니터에서 검증 포인트 또는 조치 명령 삽입 단추()를 클릭하십시오.
2. 검증 포인트 및 조치 마법사의 오브젝트 선택 페이지에서 오브젝트를 선택한 후 다음 페이지로 진행 옵션이 선택되어 있는 경우, 이를 지우십시오.
3. 오브젝트 파인더()를 사용하여 응용프로그램에서 앨범 이미지를 선택하십시오. 오브젝트 파인더를 클릭하고 이를 앨범 이미지 위로 끌어 놓으십시오. 마우스 단추를 누르고 있으면 앨범 이미지가 빨간색 경계로 아웃라인되며 오브젝트 이름이 빨간색 경계 옆의 화면 팁에 표시됩니다(javax.swing.JLabel). 마우스 단추를 해제하여 선택을 실행하면 오브젝트에 대한 인식 특성이 오브젝트 선택 페이지 맨 아래의 눈금에 나열됩니다.
4. 다음을 클릭하십시오.
5. 조치 선택 페이지에서 이미지 검증 포인트 수행을 선택하고 다음을 클릭하십시오.
6. 이미지 검증 포인트 명령 삽입 페이지에서 검증 포인트 이름으로 Album_image를 입력하십시오.
7. 전체 이미지 선택 옵션을 선택하고 다음을 클릭하십시오.
8. 검증 포인트 데이터 페이지가 오른쪽 분할창에 캡처된 이미지를 표시합니다. 완료를 클릭하십시오.

특성 검증 포인트 작성

이제 다른 검증 포인트를 삽입하여 올바른 고객에 대한 주문인지 확인할 수 있습니다. 특성 검증 포인트는 확인 화면의 텍스트를 캡처합니다.

1. ClassicsCD 응용프로그램에서 주문 → 기존 주문 상태 보기를 클릭하십시오. 이 시점에서는 암호 필드를 클릭하지 마십시오.
2. 확인을 클릭하십시오. 기존 주문 보기 창에서 "Order for Trent Culpito" 레이블을 테스트합니다.
3. 레코딩 모니터에서 검증 포인트 또는 조치 명령 삽입 단추(🔍)를 클릭하십시오.
4. 오브젝트 선택 페이지에서 오브젝트를 선택한 후 다음 페이지로 진행 옵션을 선택합니다.
5. 오브젝트 파인더를 "Order for Trent Culpito" 레이블 위로 끌어 놓아 이를 선택하십시오. 마우스 단추를 누르고 있으면 레이블이 빨간색 경계로 아웃라인되며 오브젝트 이름이 표시됩니다(javax.swing.JLabel). 다음 페이지로 진행 옵션을 선택했으므로 오브젝트를 선택하면 조치 선택 페이지가 열립니다.
6. 위에서 두 번째 조치인 특성 검증 포인트 수행을 선택하고 다음을 클릭하십시오.
7. 특성 검증 포인트 명령 삽입 페이지에서 하위 포함 필드를 없음으로 설정했는지 확인하십시오.
8. 검증 포인트 이름 아래에 제안된 기본값을 승인하십시오.
9. 표준 특성 옵션 사용이 선택된 상태에서 다음을 클릭하십시오. 검증 포인트 데이터 페이지에 테스트 오브젝트 특성 및 해당 값이 눈금 형식으로 표시됩니다. 특성 열에서 테스트할 특성을 선택하고 값 열에서 특성 값을 편집할 수 있습니다.

오브젝트 특성 선택에 대한 추가 학습: 기본적으로 어떠한 특성도 선택되어 있지 않습니다. 오브젝트 특성을 테스트하려면 테스트하려는 각각의 특성을 선택하십시오. 선택한 특성은 이 검증 포인트를 사용하여 스크립트를 재생할 때마다 테스트됩니다. 눈금 위의 모두 선택 도구 모음을 클릭하여 목록에서 모든 특성을 선택할 수 있습니다. 모두 선택 취소 단추를 사용하여 모든 특성을 지우십시오. 특성 검증 포인트를 사용하는 경우 최상의 결과를 얻으려면 관심을 두고 있는 특성만 테스트하십시오. 이 경우에는 주문이 올바른 고객에 대한 주문인지 여부를 판별하기 위해 텍스트 특성만 테스트합니다.

10. 특성 열에서 **text**, **opaque** 및 **visible** 특성을 선택하여 재생하는 동안 테스트하십시오. 계속 선택하려면 선택란을 두 번 클릭해야 합니다.
11. 완료를 클릭하십시오.
12. ClassicsCD 기존 주문 보기 창에서 닫기를 클릭하십시오.

암호 필드 테스트

이제 이전에 테스트하지 않은 암호 필드를 테스트하기 위해 다른 빠른 주문을 수행합니다.

1. Composers 트리에서 **Haydn** 폴더를 펼치십시오.
2. **Symphonies Nos. 94 & 98**을 클릭하십시오.
3. 주문하기 단추를 클릭하십시오.
4. 구성원 로그인 창에서 **Existing Customer** 및 **Trent Culpito**의 기본 설정을 유지하십시오.
5. 이번에는 암호 필드에 xxxx를 입력하십시오.
6. 암호 기억 옵션을 선택하십시오.
7. 확인을 클릭하십시오.
8. 유효한 카드 번호 및 만기 날짜를 입력하십시오(예: 7777 7777 7777 7777, 만기 날짜 07/07).

9. 주문하기를 클릭하십시오.
10. 주문 확인 메시지 상자에서 확인을 클릭하십시오.
11. x 단추를 클릭하여 ClassicsCD 응용프로그램을 닫으십시오.
12. 레코딩 도구 모음에서 레코딩 중지 단추(■)를 클릭하십시오.

레코딩을 중지하면 Rational Functional Tester가 레코딩 모니터를 닫은 다음 스크립트 및 오브젝트 맵을 프로젝트 디렉토리에 기록합니다. Rational Functional Test 창이 복원되고 기본 창에 스크립트가 표시됩니다.

연습 4: 스크립트 재생

이 연습에서는 스크립트를 재생하고 Rational Functional Tester 인터페이스의 일부를 확인할 수 있습니다. 방금 레코딩한 스크립트가 활성 스크립트이므로 재생 단추를 클릭하면 스크립트가 재생됩니다.

1. 스크립트를 재생하려면 Functional Test 도구 모음에서 **Functional Test** 스크립트 실행 단추(▶)를 클릭하십시오.
2. 로그 선택 창에서 기본 로그 이름 **Classics**를 유지한 상태에서 완료를 클릭하십시오.

Rational Functional Tester가 최소화되고 재생 모니터가 화면의 오른쪽 맨 위에서 시작됩니다. 스크립트가 재생되면 재생 모니터에 메시지가 표시됩니다. Rational Functional Tester는 검증 포인트, 응용프로그램 램에서 수행한 조치, 응용프로그램 시작과 같이 레코딩된 모든 조치를 재생합니다.

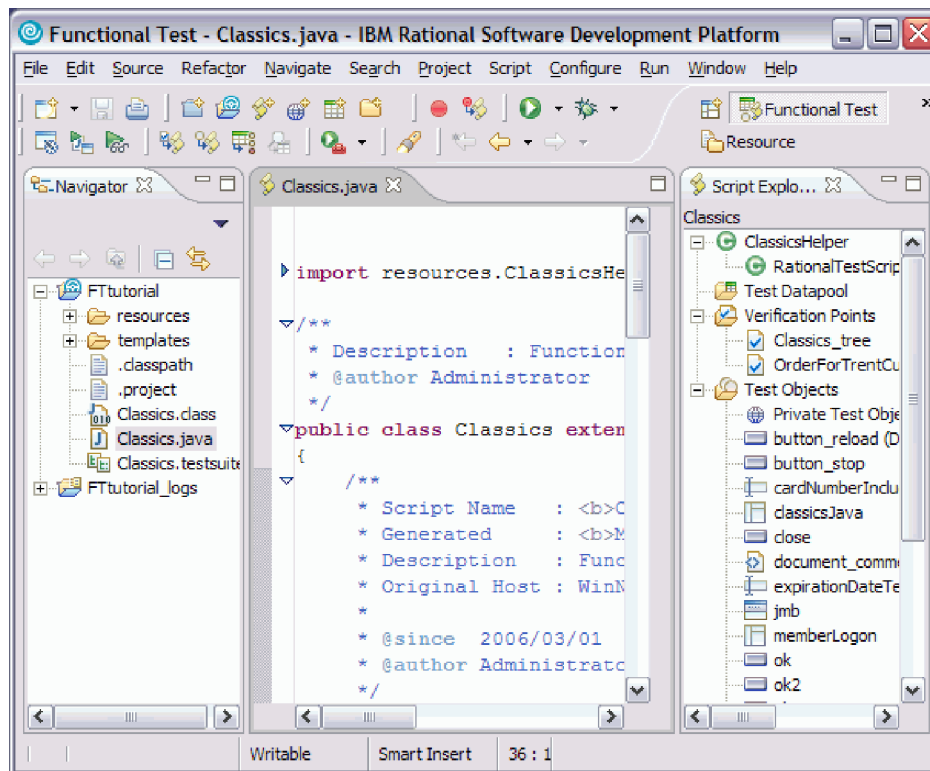
재생이 완료되면 HTML 로그가 별도의 창에 테스트 실행 결과를 표시합니다. 로그에 나열된 각 이벤트에는 초록색의 이벤트 표제에 Pass가 포함되어 있어야 합니다. 레코딩한 두 개의 검증 포인트가 나열되는지 확인하십시오.

3. 로그를 닫으십시오. 이제 스크립트를 정상적으로 레코딩하고 재생했으므로 보다 자세히 Functional Test Perspective를 볼 수 있습니다.
4. Functional Test 창이 최소화되어 있으면 이를 복원하십시오. 여러 스크립트가 있는 경우, Functional Tester는 Java 편집기(스크립트 창)에 프로젝트의 모든 열린 스크립트를 표시합니다.

Java 편집기에 대한 추가 학습: 스크립트 전체에서 스크립트에 대한 정보가 밝은 청색으로 맨 위에 표시되며 별표가 접두부로 사용되는지 확인하십시오. 이 정보는 스크립트 템플릿에서 가져오는 것으로 수정이 가능합니다. 스크립트 템플릿 수정에 대한 자세한 정보는 Functional Tester 도움말을 참조하십시오.

다음 행이 참조하는 오브젝트를 식별하기 위해 Functional Tester가 초록색 문자로 스크립트에 짧은 설명을 추가하는지 확인하십시오. 이 정보를 사용하면 더욱 쉽게 스크립트를 탐색할 수 있습니다. 사용자 입력을 포함하여 레코딩 중에 인수로서 메소드에 전달되는 문자열은 밝은 청색입니다.

커서가 스크립트의 특정 영역을 가리키면 Functional Tester가 팝업 텍스트 상자에 유용한 정보를 표시합니다. 예를 들어, 헬퍼 메소드의 경우에는 오브젝트 맵의 설명 특성 세트 다음에 오브젝트의 인식 특성이 표시됩니다. 환경 설정에서 풍선 기능을 제어합니다. 이 기능을 OFF 상태로 전환하거나 표시된 내용을 수정하려면 창 → 환경 설정을 클릭한 다음 **Java** → 편집기를 선택하고 풍선 탭을 클릭하십시오. 풍선 기능은 기본적으로 ON 상태입니다.



Java 편집기(스크립트 창)의 왼쪽에는 현재 연결되어 있는 모든 Functional Tester 프로젝트를 나열하는 Functional Test 프로젝트 보기가 있습니다. 각 프로젝트 내의 모든 스크립트는 프로젝트 이름 아래에 나열됩니다. 이 프로젝트 보기에서 또 다른 방법을 사용하여 다른 스크립트를 탐색할 수 있습니다. 프로젝트 보기에서 스크립트를 두 번 클릭하면 스크립트 창에서 스크립트가 열리고 활성 스크립트가 됩니다.

Java 편집기 오른쪽에는 활성 스크립트의 검증 포인트 및 오브젝트 맵을 나열하는 스크립트 탐색기가 있습니다. 스크립트 탐색기에서 검증 포인트 편집기를 시작하여 검증 포인트를 표시하고 편집할 수 있으며, 오브젝트 맵 편집기를 시작하여 오브젝트 맵을 표시하고 편집할 수 있습니다. 스크립트 탐색기 또는 task 보기 및 콘솔 보기와 같은 Functional Test Perspective의 기타 파트에 대한 자세한 정보는 Functional Tester 도움말을 참조하십시오.

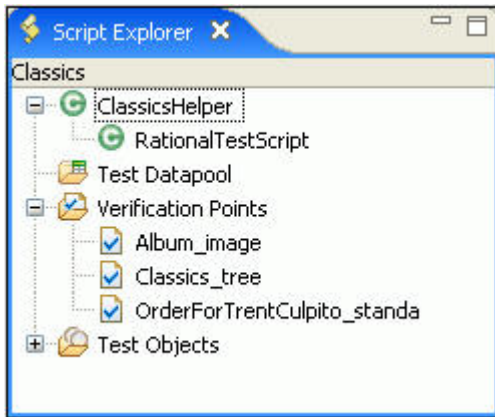
연습 5: 검증 포인트 및 오브젝트 맵 보기

이 연습에서는 검증 포인트 및 오브젝트 맵을 보고 수정하는 방법을 학습합니다.

검증 포인트 보기

검증 포인트 내의 데이터를 검사하고 수정할 수 있습니다.

1. Rational Functional Tester에서 Classics.java 스크립트가 Java 편집기에서 활성 스크립트인지 확인하십시오.
2. 레코딩한 세 개의 검증 포인트가 스크립트 오른쪽의 스크립트 탐색기에 나열되어야 합니다. 필요한 경우, 검증 포인트 옆의 더하기 부호(+)를 클릭하여 목록을 펼치십시오.



3. **Classics_tree**를 두 번 클릭하십시오.

이 검증 포인트는 Composers 목록에서 레코딩한 첫 번째 검증 포인트입니다. 검증 포인트 편집기가 시작됩니다. 향후 재생을 위해 검증 포인트 데이터를 갱신할 수 있습니다.

검증 포인트 갱신: 데이터 검증 포인트에는 여섯 개의 표시 유형이 있습니다. 이 검증 포인트는 데이터(트리) 검증 포인트입니다. 오브젝트 유형은 트리(이 경우에는 javax.swing.JTree)입니다. 이 트리의 데이터를 편집하려면 트리에서 임의의 하위 항목을 두 번 클릭하여 작은 편집 상자를 열고 변경사항을 작성하십시오. 각 항목 옆의 선택란을 사용하면 이 항목을 향후 재생 시 테스트할지 여부를 표시할 수 있습니다. 검증 포인트 편집기 사용에 대한 자세한 정보는 Functional Tester 도움말을 참조하십시오.

4. 검증 포인트 편집기를 닫으십시오.

오브젝트 맵 보기

오브젝트 맵 내의 데이터를 검사하고 수정할 수 있습니다.

1. 스크립트 탐색기에서 테스트 오브젝트 폴더를 펼치십시오.

첫 번째 항목인 개인용 테스트 오브젝트 맵은 이 스크립트의 오브젝트 맵입니다. 개인용 테스트 오브젝트 맵 아래에 나열된 개별 오브젝트는 레코딩 중에 활동한 오브젝트에 대한 참조입니다.

2. 개인용 테스트 오브젝트 맵(🌐)을 두 번 클릭하여 이를 여십시오.

오브젝트 맵 유형: 스크립트를 레코딩하면 Functional Tester는 테스트 중인 응용프로그램에 대한 오브젝트 맵을 작성합니다. 각각의 스크립트는 오브젝트 맵 파일과 연관되어 있습니다. 맵 파일은 개인용(하나의 스크립트와만 연관됨)이거나 다수의 스크립트 중에서 공유될 수 있습니다. 스크립트를 레코딩한 경우, Rational Functional Tester는 기본 설정(개인용 맵)을 사용한 것입니다. 오브젝트 맵에는 각 오브젝트에 대한 특성이 포함되어 하나의 중심 위치에서 정보를 쉽게 갱신할 수 있습니다. 또한 해당 오브젝트를 참조하는 모든 스크립트는 갱신된 정보도 공유합니다.

3. 최상위 레벨 오브젝트 Java: Frame: logFrame1: javax.swing.JFrame을 펼치십시오.

프레임 오브젝트에는 로그인 대화 상자가 포함되어 있습니다. 단일 선택 단추, 암호 필드 및 조치 단추가 프레임 오브젝트 아래에 나열됩니다.

4. 오브젝트 중 하나를 클릭하십시오.

인식 특성은 오브젝트 트리 아래의 눈금에 표시됩니다. 오브젝트 맵은 스크립트에 오브젝트 참조를 추가하는 빠른 방법도 제공합니다. 오브젝트 맵 메뉴에서 **테스트 오브젝트** → **오브젝트 삽입**을 클릭하여 오브젝트를 추가할 수 있습니다. 오브젝트 맵에서 인식 특성의 가중치 변경, 인식 특성 및 값 편집과 같은 기타 조작도 수행할 수 있습니다. 학습서에서 나중에 오브젝트 맵을 사용하여 일부 고급 프로시저를 수행합니다.

5. 오브젝트 맵 메뉴에서 **환경 설정** → **닫을 때 상태 지우기**를 클릭하십시오.

닫을 때 상태 지우기 명령은 토크 메뉴 항목으로 기본적으로 설정되어 있기 때문에 이를 지웁니다. 설정된 상태로 있는 경우, 맵을 닫을 때 모든 오브젝트가 승인됩니다. 변경사항 작성을 위해 오브젝트 맵으로 리턴할 때 추후 단계에서 이 작업을 수행할 수 있습니다.

6. 오브젝트 맵을 닫으십시오. 작성된 변경사항은 저장하지 마십시오.

연습 6: 회귀 테스트 수행

이 연습에서는 다른 빌드에서 스크립트를 실행합니다. 응용프로그램의 새 빌드가 있는 경우, 새 빌드에서 스크립트를 재생함으로써 레코딩한 자동화된 테스트를 실행할 수 있습니다. 새 빌드에서 스크립트를 실행하려면 스크립트에서 응용프로그램의 이름을 변경해야 합니다. (여기서는 응용프로그램의 새 빌드 가져오기를 시뮬레이션하기 위해 이 작업을 수행하는 것이므로 개발 프로젝트에서는 이 작업을 수행할 필요가 없습니다.)

1. Java 편집기(스크립트 창)에서 스크립트(Classics.java)가 활성 스크립트인지 확인하십시오.

스크립트 맨 위의 템플릿 정보 아래에 있는 응용프로그램 시작 명령을 참고하십시오.

```
startApp("ClassicsJavaA");
```

2. "A"를 "B"로 변경하십시오.

Java 코드는 대소문자를 구분하므로 반드시 대문자 B를 사용하십시오. 변경사항을 적용하기 위해 스크립트를 저장하거나 컴파일할 필요는 없습니다. 스크립트를 실행하면 자동으로 적용됩니다.

3. **Functional Test** 스크립트 실행 도구 모음 단추(🔍)를 클릭하여 스크립트를 재생하십시오.
4. 로그 선택 창에서 **Classics**를 선택한 다음 **완료**를 클릭하십시오. 로그를 겹쳐쓸 것인지 묻는 메시지가 표시되면 **예**를 클릭하십시오.

스크립트는 빠르게 재생을 시작하지만 구성원 로그인 창의 종료 시점에서 느려집니다. 그 이유는 응용프로그램의 빌드 B가 선택란 옆의 필드에 다른 텍스트를 보유하고 있기 때문입니다. Functional Tester는 빌드 A에서 레코딩된 인식 특성과 일치하는 오브젝트를 찾습니다. 이 학습서에서 추후에 이 문제점을 정정하는 방법을 설명합니다.

6. 재생 후에 로그가 열리면 메시지를 확인하십시오. 로그에서 두 개의 실패 및 하나의 경고가 나타나야 합니다. (연습 7의 준비 단계로 로그를 열어 두십시오.)

응용프로그램의 변경사항으로 인해 특성 검증 포인트(OrderForTrentCulpito_standar) 및 이미지 검증 포인트(Album_image)가 실패합니다. 다음으로 검증 포인트 기준선을 갱신하여 이를 정정하는 방법을 살펴봅

니다. 오브젝트 인식 경고가 암호 선택란 필드에 대해 생성되었습니다. 또한 학습서의 후반 섹션에서 정규식을 사용하여 오브젝트 맵에서 이를 정정하는 방법을 설명합니다.

ClassicsB의 기본 화면이 ClassicsA의 기본 화면과 다르게 보이는 것을 확인할 수 있습니다. 하지만 이러한 차이점으로 인해 스크립트에 실패가 발생한 것은 아닙니다. 동일 오브젝트가 존재하지만 두 개의 응용 프로그램은 서로 다른 위치에 있습니다. Functional Tester가 강력한 인식 메소드를 사용하여 오브젝트를 찾기 때문에 이로 인해 실패가 발생한 것은 아닙니다. 예를 들어, 오브젝트를 찾기 위해 화면 좌표와 같은 표면적인 특성을 사용하지는 않습니다. 그 대신에 내부 인식 특성을 사용합니다. 이 메소드는 스크립트의 변경이나 재레코딩을 수행하지 않고도 사용자 인터페이스 디자인에서 유연성을 제공합니다.

연습 7: 비교기를 사용하여 검증 포인트 갱신

스크립트를 재생한 후에 검증 포인트 비교기를 사용하여 검증 포인트 데이터를 비교할 수 있습니다. 검증 포인트는 오브젝트의 데이터 또는 특성의 기준선을 제공합니다. 검증 포인트가 응용프로그램의 후속 빌드에서 실패하는 경우, 응용프로그램에 대한 의도적 변경 또는 결함을 발견하게 됩니다. 의도적으로 변경한 경우, 테스트가 추후 빌드에 대해 계속 유효하도록 검증 포인트의 정보를 갱신할 수 있습니다.

연습 6의 마지막에서 로그를 열어 두었습니다. 로그를 닫은 경우에는 프로젝트 보기에서 로그 이름을 두 번 클릭하여 다시 여십시오.



1. 로그에서 실패한 이미지 검증 포인트 항목 끝에 있는 **결과 보기** 링크를 클릭하십시오. 이벤트 표제는 "검증 포인트(Album_image)"입니다.

Functional Tester 검증 포인트 비교기가 검증 포인트 데이터를 표시합니다. 비교기 배너에는 검증 포인트의 이름이 포함됩니다.

비교기에 문제가 있습니까?: 비교기가 열리지 않거나 오류 메시지를 수신한 경우, 브라우저의 Java 플러그인을 사용 가능하게 해야 합니다. 이 작업에 대한 지시사항은 Functional Tester 도움말의 "레코딩하기 전에" 섹션에서 브라우저의 Java 플러그인 사용 가능" 주제를 참조하십시오.

검증 포인트가 실패하면 비교기는 예상 및 실제 값을 표시하여 차이점을 분석할 수 있도록 합니다. 기준선 파일을 로드하고 이를 편집하거나 실제 파일의 값으로 이를 갱신할 수 있습니다. 실패는 빨간색으로 표시됩니다.



ClassicsA에서 검증 포인트를 작성한 경우에 캡처된 앨범 이미지는 javax.swing.JLabel 오브젝트에 기초합니다. ClassicsB에서 스크립트를 재생하는 경우, javax.swing.JLabel 오브젝트의 높이와 너비가 다르므로 이미지 검증 포인트가 실패합니다. 따라서 기준선 파일을 갱신하여 ClassicsB와 일치하도록 오브젝트를 변경해야 합니다.

2. 비교기 도구 모음에서 **편집할 기준선 로드** 단추()를 클릭하십시오.
3. 비교기 도구 모음에서 **실제 값으로 기준선 바꾸기** 단추()를 클릭하십시오. 실제 이미지가 기준선 이미지로 로드됩니다.
4. 비교기를 닫으십시오.


5. 로그에서 실패한 특성 검증 포인트 항목 끝에 있는 결과 보기 링크를 클릭하십시오. 이벤트 표제는 "검증 포인트(OrderforTrentCulpito_standard)"입니다.
6. 텍스트 특성으로 화면 이동하십시오.

ClassicsA에서 검증 포인트를 작성한 경우에 배너 제목은 "Order for Trent Culpito"였습니다. ClassicsB에서 스크립트를 재생한 경우에 배너 제목은 "Orders for Trent Culpito"이었습니다. 고객이 주문 창에서 다수의 주문을 할 수 있기 때문에 "Orders"가 올바릅니다. 따라서 기준선 파일을 갱신하여 ClassicsB와 일치하도록 텍스트를 변경해야 합니다.

기준선 파일만 편집할 수 있습니다.

7. 비교기 도구 모음에서 편집할 기준선 로드 단추()를 클릭하십시오. 이제 왼쪽 값 열에 기준선 값이 표시됩니다.
8. 텍스트 특성으로 화면 이동하는 대신에 특성 열 위의 첫 번째 차이점으로 건너뛰기 단추()를 클릭할 수 있습니다. 네 개의 탐색 단추를 사용하면 기준선과 실제 파일들 간의 차이점을 찾는 데 도움이 됩니다.

다음 두 가지 방식으로 기준선 파일을 갱신할 수 있습니다. 눈금의 해당 셀을 편집하여 단어 "Order"에 문자 's'를 추가하거나 기준선 바꾸기 명령을 사용할 수 있습니다. 기준선 바꾸기는 기준선 파일의 모든 값을 실제 파일의 값으로 대체합니다. 일반적으로 하나 또는 일부 값만 편집해야 하는 경우에는 각각의 값을 개별적으로 편집해야 합니다.

9. 이 테스트에는 갱신할 차이점이 하나만 있으므로 비교기 도구 모음에서 실제 값으로 기준선 바꾸기 단추()를 클릭하십시오. 이제 텍스트 특성의 두 값이 일치하며 특성이 더 이상 빨간색으로 표시되지 않습니다. 비교기 사용에 대한 자세한 정보는 Functional Tester 도움말을 참조하십시오.
10. 비교기를 닫으십시오.

이제 실패에 대한 기준선 값을 갱신했으므로 스크립트를 다시 재생하여 검증 포인트가 패스되는지 확인하십시오.

11. 로그를 닫으십시오.
12. Functional Tester 도구 모음에서 **Functional Test** 스크립트 실행 단추를 클릭하십시오.
13. **Classics** 로그를 선택한 다음 완료를 클릭하십시오.
14. 로그를 겹쳐줄 것인지 묻는 메시지가 프롬프트되면 예를 클릭하십시오.

아직 해당 인식 문제점을 정정하지 않았으므로 Functional Tester가 구성원 로그온 창에서 일시정지됩니다. 재생이 종료될 때 Functional Tester가 로그를 표시합니다. 검증 포인트가 이제 패스됩니다. 이처럼 간단하게 비교기를 사용하여 오브젝트 데이터 및 특성을 갱신함으로써 테스트 중인 응용프로그램의 변경 사항에 대해 설명할 수 있습니다.

15. 로그를 열어 두십시오.

연습 8: 오브젝트 맵 갱신

이 연습에서는 오브젝트 맵을 사용하여 오브젝트 인식 경고를 수정합니다. 또한 보다 유연성 있는 오브젝트 인식을 위해 정규식도 사용합니다.

인식 실패 또는 경고가 발생하면 로그 메시지를 참조하십시오. 연습 7의 마지막에서 로그를 열어 두었습니다. 열려 있지 않은 경우, 프로젝트 보기에서 로그를 두 번 클릭하여 이를 여십시오. 한 개의 경고가 로그에 남아 있습니다. 이벤트 표제는 오브젝트 인식이 약함(경고 임계값 초과)입니다.

1. 로그 맨 아래 부분의 경고 섹션에서 **ObjectLookedFor** 및 **objectFound** 필드를 확인하십시오.

ClassicsA에서 암호 필드의 이름은 **Remember Password**입니다. ClassicsB에서 암호 필드의 이름은 **Remember The Password**입니다. ClassicsB에서 스크립트를 재생한 경우, 오브젝트 인식은 이러한 차이점으로 인해 정확하게 일치하지 않습니다.

2. 로그의 행 번호 필드를 확인하십시오. 번호를 확인하고 로그를 닫아 Functional Tester로 리턴합니다.
3. 스크립트 창에서 임의의 위치를 클릭한 다음 탐색 → 행으로 이동을 클릭하십시오.
4. 로그 실패 메시지에서의 행 번호를 입력하고 확인을 클릭하십시오.

커서가 해당 행 번호의 왼쪽 여백으로 이동합니다.

주: Functional Tester 창 맨 아래의 표시기를 확인하여 행 번호를 찾을 수도 있습니다(예: "43:9"는 43행 9 열의 위치를 나타냄).

스크립트의 행은 다음과 같아야 합니다.

```
RememberPassword().clickToState(SELECTED);
```

이 행은 암호 선택란에서의 클릭을 표시합니다. 스크립트의 이 행은 실패하는 오브젝트를 표시합니다. 이제 오브젝트 맵에서 해당 오브젝트를 확인할 수 있습니다.

5. 오브젝트를 찾으려면 스크립트 탐색기(오른쪽 분할창)에서 테스트 오브젝트의 목록으로 돌아가십시오. rememberPassword가 테스트 오브젝트 폴더 아래에 나열되어 있어야 합니다.

오브젝트 맵에서 오브젝트 인식 특성 보기

1. **rememberPassword** 오브젝트를 두 번 클릭하여 이를 오브젝트 맵에서 여십시오.
2. 오브젝트 맵 메뉴에서 테스트 오브젝트 → 모두 승인을 클릭하십시오. 명령이 회색으로 나타나면 작업을 수행하지 마십시오.

모든 오브젝트가 검은색 텍스트로 변경됩니다. 맵의 오브젝트를 승인할 때까지 텍스트는 파란색(새 오브젝트를 표시하기 위해)입니다. 새로 작성된 오브젝트 맵을 처음으로 발견하는 시점에 오브젝트를 승인해야 합니다.

3. 암호 선택란 오브젝트가 맵에서 선택되지 않으면 이를 선택하십시오(**Java: checkBox: checkRemember: javax.swing.JCheckBox** 오브젝트).
4. 오브젝트 맵 맨 아래의 인식 탭에 나열된 인식 특성을 확인하십시오.

text 특성에서 Remember Password로 제시되므로 이 오브젝트가 ClassicsA의 오브젝트임을 알 수 있습니다. 이 오브젝트는 "이전" 오브젝트입니다. 그러나, ClassicsB에서 스크립트를 재생한 경우에는 해당 오브젝트에 대한 텍스트가 변경되므로 Functional Tester가 이를 "새" 오브젝트로 인식합니다. 이 경우에 새 오브젝트 특성을 사용하려면 이를 맵에 추가해야 합니다.

맵에 새 오브젝트 추가

맵에 새 오브젝트를 추가하려면 ClassicsB 및 구성원 로그인 창을 여십시오.

1. 오브젝트 맵 메뉴에서 응용프로그램 → 실행을 클릭하십시오.
2. **Classics.JavaB**를 선택하십시오(반드시 B를 선택하십시오).
3. 확인을 클릭하십시오.
4. ClassicsCD에서 임의의 CD를 선택한 다음 주문하기를 클릭하십시오.

구성원 로그인 창이 열립니다.

5. 필요한 경우, 화면에서 아래로 오브젝트 맵을 이동하여 전체를 확인할 수 있습니다. 오브젝트 맵 메뉴에서 **오브젝트 테스트** → **오브젝트 삽입**을 클릭하십시오.

이는 검증 포인트 마법사의 오브젝트 선택 페이지에서 오브젝트 파인더 도구와 동일합니다.

6. 오브젝트를 선택하면 다음 페이지로 진행 선택란이 선택되어 있는 경우 이를 지우십시오.
7. 오브젝트 파인더 도구를 사용하여 구성원 로그인 창에서 암호 기억 선택란을 선택하십시오.

선택란을 선택하면 **text** 특성이 이제 Remember The Password로 나타납니다. 필요한 경우, 오브젝트 맵의 경계를 확장하여 특성을 확인할 수 있습니다.

8. 오브젝트 선택 페이지에서 다음을 클릭하십시오.
9. 오브젝트 옵션 선택 페이지에서 변경을 수행하지 않고 완료를 클릭하십시오.

새 선택란 오브젝트가 이제 오브젝트 맵에 표시됩니다.

10. 다른 오브젝트를 클릭하고 새 항목이 파란색으로 나열되며 단어 "새로 작성"이 행 맨 앞에 나타나는지 확인하십시오.

이제 새 오브젝트 및 이전 오브젝트가 모두 맵에 나열됩니다. 두 오브젝트를 통합하고 새 오브젝트에 대해 각각의 오브젝트에서 특성을 가져오하고자 합니다.

오브젝트 통합

1. 오브젝트를 통합하려면 이전 오브젝트(원래 선택란의 레이블은 **CheckBox: checkRemember**임)를 클릭한 다음 이를 목록의 새 오브젝트로 끌어 놓으십시오. 마우스 단추를 해제하기 전에 새 오브젝트 위에 커서 화살표의 팁을 위치시키십시오. 그런 다음, 마우스 단추를 해제하십시오.

테스트 오브젝트 통합 마법사가 열립니다.

2. 필요한 경우, 통합 마법사를 확장하여 하단 섹션에서 추가 정보를 확인할 수 있습니다.

하단 왼쪽 섹션에서 원래 오브젝트의 특성이 표시됩니다. 레이블은 "Source: RememberPassword"이어야 합니다. 이는 ClassicsA의 선택란에 있던 텍스트와 동일합니다. 하단 오른쪽 섹션에서 레이블은 "Target: RememberThePassword"이어야 합니다. 이는 ClassicsB의 선택란에 있던 텍스트와 동일합니다.

이전 오브젝트를 새 오브젝트로 끌어 놓았으므로 새 오브젝트의 인식 특성이 마법사의 맨 위에서 채워집니다. 일반적으로 새 특성이 선호되는 경우에 Functional Tester는 새 특성을 맨 위에 둡니다. 그러나, 일부 이전 관리 특성이 선호될 수도 있습니다. 예를 들어, Functional Tester는 이전 특성 세트에 정규식을 보관합니다. 이전 오브젝트의 특성을 사용하려면 이전 오브젝트의 눈금에서 해당 특성을 두 번 클릭하십시오. 특성이 통합된 오브젝트로 복사됩니다. 이 경우에는 이미 채워져 있는 새 오브젝트의 모든 특성을 사용하고자 합니다.

3. 다음을 클릭하십시오.

이 변경사항에 의해 영향을 받는 모든 스크립트가 오브젝트 맵에 나열됩니다. Classics 스크립트만이 영향을 받습니다.

4. 완료를 클릭하십시오.

5. 오브젝트 맵에서 오브젝트 맵 도구 모음의 파일 → 저장 메뉴를 클릭하여 작성한 변경사항을 저장하고 오브젝트 맵을 닫으십시오.

스크립트 다시 재생

이제 ClassicsB에서 스크립트를 다시 재생하여 스크립트가 패스되는지 확인합니다.

1. ClassicsCD의 대화 상자를 모두 닫으십시오.

2. Functional Tester에서 도구 모음의 **Functional Test** 스크립트 실행을 클릭하십시오.

3. **Classics** 로그를 선택한 다음 완료를 클릭하십시오.

스크립트가 이제 경고 없이 패스됩니다! 이제 인식 특성이 일치하므로 더 이상 암호 선택란 오브젝트에서 재생이 일시정지되지 않습니다.

오브젝트의 인식 특성이 의도적으로 변경되는 경우에는 이 오브젝트 통합 기능을 사용하여 쉽게 스크립트를 갱신할 수 있습니다. 이 기능의 주요 장점 중 하나는 오브젝트 맵이 다수의 스크립트에 의해 사용되는 경우, 마법사에서 변경을 수행할 때 이들 모두를 갱신할 수 있다는 점입니다. 다수의 스크립트를 수동으로 편집하는 대신, 맵에서 한 번의 변경을 수행할 수 있으며 변경사항은 이를 사용하는 모든 스크립트에 자동으로 전파됩니다. 이 기능을 사용하면 시간이 절약됩니다.

인식 특성을 갱신하는 다른 방법: 변경이 필요한 경우에 테스트 오브젝트의 인식 특성을 갱신하는 보다 쉬운 방법이 있습니다. 이 연습에서 설명한 대로 통합 마법사를 사용하는 대신에 해당 인식 특성을 갱신하고자 하는 테스트 오브젝트를 오브젝트 맵에서 선택할 수 있습니다. 오브젝트 맵 트리에 테스트 오브젝트에 표시되면 이를 마우스 오른쪽 단추로 클릭하고 팝업 메뉴에서 인식 특성 갱신을 선택하십시오. 이 조치를 실행할 때에는 Functional Tester가 갱신된 인식 특성을 가져올 수 있도록 테스트 응용프로그램이 실행 중이어야 합니다. 이전 오브젝트의 임의의 특성을 사용하지 않으려면 이 갱신 메소드만 사용하십시오.

4. 로그를 닫으십시오.

연습 9: 인식 환경 설정 변경

이전 연습에서 변경이 수행될 때 오브젝트의 인식 특성을 갱신하는 방법을 학습했습니다. 재생 중에 Functional Tester가 사용하는 인식 가중치 또한 변경할 수 있습니다. ScriptAssure™ 인식 환경 설정을 사용하여 이를 설정할 수 있습니다. 두 번째 검증 포인트로 테스트한 레이블 오브젝트에서 작동 방법을 확인할 수 있습니다.

1. Functional Tester 메뉴에서 창 → 환경 설정을 클릭하십시오.
2. **Functional Test** → 재생 → **ScriptAssure**를 클릭하십시오.
3. 고급 단추를 클릭하십시오.

기본 설정 중에 승인된 점수가 **10000**을 초과하면 경고를 확인하십시오. 점수 10000은 중요한 특성에 오류가 있음을 표시합니다. 점수를 5000으로 낮추고 발생 내역을 참조하십시오.

4. 이 필드 옆의 기본값 사용 선택란을 선택하십시오.
5. 그리고 필드에서 5000을 입력한 다음 확인을 클릭하십시오.
6. ClassicsB에서 스크립트를 다시 재생하십시오.

이제 로그에 레이블 오브젝트에 대한 경고가 포함됩니다. **objectFound** 필드에 부여된 이유는 인식 점수가 10000이기 때문입니다. 이러한 불일치는 레이블에서 단어 "Order"를 "Orders"로 변경하여 발생한 것입니다.

7. 로그를 닫으십시오.
8. 인식 점수에 대한 기본값을 복원하십시오.
 - a. 창 → 환경 설정을 클릭하십시오.
 - b. **Functional Test** → 재생 → **ScriptAssure**를 클릭하십시오.
 - c. 고급 단추를 클릭하십시오.
 - d. 승인된 점수가 ...이면 경고 필드 옆의 기본값 사용 선택란을 선택하십시오.

5000이 다시 10000으로 변경됩니다.

- e. 확인을 클릭하십시오.
- f. 스크립트를 다시 재생하십시오.

이제 경고가 사라지고 모두 패스됩니다.

- g. 로그를 닫으십시오.

이 연습에서는 오브젝트 인식에 대해 원하는 민감도를 얻기 위해 인식 점수를 변형하는 방법을 학습했습니다. ScriptAssure 사용에 대한 자세한 정보는 Functional Tester 도움말을 참조하십시오.

연습 10: 정규식 사용

오브젝트 맵을 사용하여 수행하는 마지막 작업은 특성 값을 정규식으로 변환하는 것입니다. 이 경우에 정규식은 오브젝트 인식에서 보다 많은 유연성을 제공합니다.

스크립트가 ClassicsB에서 패스되는 방법을 확인했습니다. ClassicsB에서 응용프로그램에 대한 변경사항이 올바르게 적용되었습니다. 따라서 스크립트를 다음 단계로 진행할 수 있습니다. 이제 ClassicsA에 대해 스크립트를 재생하면 이전의 변경사항으로 인해 재생에 실패하게 됩니다. 둘 이상의 오브젝트에 대한 변형을 패스하려 합니다. 동적 오브젝트를 보유하거나 오브젝트의 버전이 약간 다른 응용프로그램의 다수 버전을 보유할 수 있으며, 이 경우에는 둘 모두 올바릅니다. 정규식을 사용하여 텍스트와 같은 특성 값의 둘 이상의 버전이 이 시나리오를 수용하도록 할 수 있습니다.

오브젝트 맵 열기 및 오브젝트 통합

1. ClassicsA에 대해 재생하려면 스크립트 맨 위에서 `startApp` 명령을 편집하고 B를 A로 변경하십시오.
2. Functional Test 도구 모음에서 **Functional Test** 스크립트 실행을 클릭하십시오. 재생 중에 Functional Tester는 암호 선택란 오브젝트에서 잠시 일시정지하지만 결국에는 완료됩니다. 이제 스크립트에서 경기가 발생합니다. 로그에서 **rememberPassword** 테스트 오브젝트와 동일한 오브젝트입니다.
3. 로그를 닫은 다음 스크립트 탐색기의 **rememberPassword**를 두 번 클릭하여 연습 8에서 수행한 대로 암호 선택란 오브젝트에서 오브젝트 맵을 여십시오.
4. 오브젝트 맵에서 **응용프로그램** → **실행**을 클릭하여 응용프로그램을 여십시오. **ClassicsJavaA**를 선택한 다음 **확인**을 클릭하십시오.
5. 임의의 CD를 선택하고 ClassicsCD에서 **주문하기**를 클릭하여 구성원 로그인 창을 여십시오.
6. **테스트 오브젝트** → **오브젝트 삽입**을 클릭하여 새 오브젝트를 맵에 추가하십시오.
7. 오브젝트 파인더를 사용하여 응용프로그램의 구성원 로그인 창에서 암호 선택란을 선택하십시오.
8. 다음을 클릭한 다음 **완료**를 클릭하십시오.
9. 오브젝트 맵의 맨 위 분할창에서 이전의 선택란 오브젝트를 새 선택란 오브젝트로 끌어 놓아 오브젝트를 통합하십시오.
10. 필요한 경우, 필드의 한 쪽 면을 바깥쪽으로 끌어 필드 길이를 늘림으로써 테스트 오브젝트 통합 마법사를 확장할 수 있습니다.

두 개의 서로 다른 정규식(이름 특성의 정규식 및 **accessibleName** 특성의 정규식)을 사용합니다.

통합된 오브젝트가 통합된 테스트 오브젝트 특성 눈금(맨 위 분할창)에 표시됩니다. 이름 특성의 값은 `checkRemember`입니다.

특성 값을 정규식으로 변환

1. 맨 위 분할창에서 `checkRemember` 값을 마우스 오른쪽 단추로 클릭한 다음 **정규식으로 값 변환**을 클릭하십시오.

Functional Tester는 값 텍스트의 앞에 있는 "xy" 아이콘을 사용하여 값을 정규식으로 지정합니다.

2. 이름 값을 다시 두 번 클릭하면 필드를 편집할 수 있습니다.
3. `check`를 삭제하고 다음과 같이 나머지를 편집하십시오(`[rR]emember`).
4. 셀의 외부를 클릭하십시오.

이 패턴은 대문자 "R" 또는 소문자 "r"을 구분하여 "remember"가 패스되도록 허용합니다. 대소문자를 구분하여 비교하므로 정확히 일치해야만 패스되므로 이는 중요합니다. `accessibleContext.accessibleName` 특성의 값은 "Remember Password"입니다.

5. "Remember Password" 값을 마우스 오른쪽 단추로 클릭한 다음 정규식으로 값 변환을 선택하여 이를 변환하십시오.
6. 값을 두 번 클릭하고 다음과 같이 나머지를 편집하십시오(Remember.*Password). 스페이스를 제거하며 마침표(.) 및 별표(*) 문자를 추가합니다.
7. 다른 셀을 클릭하십시오.

".\" 를 사용하여 해당 위치에 임의의 문자를 표시할 수 있습니다. 응용프로그램의 한 버전에서는 이 특성의 두 단어 사이에 공백이 있고 다른 버전에서는 공백이 없습니다. 이 패턴은 두 가지 경우를 모두 포함합니다.

8. 다음을 클릭한 다음 완료를 클릭하십시오.
9. 오브젝트 맵에서 파일 → 저장을 클릭하여 변경사항을 저장한 다음 오브젝트 맵을 닫으십시오.
10. ClassicsCD를 닫으십시오.
11. ClassicsA에서 스크립트를 다시 재생하십시오. 이 경우에 Orders for Trent Culpito 텍스트가 정규식으로 변경되지 않았기 때문에 검증 포인트가 실패하게 됩니다. ClassicsA에서의 오브젝트 인식 경고가 더 이상 로그에 표시되지 않습니다.
12. 로그를 닫으십시오.
13. startApp 명령을 변경하여 ClassicsB를 재생한 다음 스크립트를 실행하십시오.

오브젝트 인식도 역시 ClassicsB에서 패스됩니다. 정규식은 서로 다른 버전의 응용프로그램에서 다른 특성을 보유하는 오브젝트에 대해 보다 유연성 있는 인식을 제공하며 둘 모두 재생 중에 인식됩니다. 정규식에 대한 자세한 정보는 Functional Tester 도움말을 참조하십시오.

요약: Functional Test 작성

이 Functional Tester 학습서에서는 스크립트를 테스트, 레코딩 및 재생하고 검증 포인트를 작성하며 검증 포인트 비교기를 사용하여 오브젝트 특성 또는 데이터를 갱신하는 방법을 비롯하여 사용자에게 유용하도록 오브젝트 맵을 사용하는 몇 가지 방법을 살펴보았습니다.

학습 내용

이 학습서를 통해 다음을 수행하는 방법을 학습했습니다.

- Functional Tester 프로젝트 작성
- 테스트 응용프로그램의 조치에 대해 스크립트 레코딩
- 레코딩 중 적절하게 테스트 응용프로그램 시작
- 검증 포인트 작성
- 스크립트 재생

- Functional Tester 로그 사용
- 비교기를 사용하여 검증 포인트 갱신
- 오브젝트 맵 갱신
- 오브젝트에 대한 인식 환경 설정 변경
- 오브젝트 인식에서의 추가 유연성을 위한 정규식 사용

추가 자원

이 학습서에서 다른 주제에 대한 추가 학습이 필요한 경우, 다음 자원을 참고하십시오.

- Functional Tester 도움말
- Functional Tester API 참조
- Functional Tester 환영 페이지

관련 정보

ibm.com

eclipse.org