

# Ranorex

## インストール&チュートリアル

評価版用

Copyright © 2015 Ranorex GmbH

Last updated: 2017.2

Ranorex version: 6.1

# 目次

<b>目次</b>	<b>2</b>
<b>はじめに</b>	<b>4</b>
本書について .....	4
チュートリアルについて .....	4
<b>インストールとセットアップ</b>	<b>5</b>
稼動環境 .....	5
サポート OS .....	5
必要なソフトウェア .....	5
必要なハードウェア .....	6
必要なディスク容量 .....	6
Ranorex License Manager に必要なファイアウォール設定 .....	6
Ranorex Agent に必要なファイアウォール設定 .....	6
インストール .....	7
ライセンスのセットアップ .....	10
アンインストール .....	10
<b>Ranorex Studio の画面構成</b>	<b>11</b>
スタートページ .....	11
サンプルプロジェクト .....	12
Desktop サンプル .....	13
Web サンプル .....	13
Mobile サンプル (Android, iOS) .....	13
Cross-Platform サンプル .....	13
Ranorex の画面構成 .....	14
#1 – プロジェクトビュー .....	14
#2 – モジュールブラウザー .....	15
#3 – ファイルビュー .....	15
<b>レッスン 1: はじめに</b>	<b>19</b>
ユーザーアクションのレコーディングの準備 .....	19
テスト対象システム：KeePass アプリケーション .....	20
テストのレコーディング .....	21
Ranorex Studio の起動と空のレコーディングファイルのオープン .....	21
KeePass アプリケーションのテストのレコーディングの開始 .....	22
初期パスワードの入力 .....	24
“Add Entry”フォームのオープン .....	26
新規エントリーの追加 .....	26
結果の検証 .....	28

追加したエントリーの削除とアプリケーションの終了 .....	30
レコードした操作の確認 .....	30
テストの実行 .....	31
レポート .....	32
<b>レッスン 2: Ranorex モジュール - テストアクション</b> .....	<b>35</b>
リファクタリング: レコーディングを分割する必要性 .....	35
モジュールの特定 .....	35
<b>レッスン 3: データ駆動テスト</b> .....	<b>39</b>
レコーディングでの変数の利用 .....	39
リポジトリ内での変数の使用 .....	40
テストデータの作成 .....	43
テストデータと変数のバインド .....	45
パラメーターと変数のバインド .....	47
選択アクションの実行 .....	49
データ駆動テストの実行 .....	52
<b>おわりに</b> .....	<b>54</b>
Ranorex に関するその他の情報 .....	54

# はじめに

## 本書について

---

本書では、Ranorex 評価版のインストール・セットアップ方法と、簡易チュートリアルについて説明します。

## チュートリアルについて

---

本書のチュートリアルでは、簡単な Windows アプリケーションについて、ユーザーアクションのキャプチャによる機能テストの実施方法について学習します。

Windows アプリケーションのテストに関する詳細、Web アプリケーション、および、モバイルアプリケーションのテストの実施方法については、別紙『Ranorex ユーザーガイド』を参照してください。

# インストールとセットアップ

この章では、Ranorex のインストールとセットアップ方法について説明します。

## 稼動環境

---

### サポート OS

Ranorex は、以下の OS (32bit/64bit いずれも。Itanium プロセッサは除く) をサポートしています。

- Windows Server 2008 2008 R2
- Windows 7
- Windows 8 / 8.1
- Windows 10
- Windows Server 2012 / 2012 R2

### 必要なソフトウェア

Ranorex コンポーネント (Runtime, Spy, Recorder, License Manager, Agent) の動作には、以下のソフトウェアが必要です。通常、これらのソフトウェアは Ranorex のインストーラーによって自動でインストールされます。

- Microsoft Windows Installer 3.1 以上
- Microsoft Visual C++ 2008 x86
- Microsoft Visual C++ 2008 x64 (64 bit Windows の場合のみ)
- Microsoft Visual C++ 2010 x86
- Microsoft Visual C++ 2010 x64 (64 bit Windows の場合のみ)
- Microsoft Visual C++ 2015 x86
- Microsoft .NET Framework 4.0 以上
- Internet Explorer 7 以上 (最新のアップデートバージョン)

## 必要なハードウェア

必要なハードウェアスペックは、各 Ranorex コンポーネントが使用する .NET Framework のバージョンに依存します。

Ranorex Runtime, Spy, Recorder, License Manager, Studio, Agent の動作には、.NET Framework 4.0 以上のインストールが必要です。

Ranorex Runtime, Spy, Recorder, License Manager, Studio, Agent の動作に必要な最小ハードウェアスペックは、.NET Framework 4.0 の動作環境に準じます。

[www.microsoft.com/en-US/download/details.aspx?id=17718](http://www.microsoft.com/en-US/download/details.aspx?id=17718)

推奨する最小ハードウェアスペックは、以下の通りです。

- プロセッサ : 2GHz
- メモリー : 512MB

## 必要なディスク容量

完全インストール (Ranorex Studio, main components, documentation, etc.) の場合、約 250MB のディスク容量が必要です。最小インストール (Ranorex runtime environment) の場合、約 90MB のディスク容量が必要です。

Ranorex Agent のインストールには、約 25MB のディスク容量が必要です。また、Ranorex Agent の実行には、ランタイムライセンスが必要です。

## Ranorex License Manager に必要なファイアウォール設定

Ranorex License Manager は、TCP および UDP の 7266 番を使用します。ファイアウォールが存在する場合、これらの通信を許可するよう設定してください。

## Ranorex Agent に必要なファイアウォール設定

Ranorex Agent は、TCP の 8081, 29189 番、および、UDP の 10000-10001 番を使用します。ファイアウォールが存在する場合、これらの通信を許可するよう設定してください。

## インストール

1. ダウンロードしたインストーラー (**Ranorex-6.x.x.exe**) をダブルクリックします。セットアップウィザードが起動します。



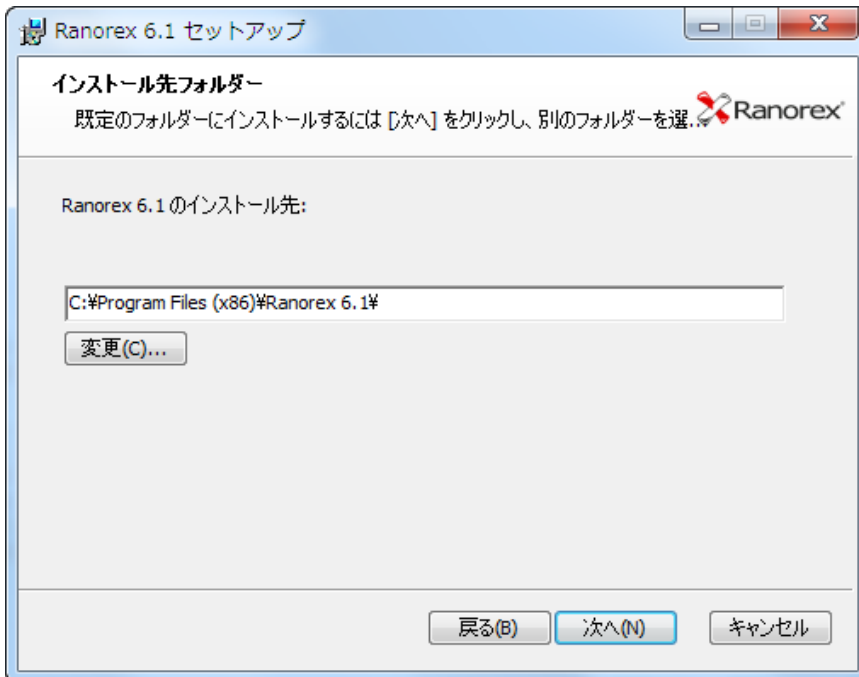
[次へ]ボタンをクリックします。

2. End-User License Agreement (ソフトウェア利用許諾契約) が表示されます。

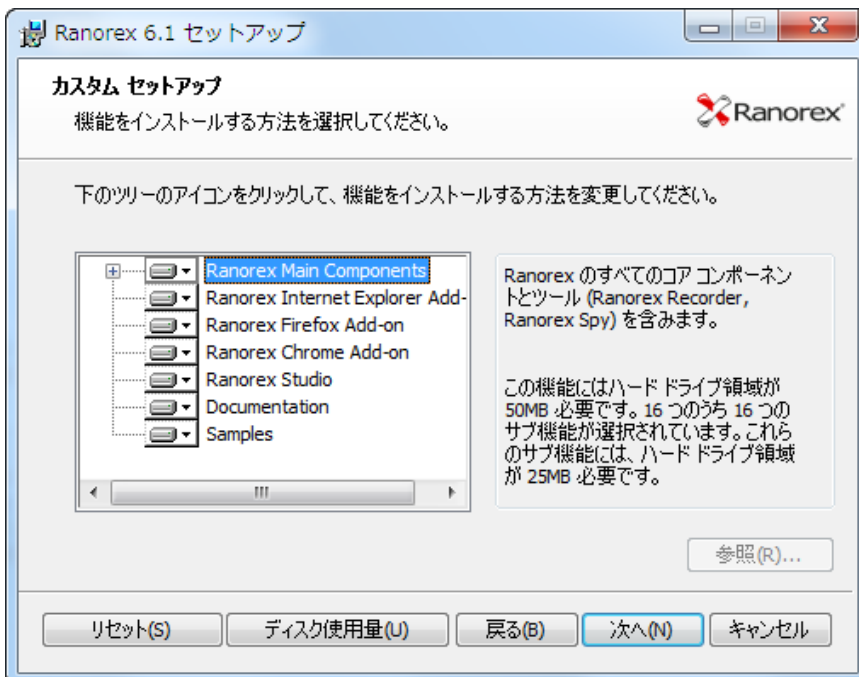


インストールを続行するためには、利用許諾契約への同意が必要です。内容を確認の上、同意する場合には、“使用許諾契約書に同意します”にチェックを入れ、[次へ]ボタンをクリックします。

3. インストールフォルダーを指定し、[次へ]ボタンをクリックします。

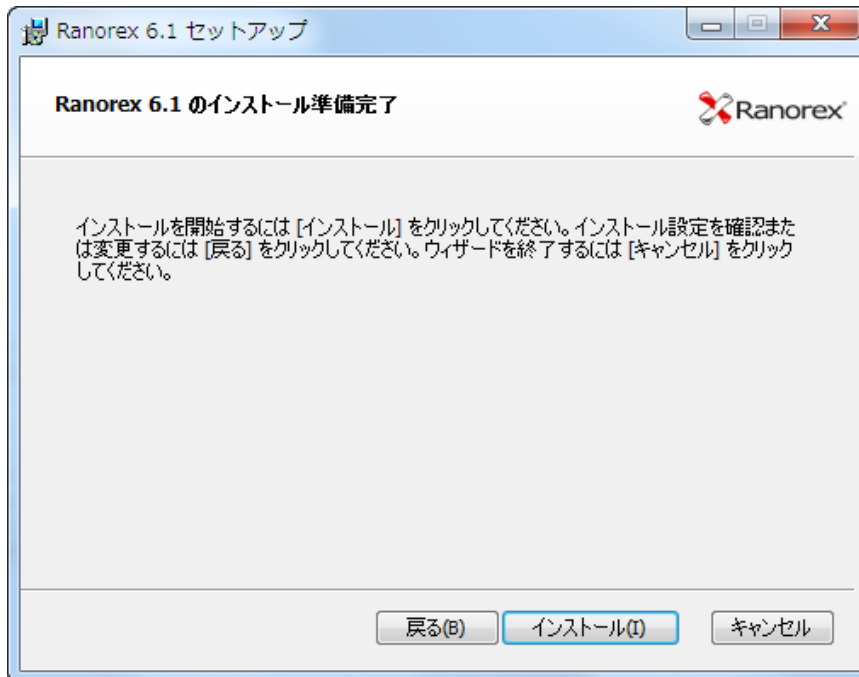


4. インストールするコンポーネントを指定し、[次へ]ボタンをクリックします。





5. [インストール]ボタンをクリックすると、インストールが開始されます。



6. インストールが完了すると以下の画面が表示されます。



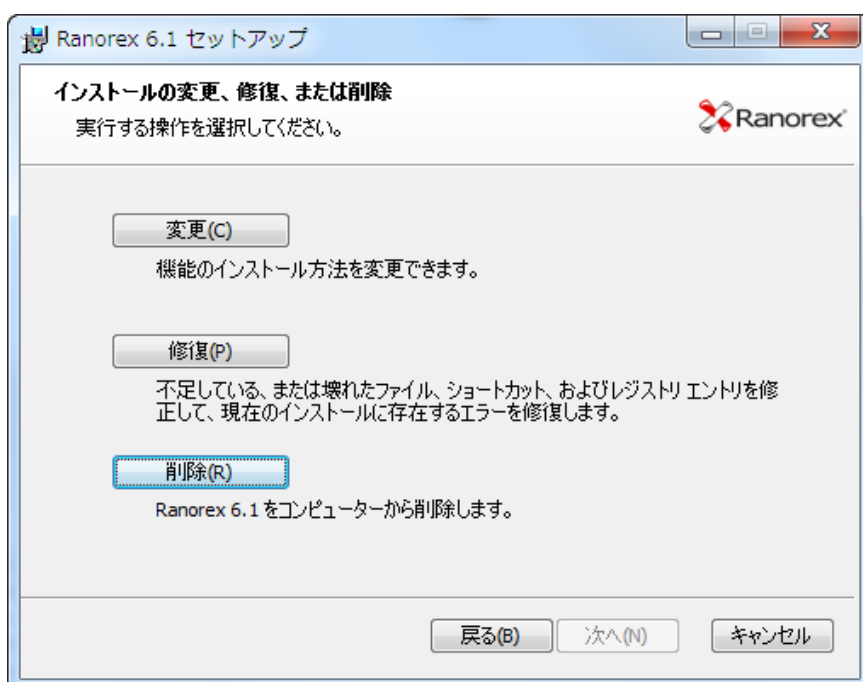
“Ranorex Studio を起動する”をチェックしていると、[完了]ボタンをクリックしてセットアップウィザードを終了したあとに、Ranorex Studio が起動します。

## ライセンスのセットアップ

評価版では、インストール直後から、30日間利用可能なライセンスが自動で有効になります。製品版でのライセンスのセットアップについては、別紙『Ranorex インストール&セットアップ』を参照してください。

## アンインストール

1. インストーラー (**Ranorex-6.x.x.exe**) をダブルクリックし、セットアップウィザードを起動します。[削除]ボタンをクリックします。



2. 確認画面で、[削除]ボタンをクリックすると、アンインストールがおこなわれます。



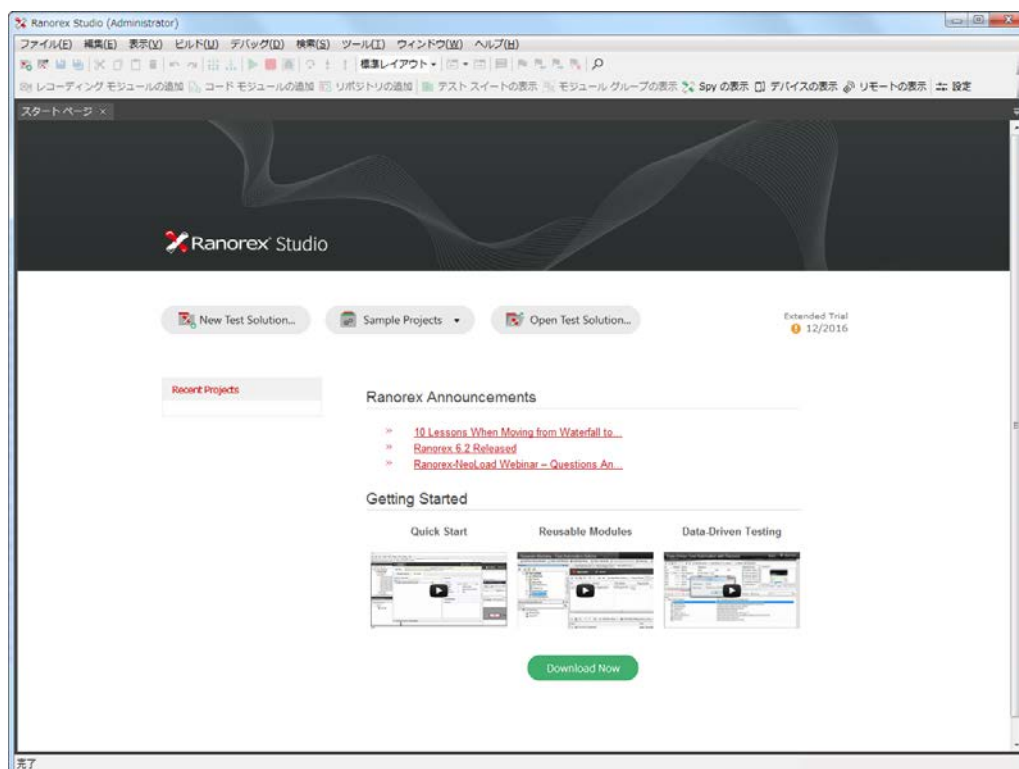
# Ranorex Studio の画面構成

この章では、Ranorex Studio の主なコンポーネントについて説明します。

## スタートページ

Ranorex Studio を起動すると、最初にスタートページが表示されます。スタートページからは、以下がおこなえます。

- 最近使用したプロジェクトのオープン
- 新規テストソリューションの作成
- サンプルプロジェクトのオープン
- 既存のソリューションのオープン
- インストールされているライセンスの詳細情報の確認
- ドキュメントとフォーラムの検索
- 最新バージョンのダウンロード
- 最新ニュースの閲覧
- 最新の Blog 投稿の閲覧

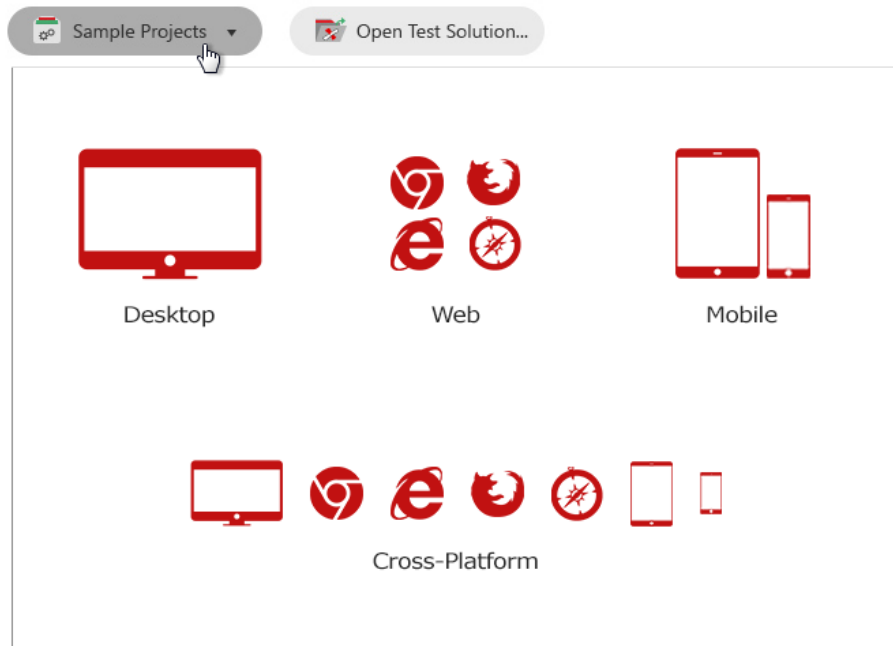


Ranorex スタートページ

## サンプルプロジェクト

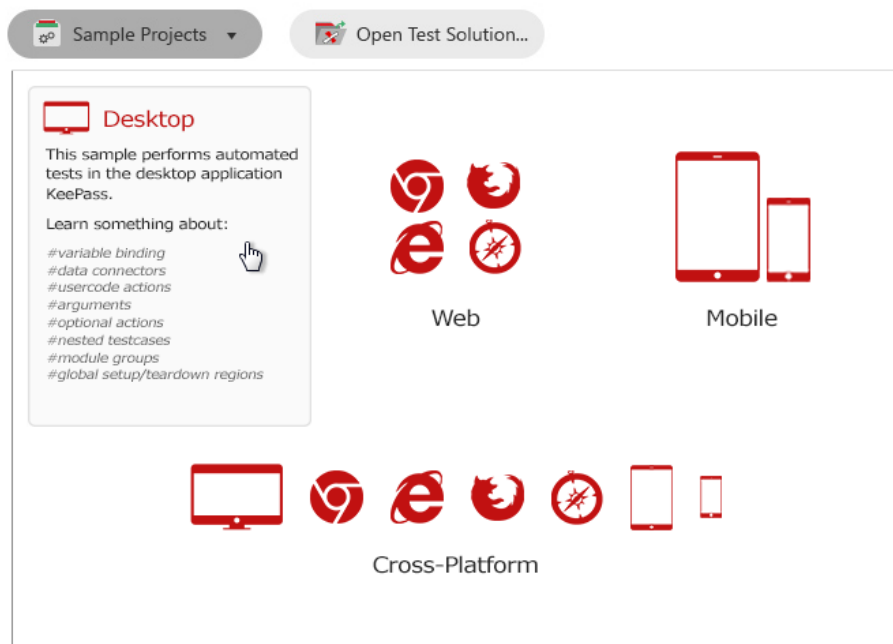
Ranorex Studio では、スタートページから、いくつかのサンプルプロジェクトを利用できます。これらのサンプルプロジェクトを利用することで、デスクトップ/Web/モバイルの各アプリケーションの自動テストの作成方法について学習することができます。さらに、一つのテストシナリオですべてのプラットフォーム上でテスト可能な、クロスプラットフォームサンプルも利用できます。

これらのサンプルプロジェクトにアクセスするには、[Samples]ボタンを選択してください。以下のように、各サンプルプロジェクトへのリンクが表示されます。



サンプルプロジェクトへのアクセス

Desktop, Web, Mobile, Cross-Platform いずれかを選択することで、詳細な説明が表示されます。



## Desktop サンプル

デスクトップアプリケーション「KeePass」の自動化テストのサンプルプロジェクトです。

学習キーワード: #variable binding #data connectors #usercode actions #arguments  
#optional actions #nested test cases #module groups #global setup/teardown  
regions



Desktop

## Web サンプル

WordPress サイトの自動化テストのサンプルプロジェクトです。

学習キーワード: #module library #popup watcher #project organization #report  
screenshot #validation #web testing #handling flyout menu #setup/teardown  
regions #data binding



Web

## Mobile サンプル (Android, iOS)

モバイルアプリケーション「KeePass」の自動化テストのサンプルプロジェクトです。

学習キーワード: #mobile testing #android #ios #deploy mobile app #module  
groups #usercode actions #setup/teardown regions #data connectors #arguments



Mobile

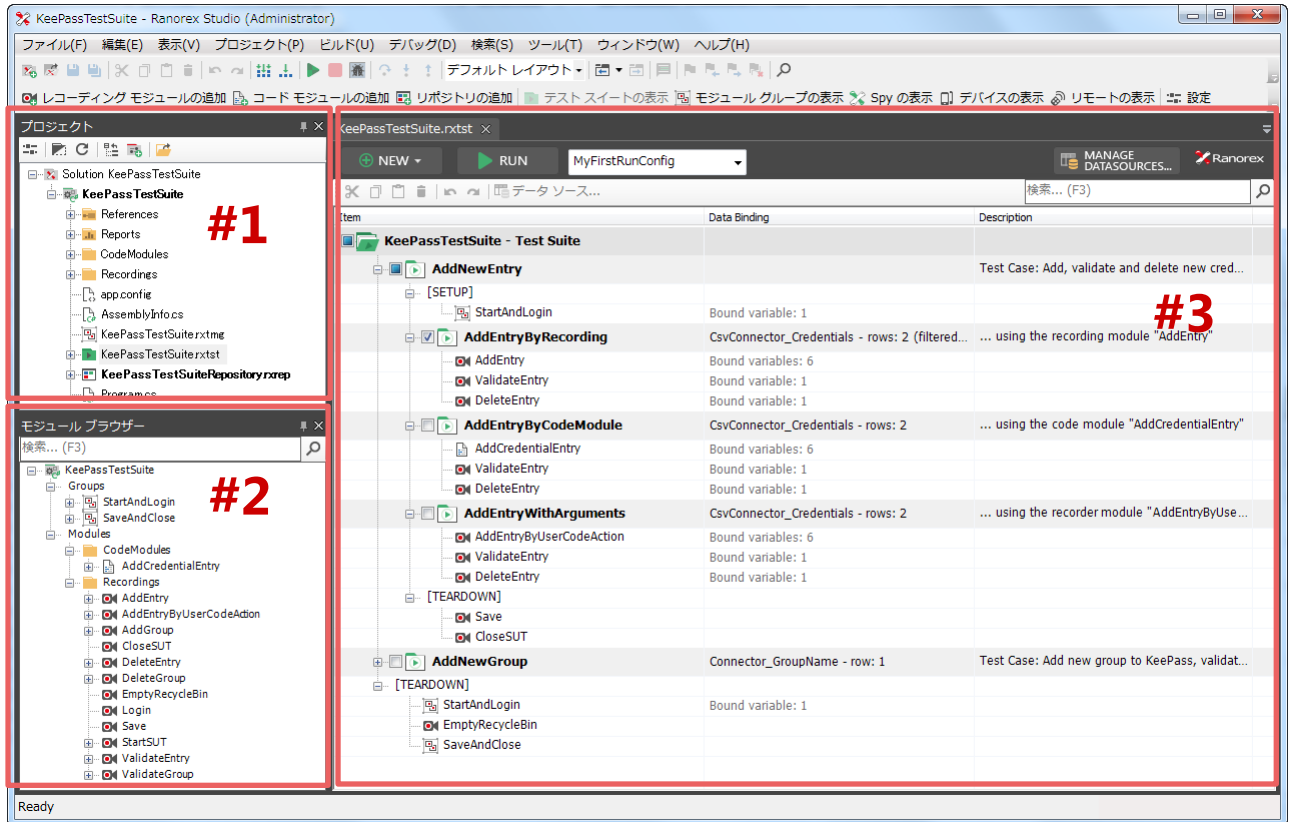
## Cross-Platform サンプル

Desktop, Web, Mobile アプリケーションの自動化テストのサンプルプロジェクトです。

学習キーワード: #cross-platform test #end2end test #ios #android #deploy mobile app #module  
library #project organization #popup watcher #web testing #mobile web testing #setup/teardown regions  
#data binding

## Ranorex の画面構成

Ranorex Studio で最初のテストケースを作成する前に、Ranorex Studio の基本的な画面構成について説明します。



Ranorex Studio メインビュー



Ranorex の概要については、以下の動画『Quick Start with Ranorex Test Automation』もご覧ください。

[youtu.be/0S\\_YC7uWI-s](https://youtu.be/0S_YC7uWI-s)

### #1 – プロジェクトビュー

Ranorex Studio プロジェクトは、Microsoft Visual Studio 2008/2010 プロジェクトをベースとしたファイルで構成されます。プロジェクトビューは、プロジェクトに関連付けられているすべてのファイルと参照を表示します。Ranorex Studio プロジェクトには、以下の種類のファイルが含まれます。

テストスイート	テストスイート (*.rxtst)
モジュールグループ	モジュールグループ (*.rxtmg)
リポトリ	UI 要素の管理 (*.rxrep)
レコーディング	キャプチャ/リプレイベースのオートメーションモジュール (*.rxrec)
コードファイル	コードベースのオートメーションモジュールの作成に使用される C#,VB.NET コード

プロジェクトビューは主に、Ranorex レコーダー、Ranorex リポジトリ、モジュールグループ、コードモジュールといった、新たなアイテムを追加する際に使用されます。

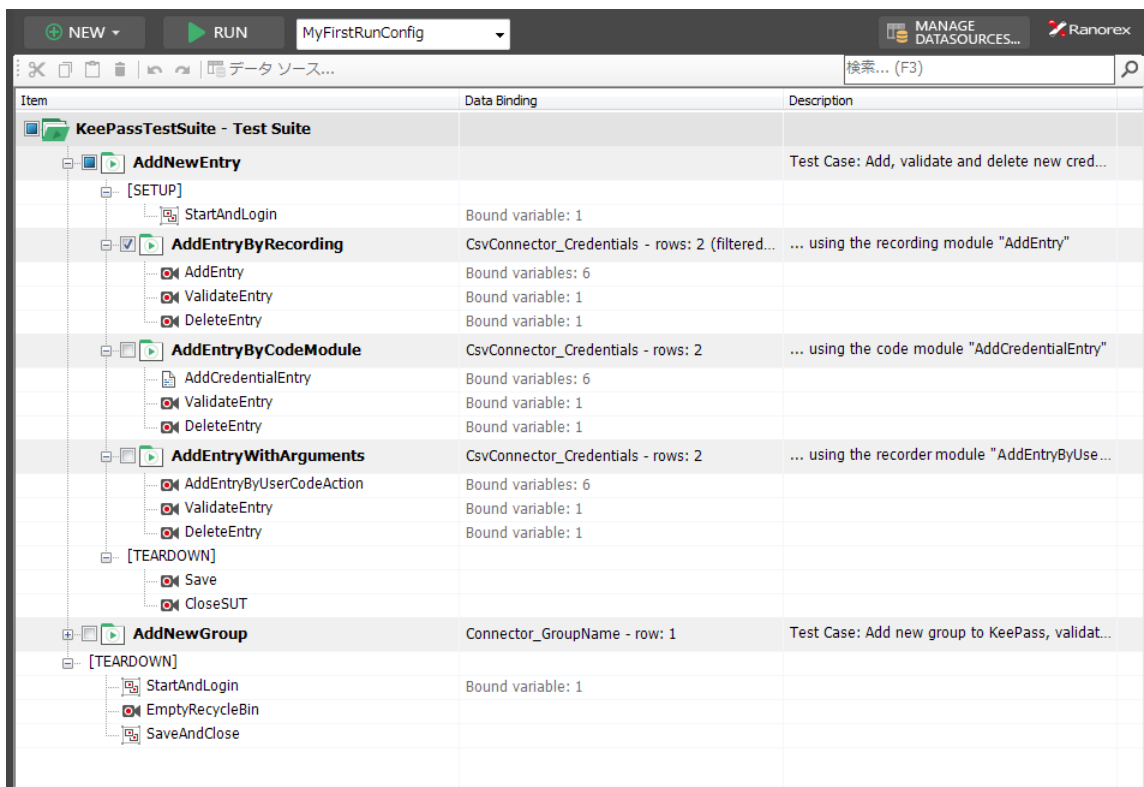
## #2 – モジュールブラウザー

モジュールブラウザーは、利用可能なすべてのモジュールとモジュールグループを表示します。また、モジュールとモジュールグループで定義されている、すべての変数も表示します。モジュールブラウザーは主に、テストスイートビューで、オートメーションモジュールとモジュールグループを再利用するために使用されます。

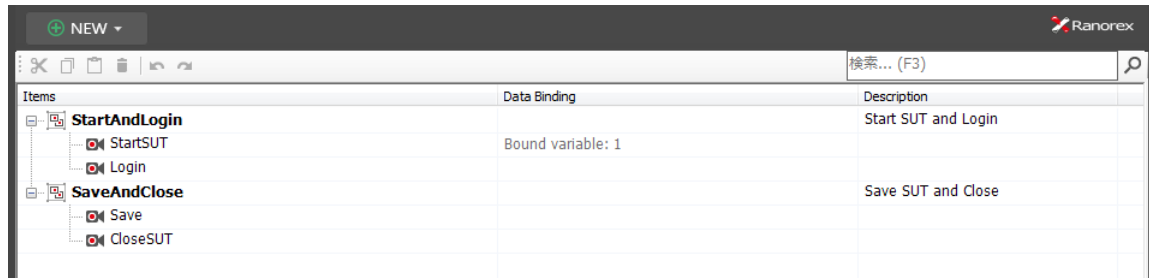
モジュールのグループ化は、モジュールグループビューでおこなえます。既に存在するモジュールを検索するには、モジュールブラウザーの検索フィールドを使用します。

## #3 – ファイルビュー

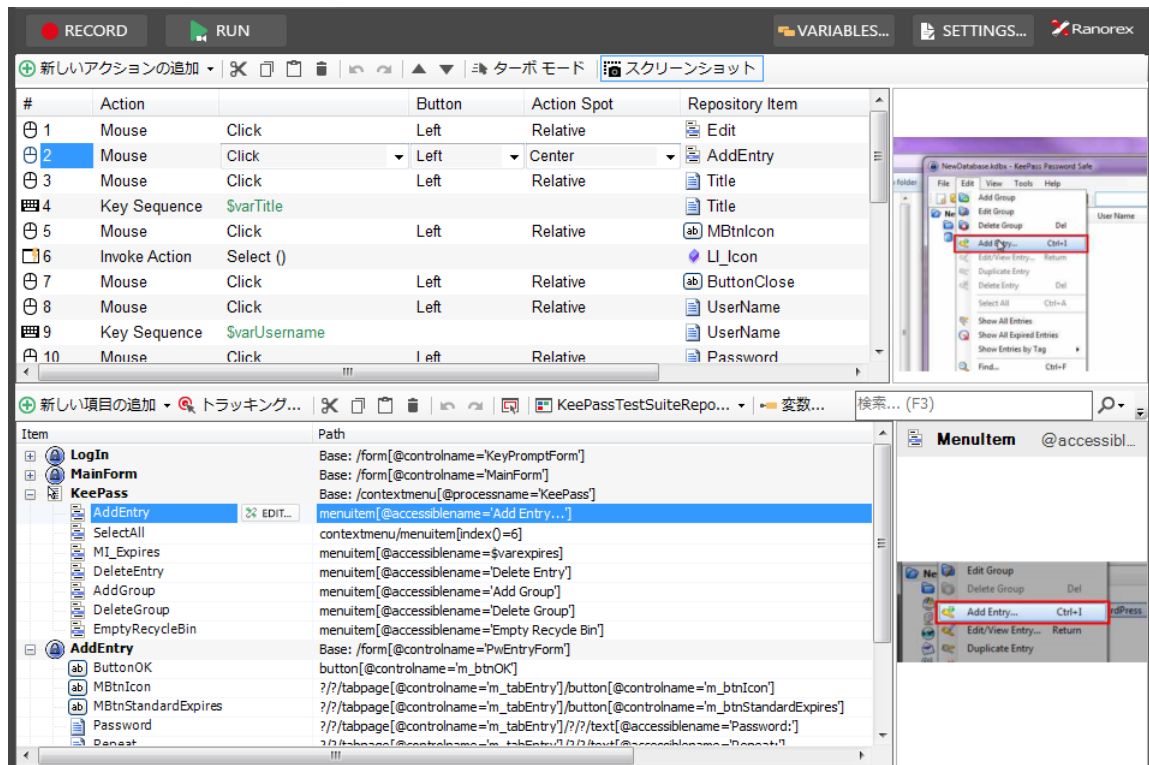
プロジェクトビューでファイルを、または、モジュールブラウザーでモジュールをダブルクリックすることで、関連付けられたファイルが、ファイルビューでオープンされます。ファイルビューは主に、以下に挙げるアクションのために使用されます。



テストスイートでの作業

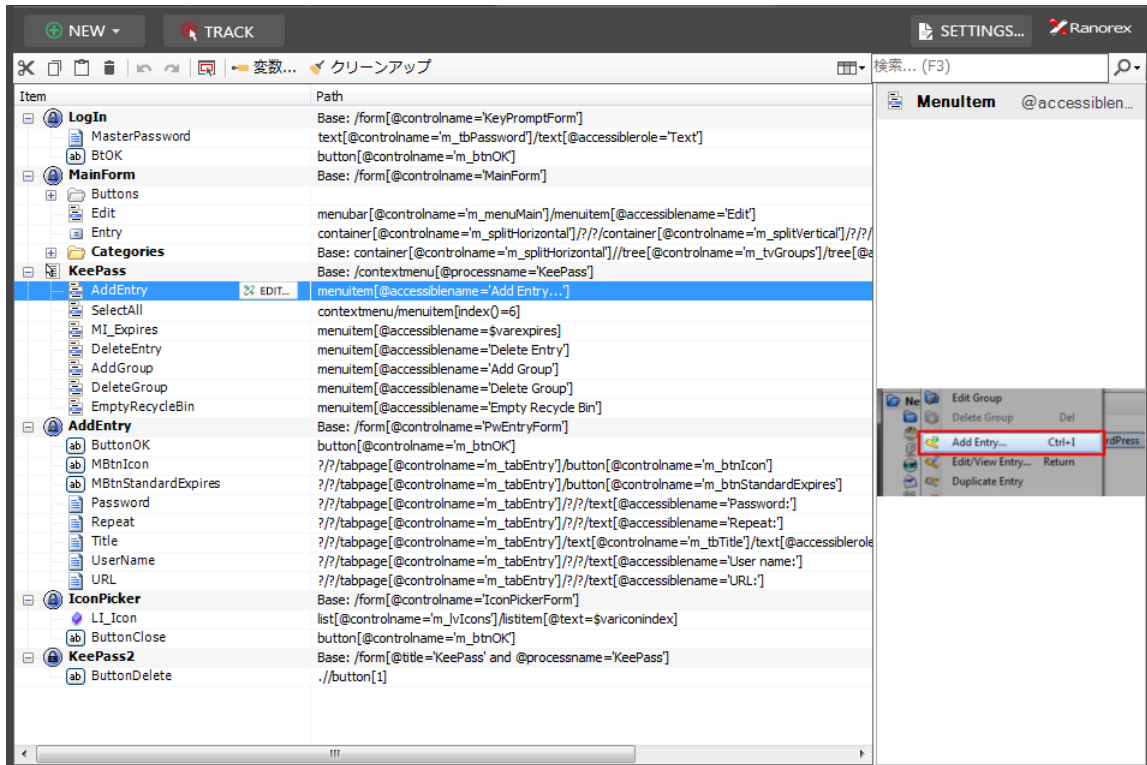


モジュールグループでの作業

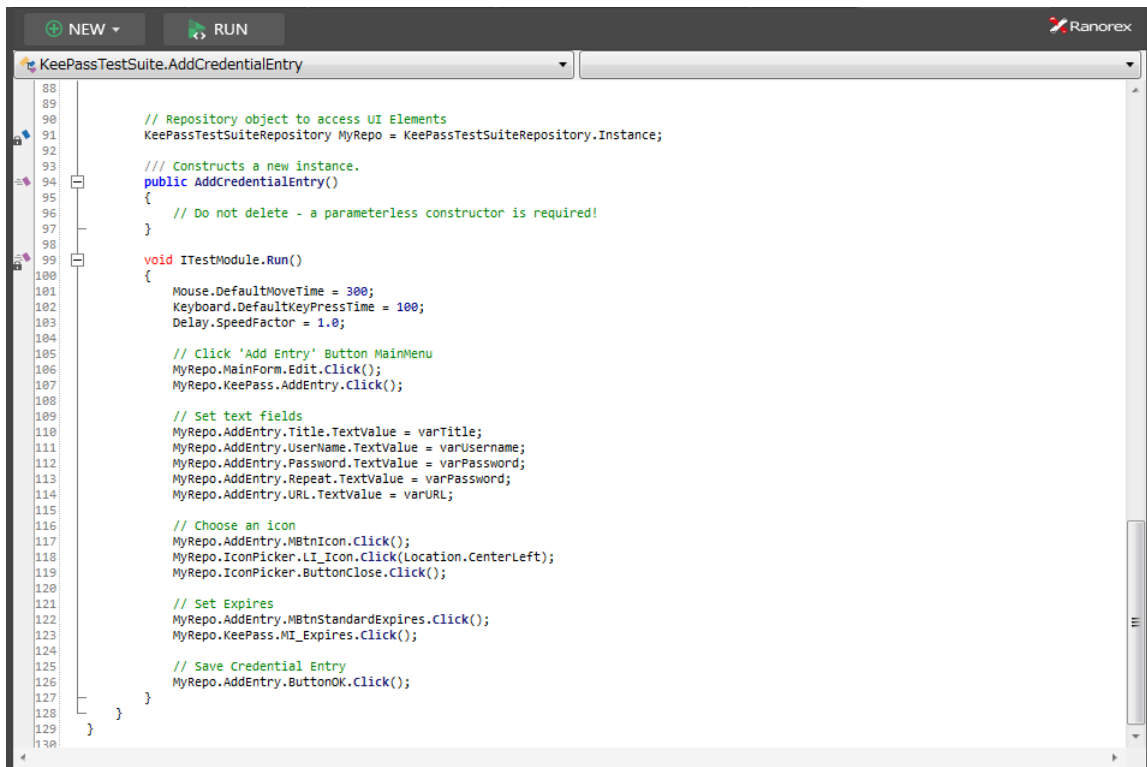


レコーディングモジュールの作成と割り当て

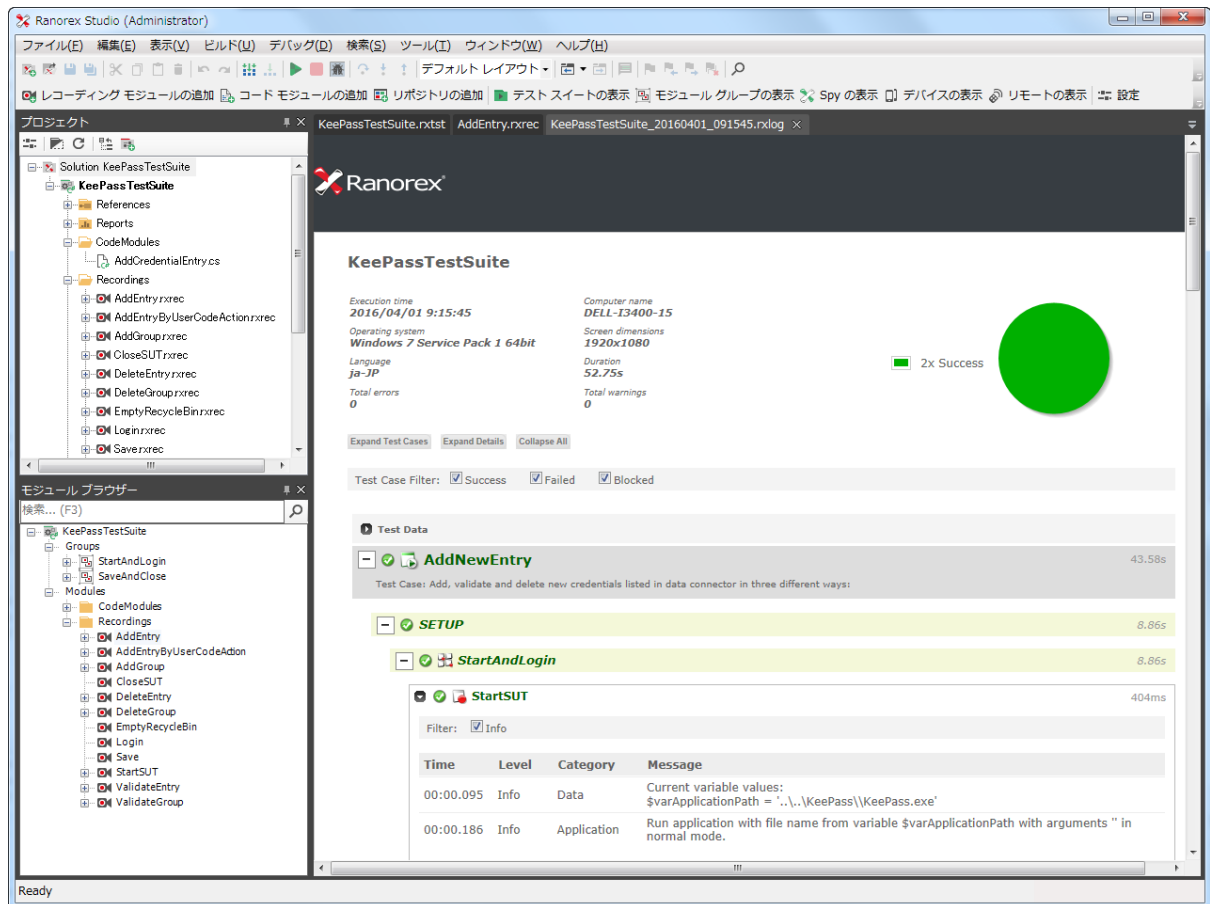




リポトリでの作業



コードモジュールの作成



テストレポートの閲覧

# レッスン 1: はじめに

この章では、レコーダーについて学習します。レコーディングの最初のステップは、デスクトップアプリケーション「KeePass」を操作する、各ユーザーアクション（マウスクリック、キー操作）のキャプチャから開始します。KeePass は、オープンソースのパスワードマネージャーアプリケーションです。

Ranorex で最初のテストを作成するもっとも簡単な方法は、手作業で実行するテストシナリオをレコーディングすることです。マウスクリックやキー操作などのレコーディングされた各アクションは、堅牢なテストケースを作成するベースとなります。この章では、以下を学習します。

- ユーザーアクションのレコーディングの準備
- テスト対象システム：KeePass アプリケーション
- テストのレコーディング
- レコードした操作の解析
- テストの実行



Ranorex レコーダーの概要については、以下の動画『Quick Start with Ranorex Test Automation』もご覧ください。以下のリンクから、該当シーンに直接ジャンプできます。

[youtu.be/0S\\_YC7uwI-s#t=1m8s](https://youtu.be/0S_YC7uwI-s#t=1m8s)

## ユーザーアクションのレコーディングの準備

レコーディングをおこなう前に、テスト対象のシステム (SUT; System Under Test) が、テストをおこなうための準備ができていることを確認してください。後でレコーディングとリポジトリのメンテナンスがおこないやすくなるよう、特に以下の点に気を付けてください。

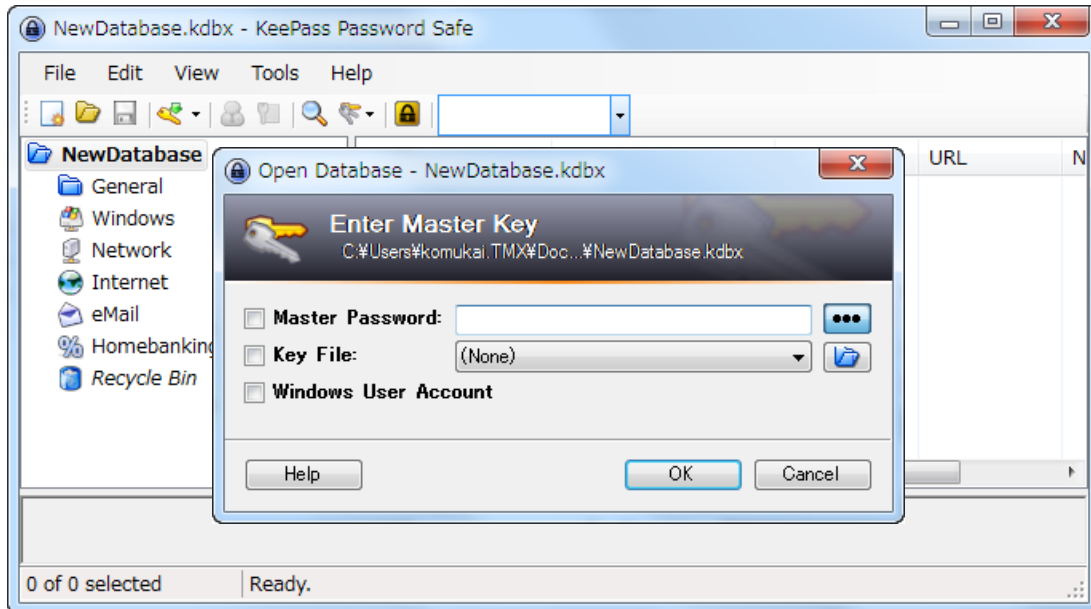
- 特に意図したテストでない限り、テスト対象のアプリケーションは複数起動しないでください。
- 初期設定では、マウスの移動は記録されません。そのため、メニュー階層を移動する場合には、明示的にマウスクリックしてください。



マウス移動の記録については、別紙『Ranorex ユーザーガイド』の「レッスン 5: Ranorex レコーダー」の「レコーダーのホットキー」を参照してください。

## テスト対象システム : KeePass アプリケーション

このチュートリアルでは、オープンソースのパスワード管理アプリケーション「KeePass」を使用して、Ranorex の基本的な機能を学習します。KeePass についての詳細は、[keepass.info](http://keepass.info) を参照してください。



KeePass アプリケーション

Ranorex バージョン 3.3 より、KeePass は、KeePass のサンプルデータベースと共に、Ranorex Studio に含まれています。



KeePass のサンプルデータベースのパスワードは **"rx"** です。

## テストのレコーディング

この節では、新規 Ranorex ソリューションの作成と、KeePass にエントリーを追加する手順のレコーディングをおこないます。

### Ranorex Studio の起動と空のレコーディングファイルのオープン

スタートメニューから、[Ranorex]-[Ranorex Studio] をクリックし、Ranorex Studio を起動します。[New Test Solution] ボタンをクリックし、空のテストスイートプロジェクトを作成します。



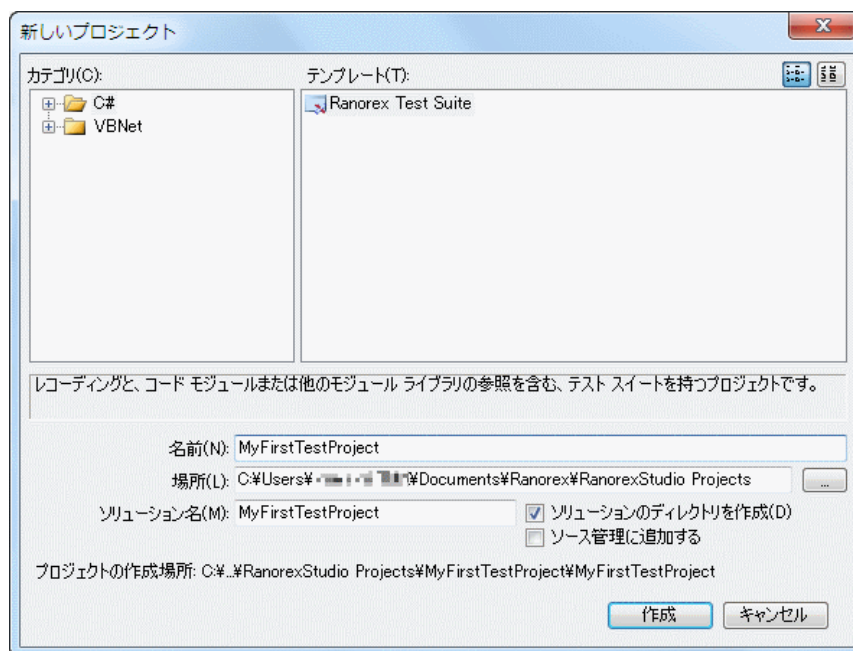
Ranorex Studio スタートページ - 新規テストソリューションの作成

カテゴリから“C#”を選択し、テンプレートから“**Ranorex Test Suite**”を選択します。“名前”と“場所”で、テストスイートプロジェクトの名前と保存場所を指定します。



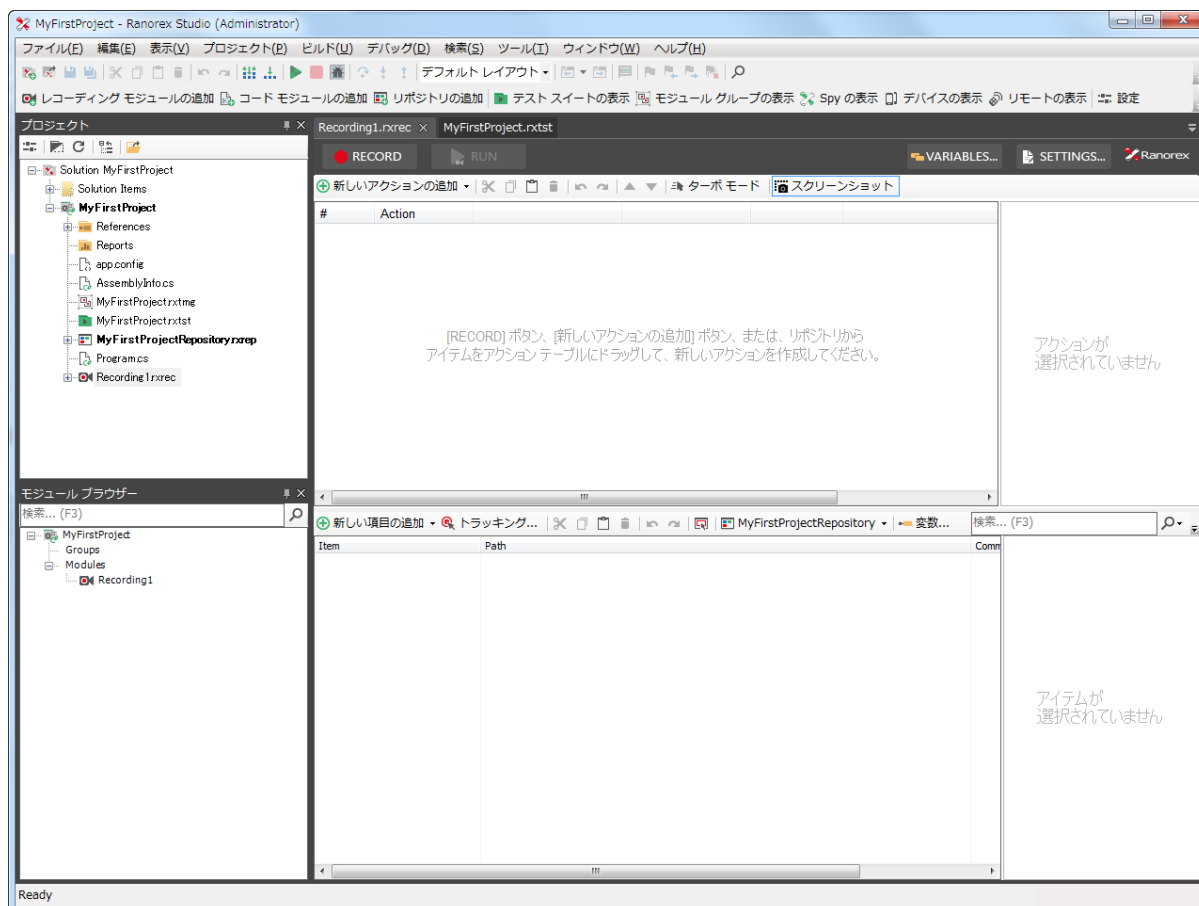
パフォーマンスとセキュリティの観点から、Ranorex プロジェクトはネットワークドライブ上には配置しないでください。同様に、Ranorex 自体もネットワークドライブ上には配置しないでください。

[作成]ボタンをクリックし、新規テストスイートプロジェクトを作成します。



使用する開発言語と、プロジェクト名・保存場所の指定

作成直後のプロジェクトでは、まだ何も記録されていない空のレコーディング”Recording1”が開かれています。



空のレコーディング

## KeePass アプリケーションのテストのレコーディングの開始



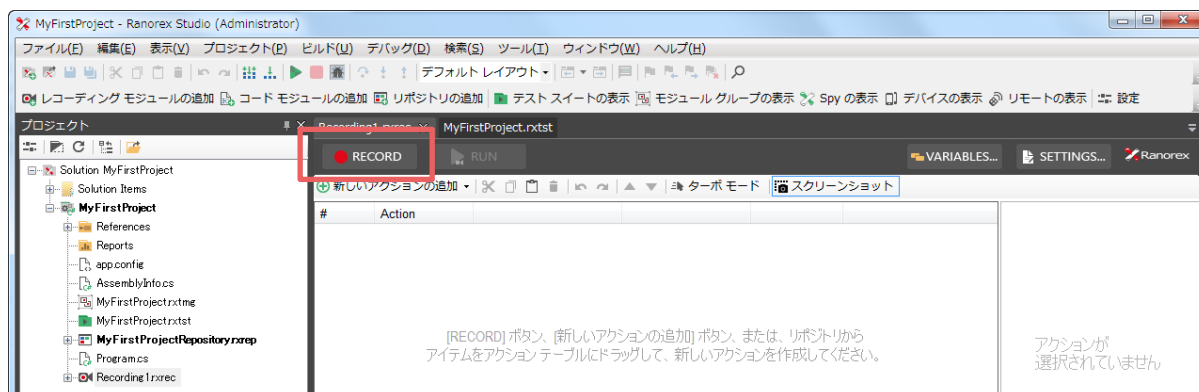
Windows におけるフォルダーアクセス権の問題から、レコーディングを開始する前に、Ranorex インストールフォルダーに含まれる、サンプルの KeePass アプリケーション (<Ranorex インストールフォルダー>¥Samples¥KeePassSample¥C#¥KeePassTestSuite¥KeePass) を、作成したプロジェクトフォルダー内にコピーしておくことをお勧めします。

プロジェクトビューでプロジェクトを選択し、右クリックメニューから[エクスプローラーでフォルダーを開く]をクリックすることで、プロジェクトフォルダーを Windows エクスプローラーで開くことができます。



プロジェクトフォルダーをエクスプローラーでオープン

レコーディングを開始するには、[Record]ボタンをクリックします。



[Record]ボタンによるレコーディングの開始

レコーダーの指示に従って、テスト対象アプリケーションを準備します。テクノロジーで“Desktop”を選択し、[アプリの追加]ボタンで、テスト対象アプリケーションに KeePass を追加します。

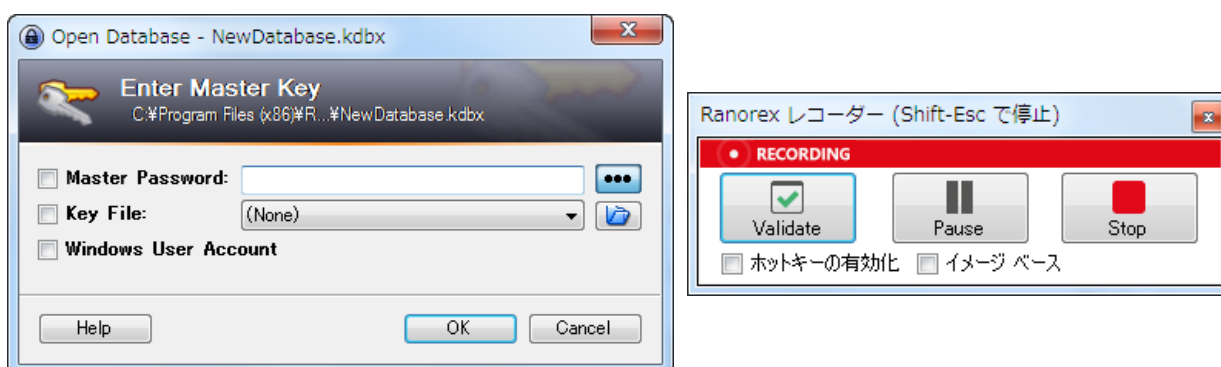


テスト対象のテクノロジーの選択



テスト対象アプリケーションの選択

[Record]ボタンをクリックすることで、レコーディングが開始されます。KeePass が起動され、Ranorex レコーダーがレコーディングを開始します。



KeePass アプリケーションのアクションのレコーディング

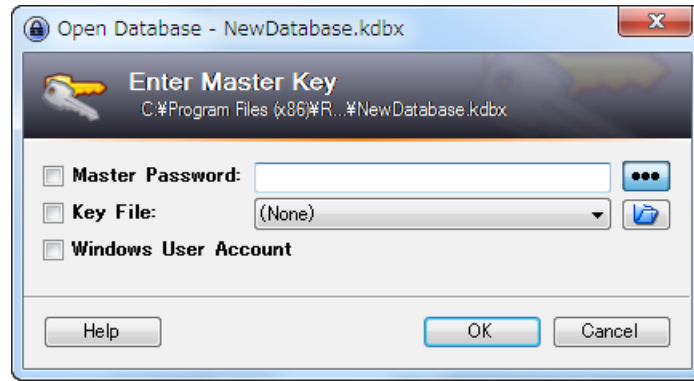


マウスを UI 要素上に動かすと、赤色のボーダーラインが表示され、該当要素が認識されたことを示します。この赤色のボーダーラインは、レコーディングをおこなう際の助けになります。

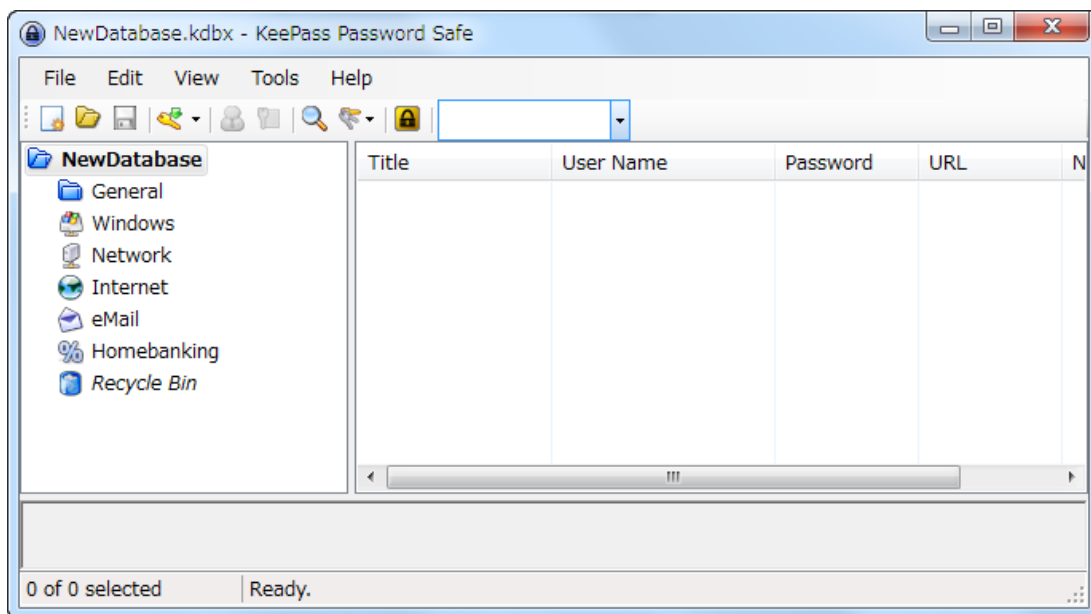
## 初期パスワードの入力

- [Master Password]テキストボックスを選択します。
- サンプルデータベースの初期パスワード“rx”を入力し、[OK]ボタンをクリックします。





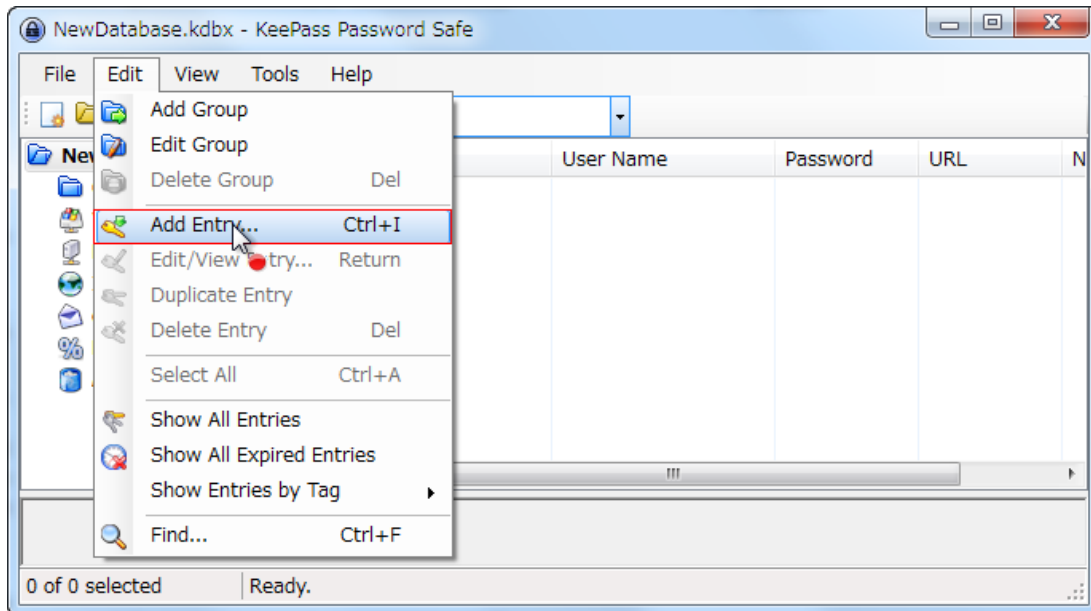
Keepass ログインダイアログ



Keepass メイン画面

## “Add Entry”フォームのオープン

- メインツールバーから[Edit]をクリックします。
- メニューから[Add Entry]をクリックします。

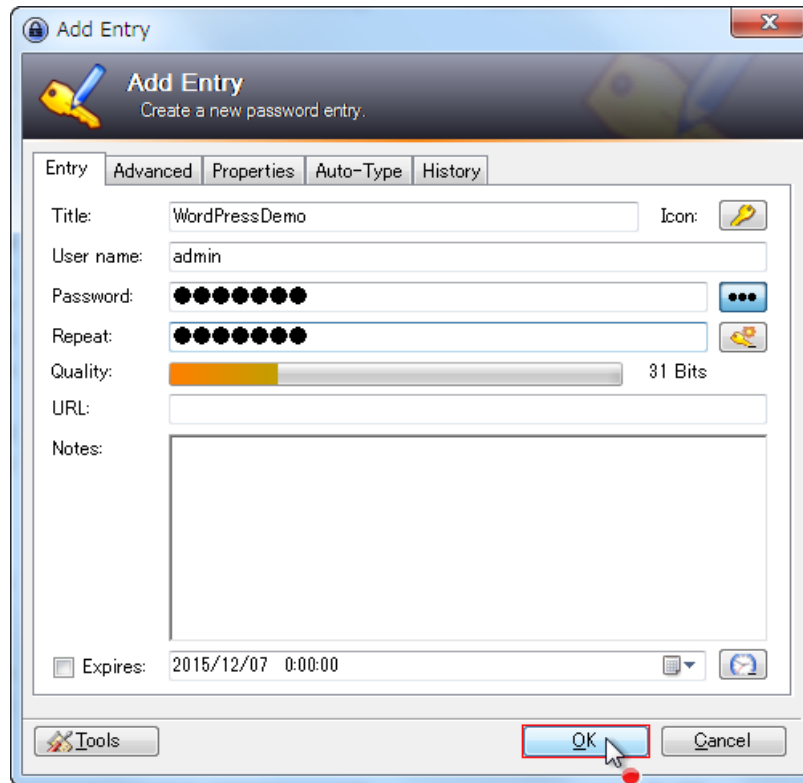


Add Entry メニュー

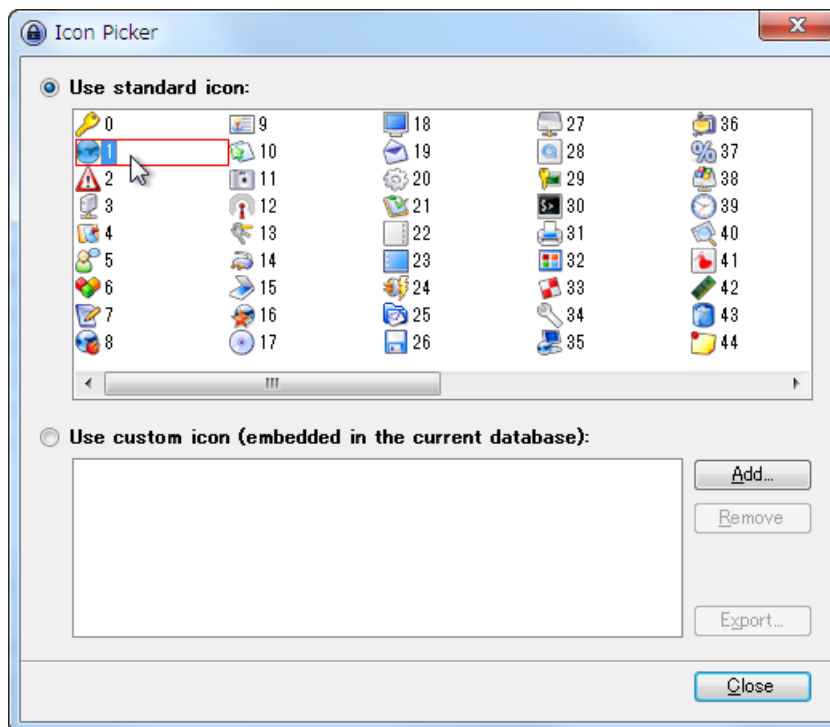
## 新規エントリーの追加

- [Title]テキストボックスに、“WordPressDemo”と入力します。
- [Icon]ボタンをクリックします。
  - アイコンの選択ダイアログで任意のアイコン (例: 2 番目のアイコン) を選択します。
  - [Close]ボタンをクリックします。
- [User Name]テキストボックスに、“admin”と入力します。
- [Password]テキストボックスをクリックします。
- [Password]テキストボックスで右クリックし、右クリックメニューを開きます。
- [Select All]を選択します。
- “demo123”と入力します。
- [Repeat]テキストボックスに、“demo123”と入力します。
- [URL]テキストボックスをクリックし、“http://bitly.com/wp\_demo”と入力します。
- [Expires]ボタン (右下の時計のシンボル) をクリックします。
  - メニューから“1 Year”を選択します。

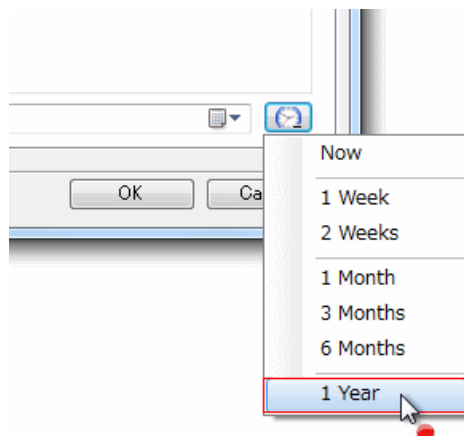
- [OK]ボタンをクリックします。



Password の入力



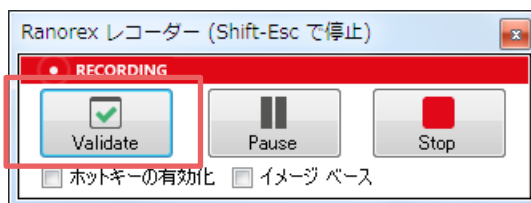
アイコンの選択



Expires の選択

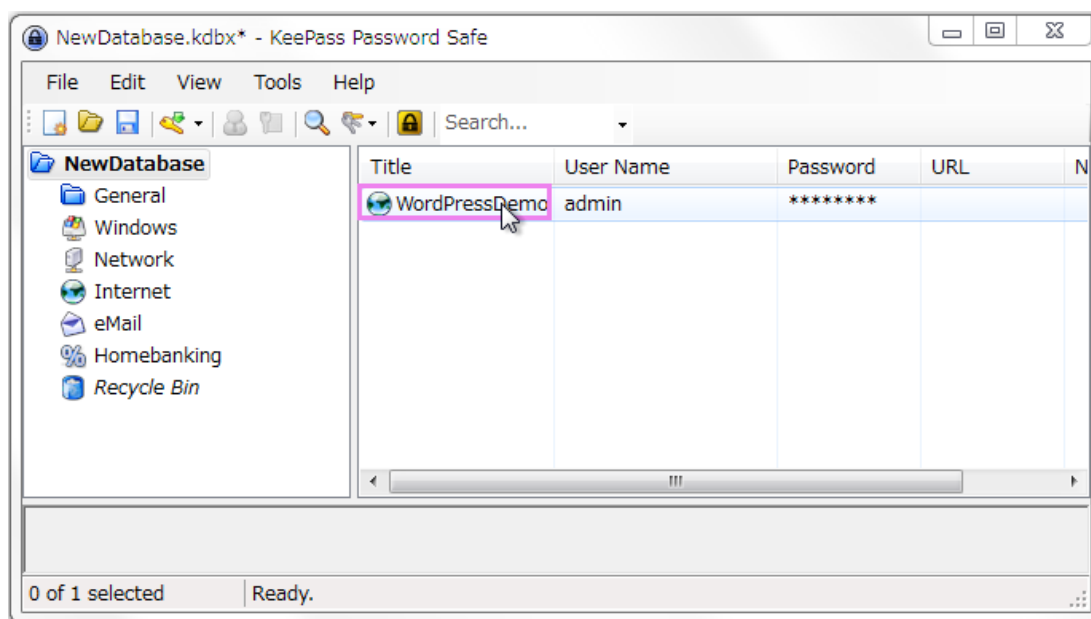
## 結果の検証

これまでの操作によって、KeePass のグリッド上に新しいエントリー行が追加されているはずです。次に、レコーディングツールバー上の[Validate]ボタンをクリックし、エントリーがグリッド上に正しく追加されているかを検証します。



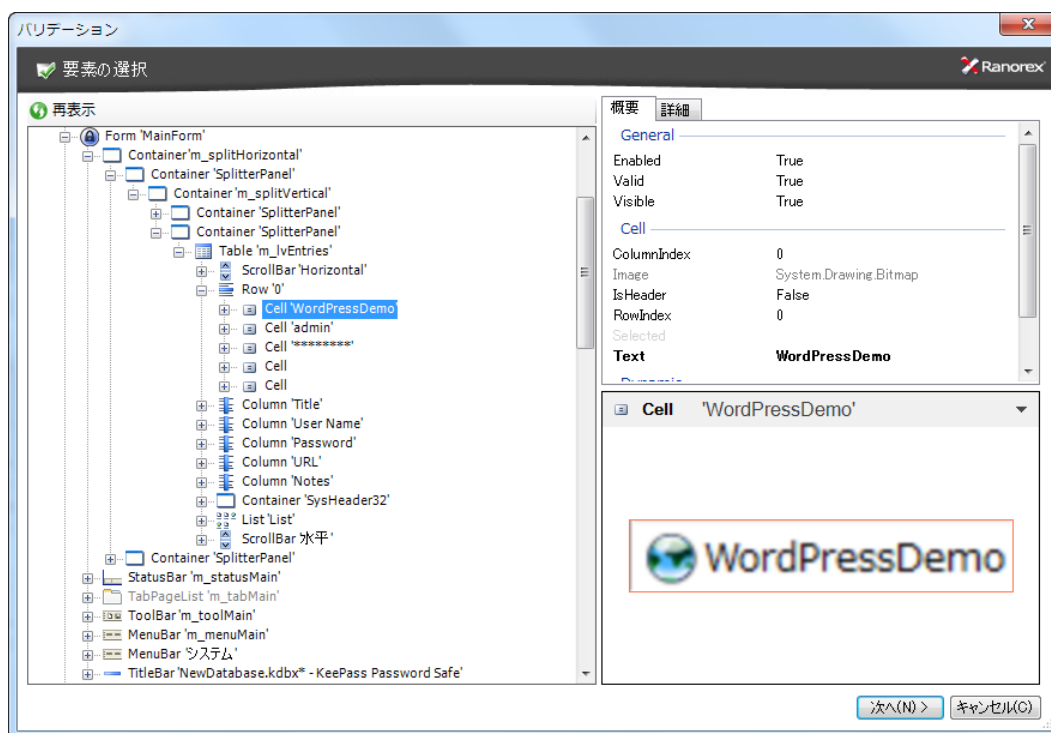
[Validate]ボタン

マウスを KeePass のグリッド上、先ほど追加したエントリーの Title ラベル部分 (“WordPressDemo”) に移動させ、UI 要素が認識されハイライトされるまで少し待ち、クリックで選択します。



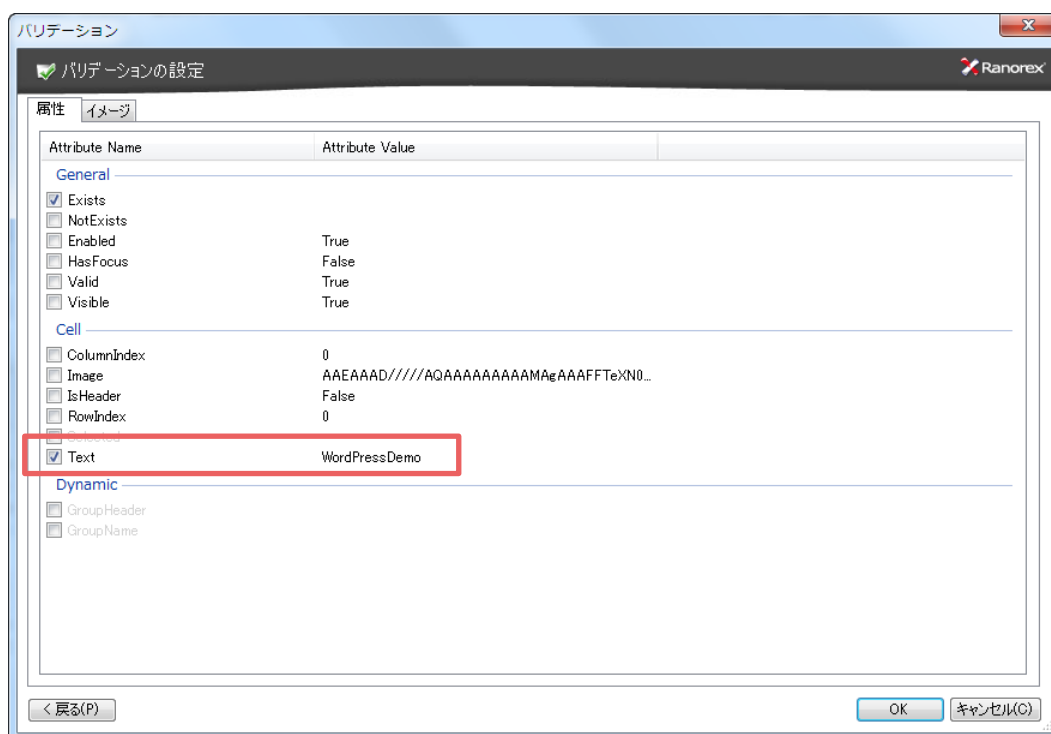
認識された UI 要素のハイライト

検証の最初のステップとして、[バリデーション]ダイアログで、確認したい UI 要素を選択します。正しい要素が選択されていることを確認し、[次へ]ボタンをクリックします。



検証する UI 要素の選択

次のステップとして、検証対象とする属性を指定します。この UI 要素についての確認可能なすべての属性が表示されます。ここでは、“Text”属性を検証の対象として指定し(チェックボックスにチェックを入れ)、[OK]ボタンをクリックします。

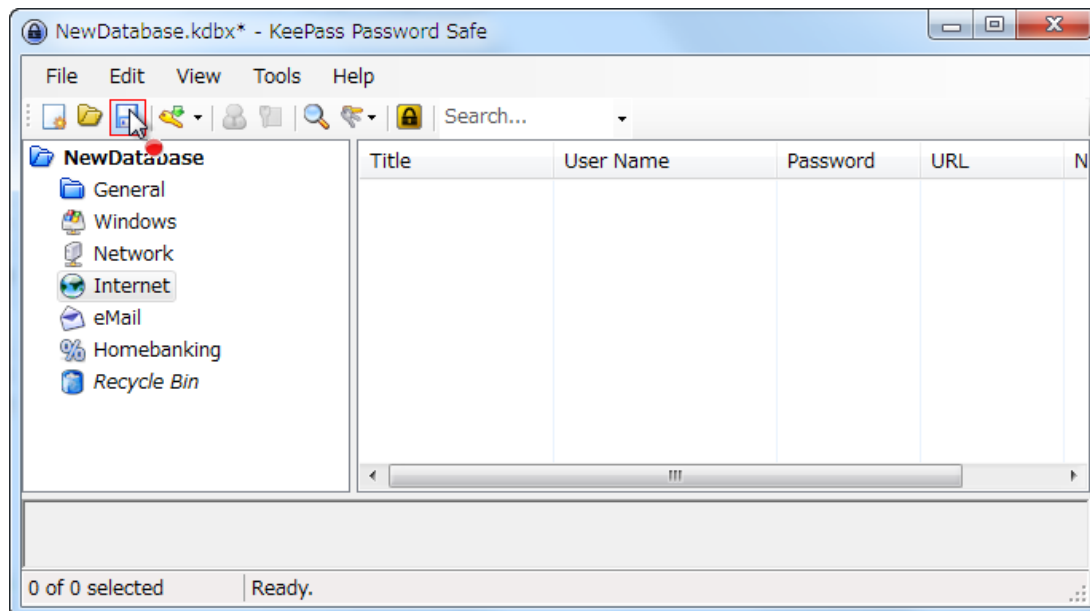


検証する属性の選択

## 追加したエントリーの削除とアプリケーションの終了

これまでの手順で追加したエントリーを選択し、Delete キーで削除します。

最後に、ツールバーの[Save]ボタンをクリックし、アプリケーション右上の閉じるボタンでアプリケーションを閉じます。



Save ボタン

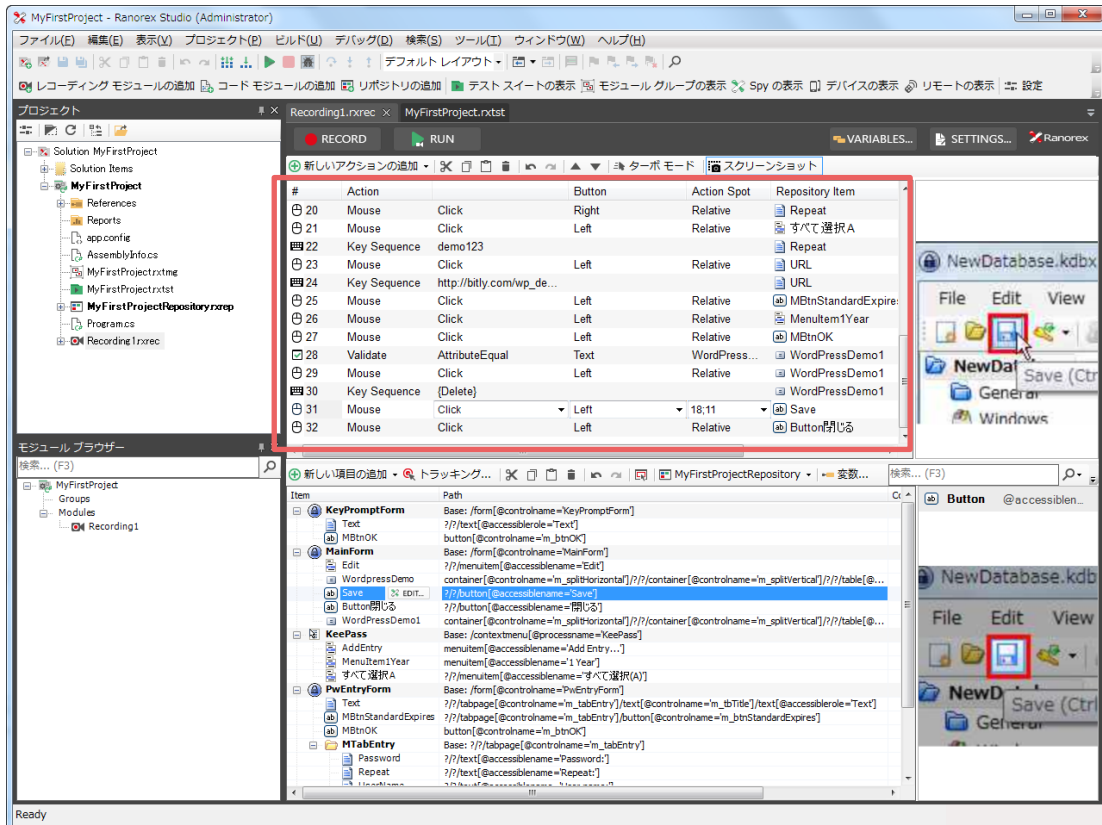
Ranorex レコーダーのツールバーの[Stop]ボタンをクリックし、レコーディングを終了します。



[Stop]ボタン

## レコードした操作の確認

レコーダーは、KeePass アプリケーション上でおこなったそれぞれの操作について、操作ステップを作成しています。これらの操作ステップはアクションテーブルに表示されます。それぞれのアクションアイテムは、レコーディングで使用されている各 UI 要素 (テキストボックス、ラジオボタン、ボタン、など) をあらわすリポジトリアイテムと関連付けられています。

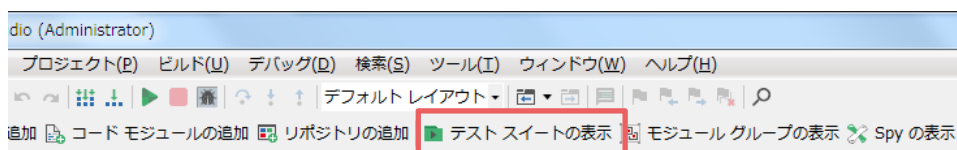


アクションテーブルに表示された操作ステップ

Ranorex レコーダーで扱われるアクションタイプについての詳細は、別紙『Ranorex ユーザーガイド』の「レッスン 5: Renorex レコーダー」の「追加の編集オプション」を参照してください。

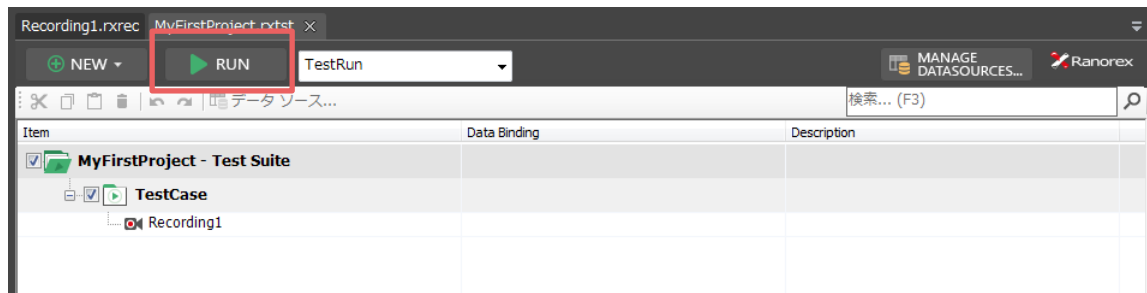
## テストの実行

ツールバーの[テストスイートの表示]をクリックし、テストスイートに戻ります。



テストスイートの表示

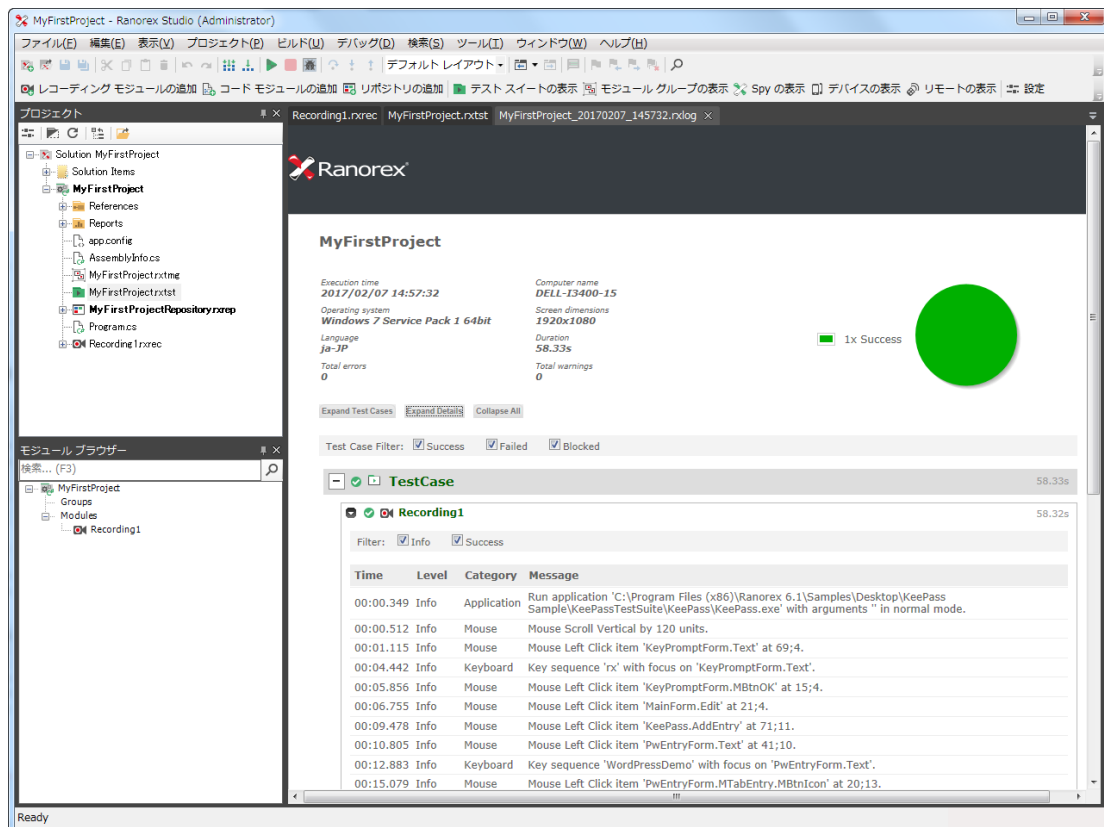
テストスイートで[Run]ボタンをクリックすることで、レコードしたテストのテストスイートが実行されます。テスト実行の間、Ranorex は、ユーザーが操作したのと同じように、ユーザーのすべてのアクション（マウス移動、キーボードイベント）をシミュレートします。



Run ボタンでのテストの実行

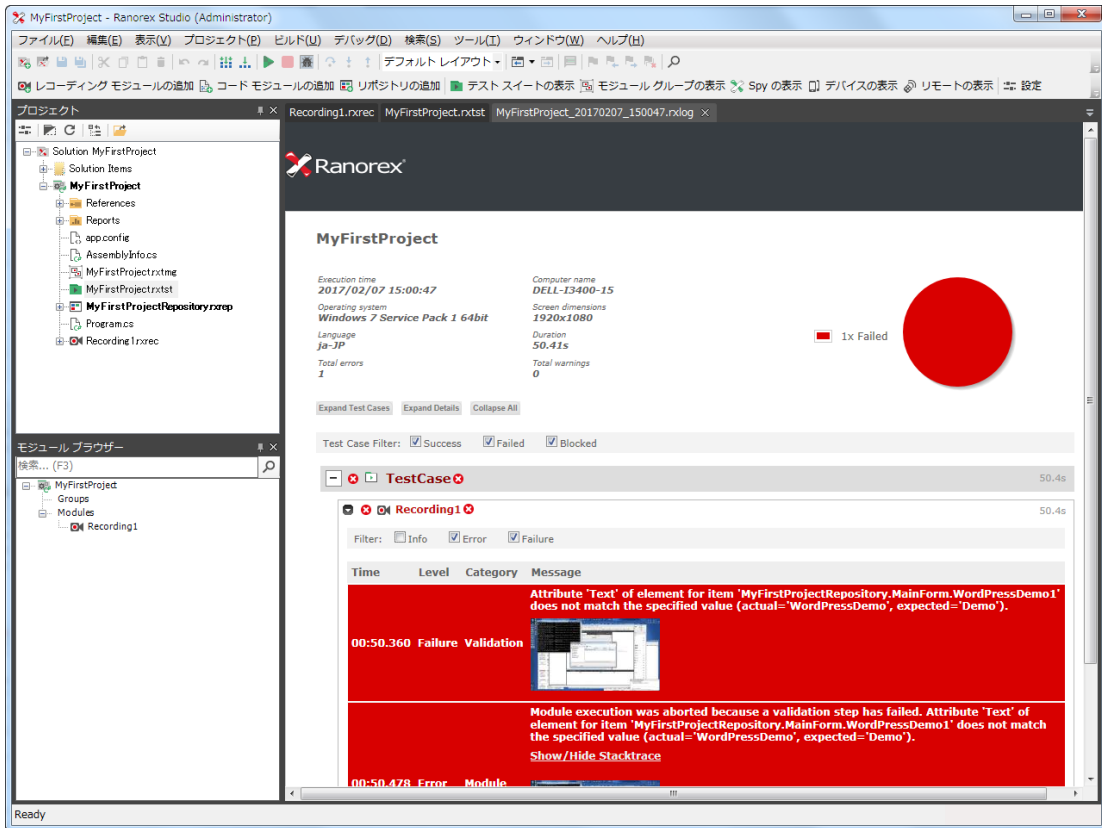
## レポート

テストの実行後、Ranorex Studio はテストレポートファイル (\*.rxlog) を自動でオープンします。



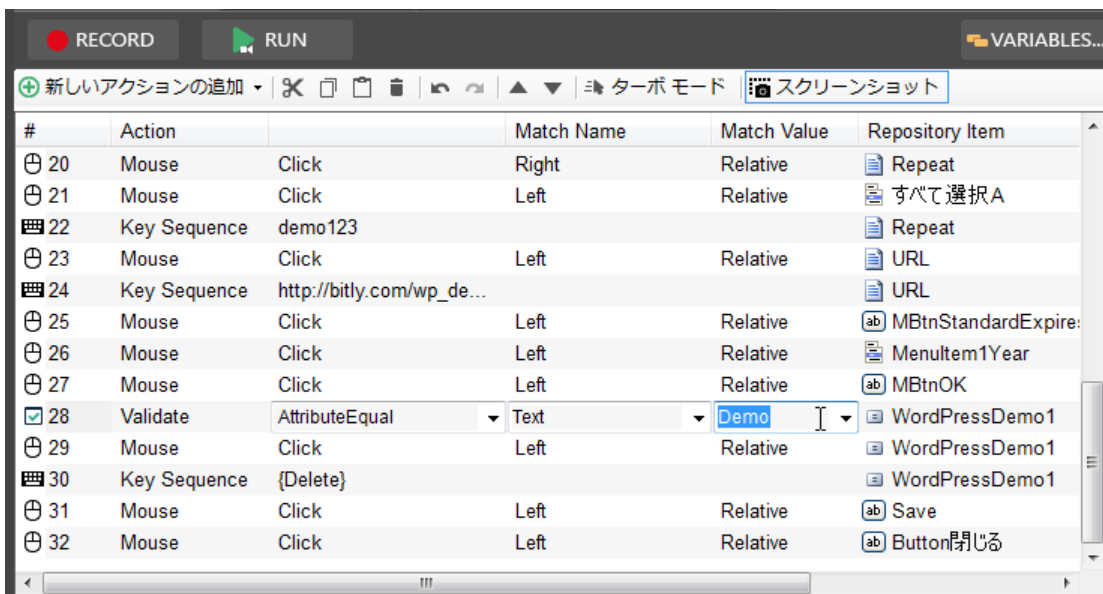
テスト成功





テスト失敗

上記のテスト失敗レポートを強制的に表示するには、レコーディングの確認ステップで、検証の期待値を変更します。(例: 属性値"WordPressDemo"を"Demo"などに変更します)

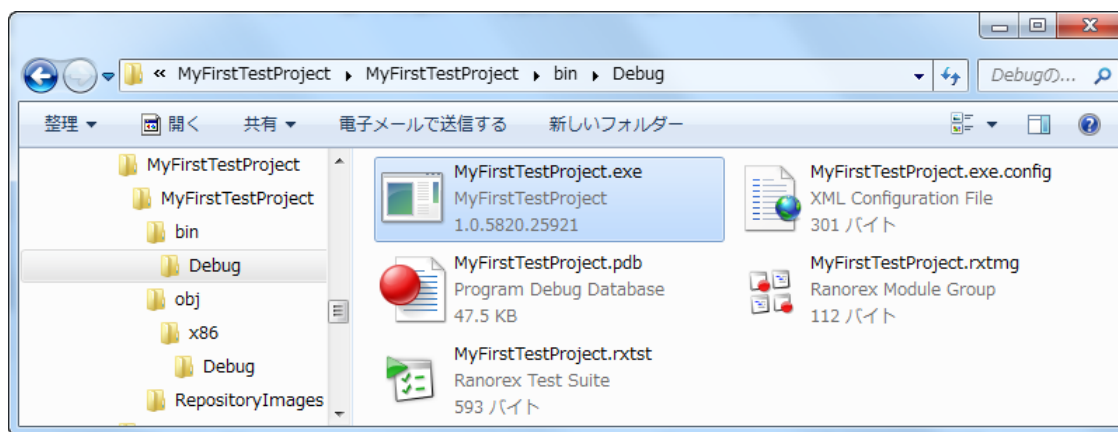


検証の期待値の変更

テストの自動実行は、作成したプロジェクトフォルダーからもおこなえます。Ranorex Studio を起動せずにテストを実行するには、テストスイートの実行ファイルをダブルクリックして起動するだけです。



別のマシンでテストスイートプロジェクトを実行するには、テストスイート実行ファイル (\*.exe) と、テストスイートファイル (\*.rxtst) を同じフォルダーに配置する必要があります。Ranorex Studio ソリューションファイルが複数のプロジェクトで構成されている場合、ライブラリファイル (\*.dll) も同じフォルダーに配置する必要があります。よりシンプルには、出力フォルダー（例: bin/debug）とその配下のファイルをすべてコピーするのが簡単です。



出力フォルダーに配置されたテストスイート実行ファイル“MyFirstTestProject.exe”

初期設定では、レポートファイルも同じフォルダーに出力されます。レポート出力フォルダーの変更については、別紙『Ranorex ユーザーガイド』の「レッスン 4: Ranorex テストスイート」の「テストスイートの設定」を参照してください。

テストスイートを、コマンドラインやスタンドアロンのテストスイートランナーから実行する方法については、別紙『Ranorex ユーザーガイド』の「レッスン 4: Ranorex テストスイート」の「Ranorex Studio を使用しないテストの実行」を参照してください。

# レッスン 2: Ranorex モジュール - テストアクション

この章では、レコーディングを再利用性の高い小さな部品に分割すること、Ranorex Studio で、ドラッグアンドドロップ操作によって簡単にプロジェクトを構成していく方法について学習します。

## リファクタリング: レコーディングを分割する必要性

レコーディングのしやすさを考えた場合、再利用性を考慮せずに、すべてのテストケースを一つのレコードで管理する方が簡単です。しかし長い目で見れば、これは、テストオートメーションの管理コストを増大させる原因となります。このため、キーワード駆動型手法で推奨されているように、レコーディングを再利用性の高い小さな部品に分割する必要があります。以下の節では、既存のレコーディングを、他のテストケースで再利用可能な、小さな部品に分割する方法を学習します。



Ranorex モジュールの概要については、以下の動画『Ranorex Modules – Test Automation Actions』もご覧ください。

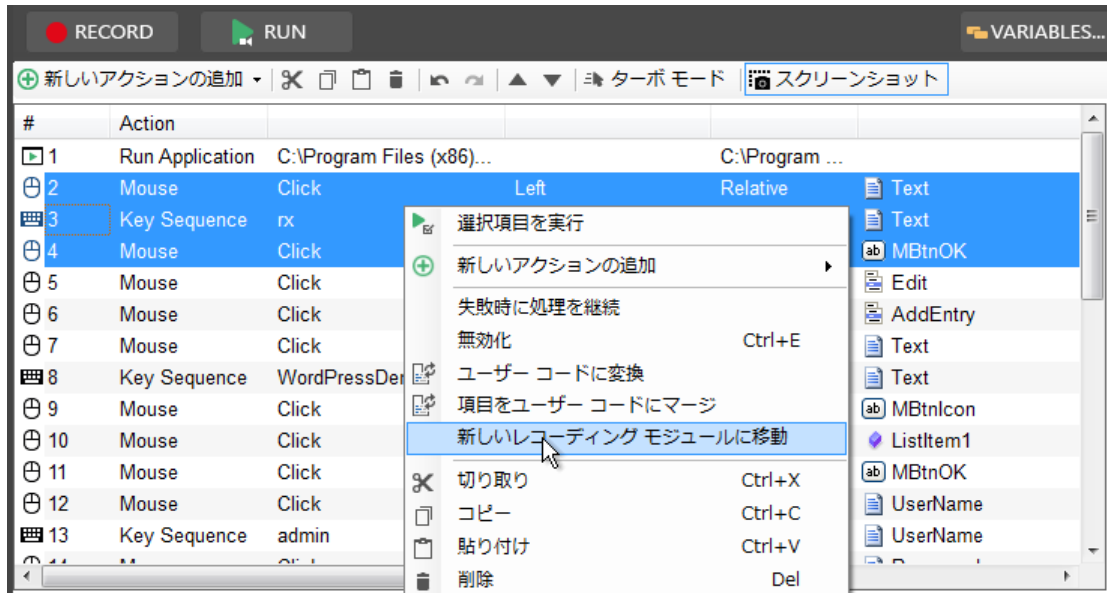
[youtu.be/GrrKnf8cQ2g](https://youtu.be/GrrKnf8cQ2g)

## モジュールの特定

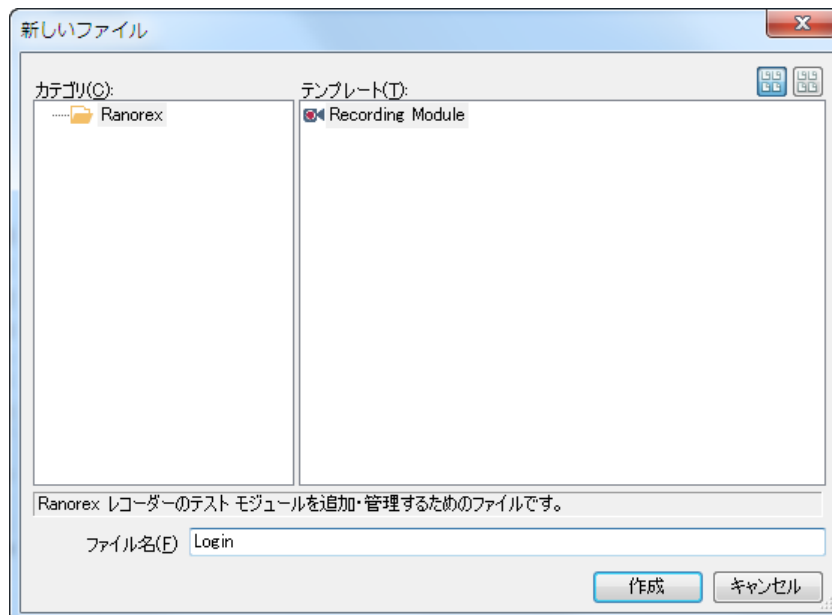
前章でおこなった最初のレコーディングは、キーワード駆動の観点からは、以下の操作をおこなっています。

- KeePass アプリケーションの開始
- ログイン
- 新規エントリーの追加
  - タイトル・ユーザー名・パスワード・URL の入力
  - アイコンの選択
  - 期限の設定
- 作成したエントリーの検証
- エントリーの削除
- 変更の保存
- アプリケーションの終了

レコーディングをオートメーションモジュールに分割するためには、関連するアクションを選択し、コンテキストメニューから、[新しいレコーディングモジュールに移動]をクリックします。元となるレコーディングで、ログインをおこなうための 3 つのアクションを選択し、レコーディングモジュールに移動し、“Login”という名前を付けます。3 つのアクションはそれぞれ、Password フィールドのクリック、Password フィールドでのキー入力、OK ボタンのクリックです。



ログインアクションを新規レコーディングに移動



新規レコーディング“Login”の作成

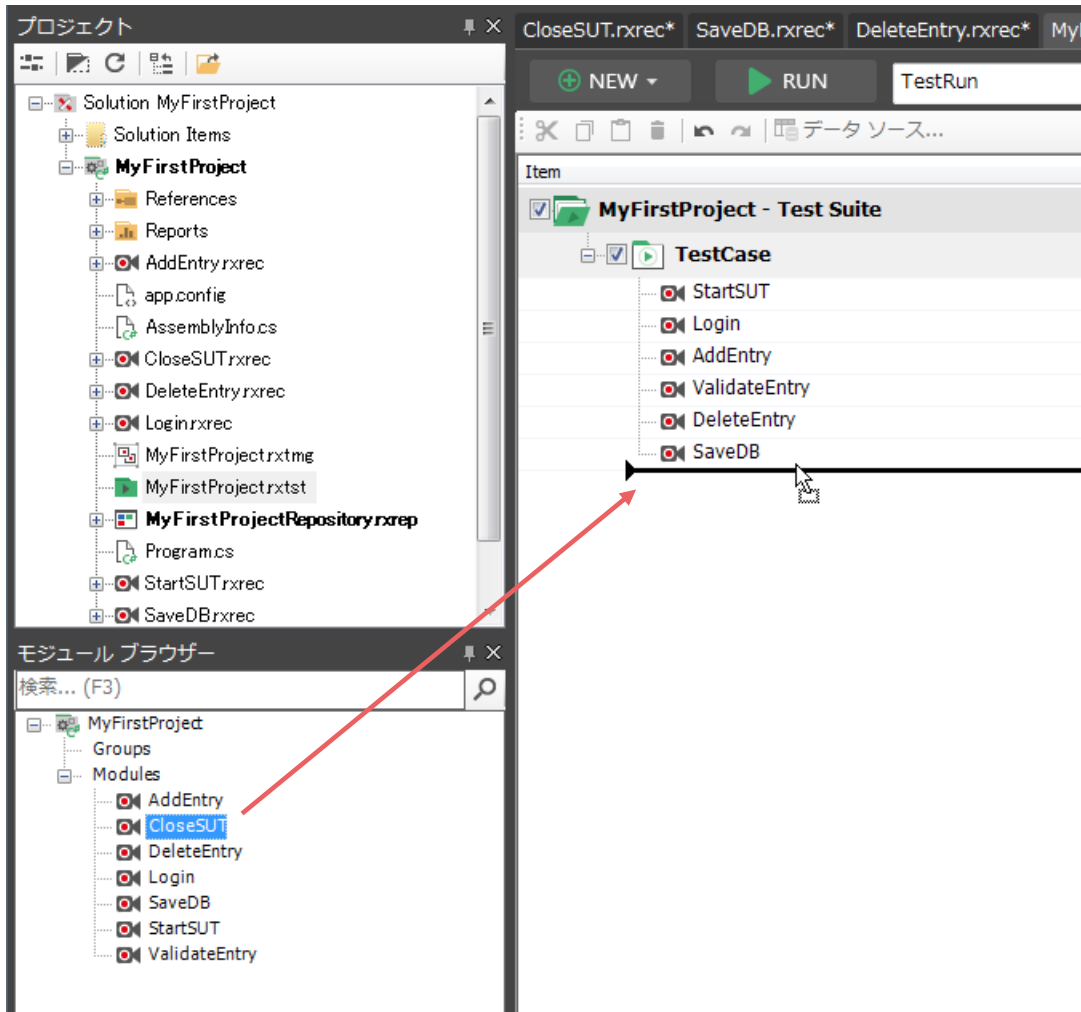
次に、これまでの手順にならって、新規レコーディングモジュール“AddEntry”、“ValidateEntry”、“DeleteEntry”、“SaveDB”、“CloseSUT”を同じように作成します。

最後に、プロジェクトビューまたはモジュールブラウザーで、アプリケーションを起動するアクションだけが残ったレコーディング **Recording1** の名前を変更し、**StartSUT** とします。この結果、7 つに分割されたレコーディングモジュールが作成されたはずですが。



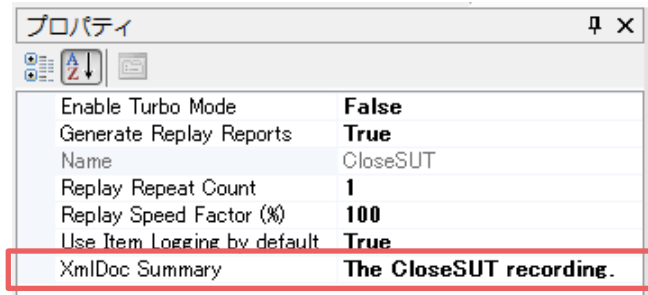
モジュール名には、マルチバイト文字を使用することもできます。

モジュールブラウザーで、分割されたモジュールが確認できます。ドラッグアンドドロップで、これらのモジュールをテストケースに配置できます。



ドラッグアンドドロップによるモジュールのテストケースへの配置

非常に多くのモジュールが存在する場合には、モジュールブラウザー内で、モジュール名や XmlDoc Summary で、モジュールの検索をおこなうことができます。XmlDoc Summary は、モジュールのプロパティから指定できます。モジュールブラウザーからモジュールを選択し、コンテキストメニューから [プロパティ] をクリックすることで、プロパティウィンドウがオープンします。



XmlDoc Summary の設定

レコーディングを小さなモジュールに分割することは、堅牢で再利用性の高いオートメーションモジュールを作成する、最初のステップです。

また、これらのモジュール中で使用されている定数値を、変数で置き換えることも重要です。変数の利用方法については、「レッスン 3: データ駆動テスト」で学習します。

## レッスン 3: データ駆動テスト

この章では、データ駆動テストをおこなうための、内部データセット (データテーブル)、外部データセット (Excel, CSV, SQL データベース) を使用したテストについて学習します。内部/外部データソースと結合される、レコーディングおよびリポジットリにおける、変数の利用方法についても学習します。さらに、データ駆動テストが適切に実行できるよう、通常非表示な UI 要素で実行されるアクションの選択方法について学習します。



データソースを使用したデータ駆動テストの概要については、以下の動画『Data Driven Test Automation with Ranorex』もご覧ください。

[youtu.be/TRQy2JI79-U](https://youtu.be/TRQy2JI79-U)

アプリケーションをテストする時、同じテストを異なる入力データでおこなう必要があります。この章では以下を学習します。

- レコーディングでの変数の利用
- リポジットリでの変数の利用
- テストデータの作成
- テストデータと変数の結合
- パラメーターと変数の結合
- 選択アクションの実行
- データ駆動テストの実行

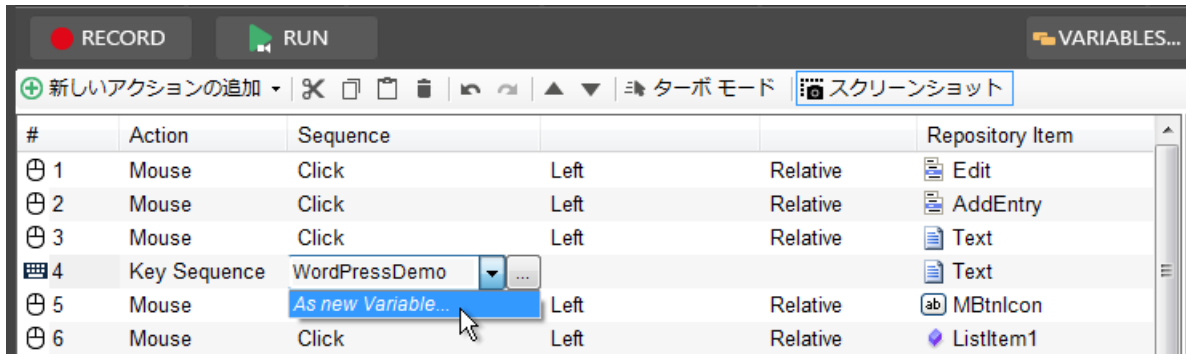
### レコーディングでの変数の利用

以前の章でおこなった KeePass アプリケーションのレコーディングでは、6 つの変数入力アクションがあります。

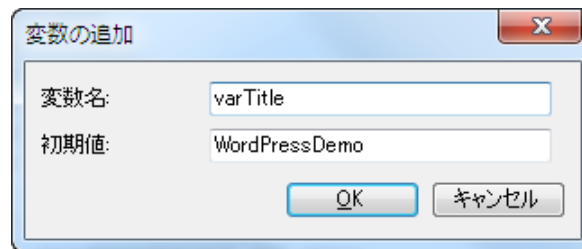
- Title (WordPressDemo)
- Username (admin)
- Password (demo123)
- URL ([http://bitly.com/wp\\_demo](http://bitly.com/wp_demo))
- Expires (1 Year)
- IconIndex (1)

これらの入力アクションは、“AddEntry”レコーディングで使用されています。レコーディングファイルをオープンし、変数にすべきアクションを確認します。

Title に入力する値を変数にするには、以下のようにコンボボックスから[As new Variable]を選択し、新しい変数を作成します。



キー操作アクションへの変数の作成



変数の作成

変数名に **varTitle** を指定します。初期値には、初回のレコーディング時に指定された値が入力されています。変数名と初期値を変更する方法については、次節の「リポジトリ内での変数の使用」を参照してください。



変数名には、レコーディングやコードモジュールで使用されている名前は使用しないでください。また、マルチバイト文字は使用しないでください。

上記の手順を繰り返し、Password, Username, URL についても、それぞれのキー操作アクションについて、変数 **varUsername**, **varPassword**, **varURL** を作成します。

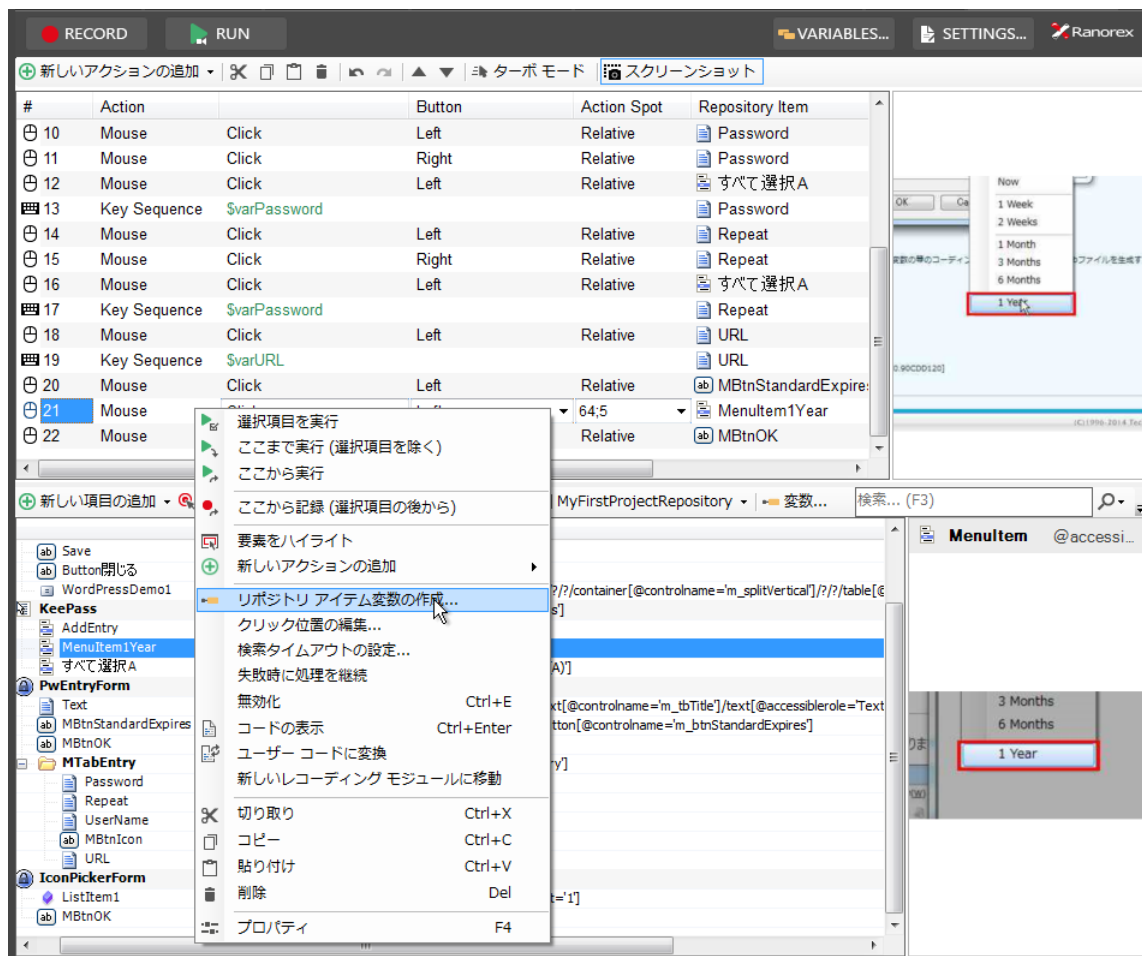
## リポジトリ内での変数の使用

前節で、“AddEntry”レコーディングのキー操作アクションについて、“**varTitle**”, “**varUsername**”, “**varPassword**”, “**varURL**”変数を作成しました。

“Expires”の値 (1 week, 1 year, ...) の選択のような、コンテキストメニューの選択などのクリックアクションをデータ駆動にするには、RanoreXPathを使用して変数を定義する必要があります。

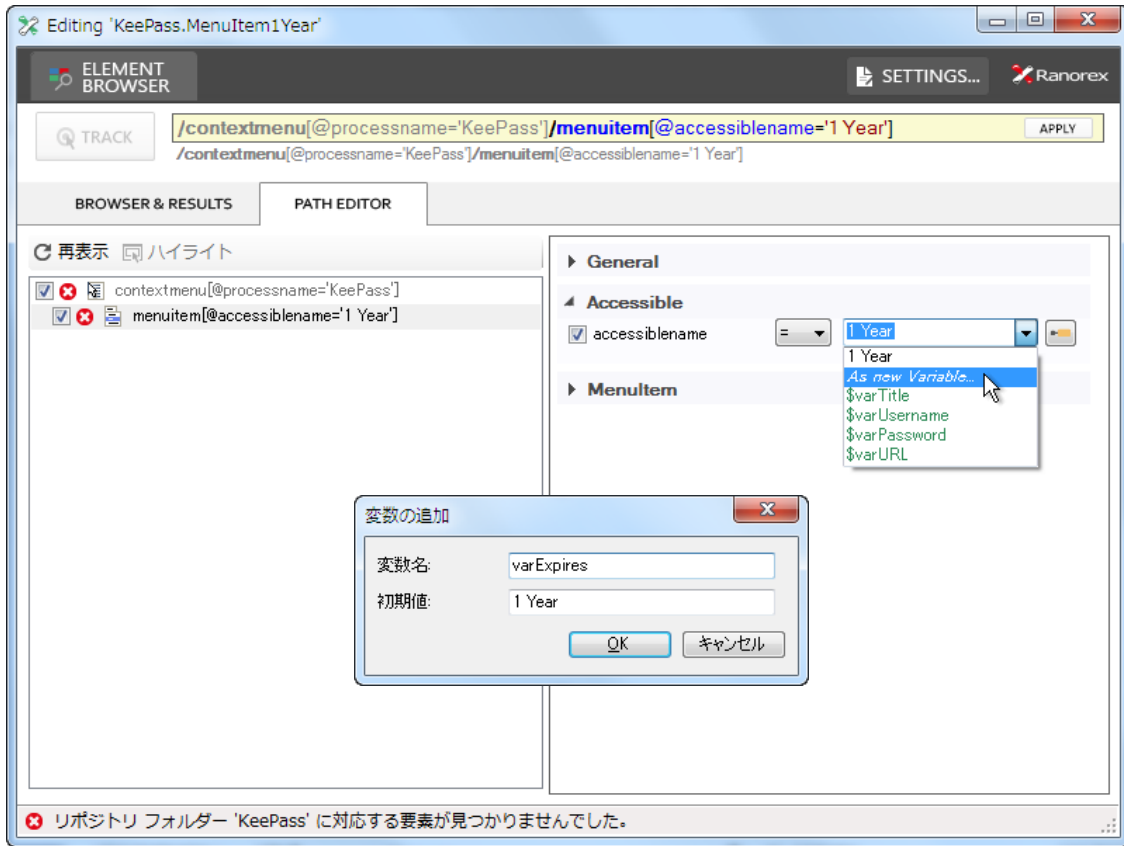
“AddEntry”レコーディングで、以下に示すような、メニューの選択とコンテキストメニューのオープンに対応するアクションを選択します。コンテキストメニューから、[リポジトリアイテム変数の作成]をクリックし、パスエディターを開きます。





コンテキストメニュー [リポジトリアイテム変数の作成]

通常、KeePass アプリケーションの“Expires Data”メニューは、accessible name 属性で識別されます。画面右のリストボタンをクリックし、新しい変数 **varExpires** を定義します。



Expires メニューアイテムへの変数の作成

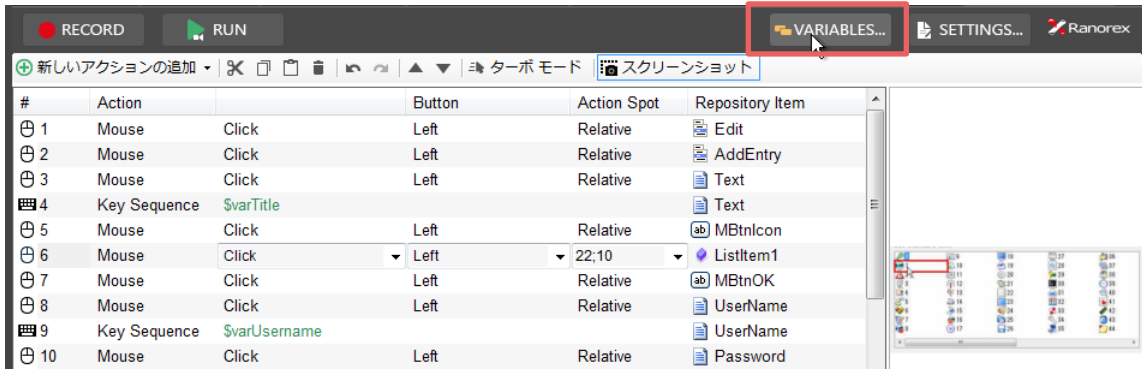
リポジトリを確認すると、メニューアイテムに対応する要素が選択されていて、この要素が、上記手順で指定したパス表現を使用していることが確認できます。

Item	Path
KeyPromptForm	Base: /form[@controlname='KeyPromptForm']
MainForm	Base: /form[@controlname='MainForm']
KeePass	Base: /contextmenu[@processname='KeePass']
AddEntry	menuItem[@accessiblename='Add Entry...']
MenuItem1Year	menuItem[@accessiblename=\$varExpires]
すべて選択A	??/menuItem[@accessiblename='すべて選択(A)']
PwEntryForm	Base: /form[@controlname='PwEntryForm']
IconPickerForm	Base: /form[@controlname='IconPickerForm']

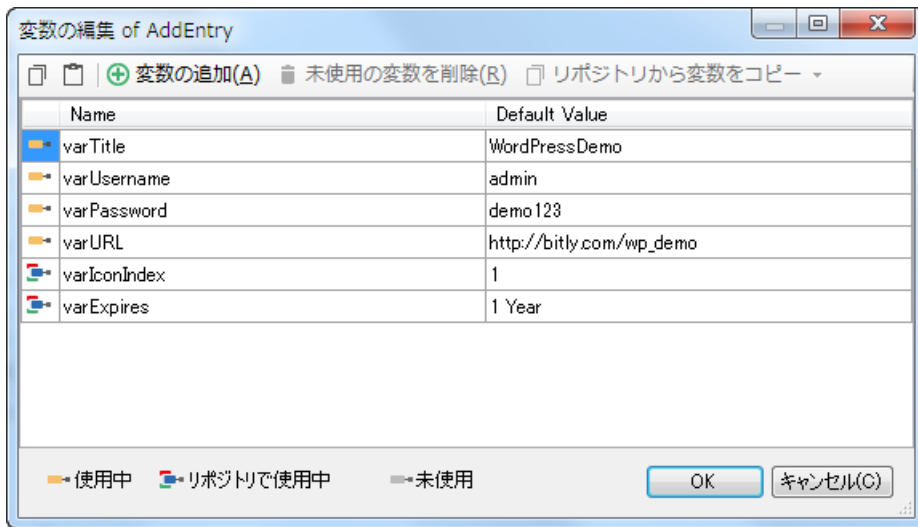
変数を使用しているリポジトリアイテム

また、リポジトリアイテム **MenuItem1Year** の名前を、このアイテムが既に特定のメニューアイテムを指しているのではなく、データ駆動でメニューアイテムを指していることを明示するために、**MI\_Expires** に変更します。さらに、これまでの手順を繰り返し、アイコンリストに関連付けられているリポジトリアイテム **ListItem1** について、新しい変数 **varIconIndex** 作成します。

これまでの手順で、"AddEntry"レコーディングでは、6 つの変数が定義されました。4 つは、レコーディングで記録されたキー操作で使用される変数、残り 2 つは、リポジトリで使用される変数です。レコーディングで使用されるすべての変数についてのサマリーを確認するには、以下のように[Variables]ボタンをクリックします。



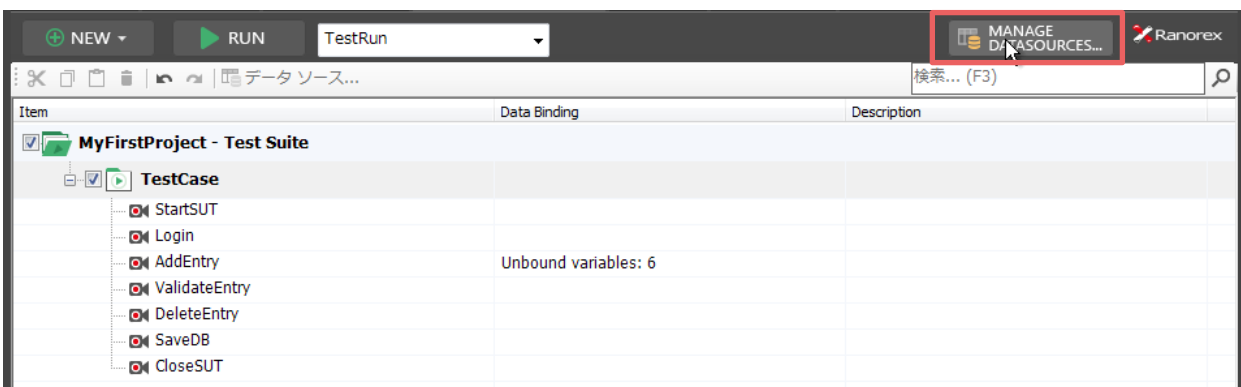
変数ダイアログのオープン



レコーディングとリポジトリで使用されている変数の一覧

## テストデータの作成

この節では、前節で作成した変数と、Excel テーブルに記述した値のバインド方法について学習します。新規 Ranorex データソースを作成するには、テストスイートビューを開きます。“AddEntry”レコーディングを使用しているテストケースを選択し、ツールバーの[Manage Data Sources]、または、右クリックメニューの[データソースの管理]をクリックし、データソースダイアログを開きます。

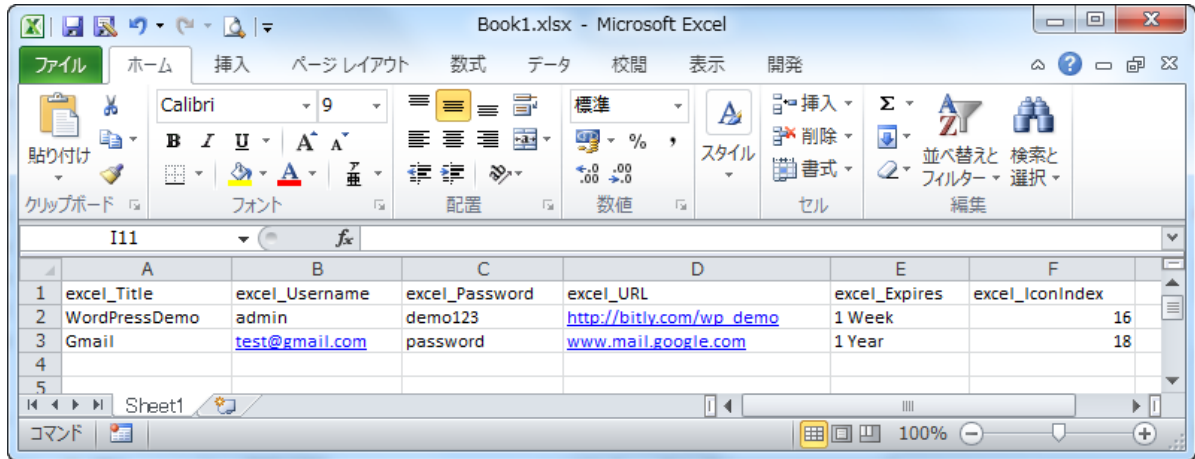


[データソースの管理]ダイアログのオープン

また、別途 Title, Username, Password, URL, Expires, IconIndex 列が含まれる Excel ファイルを用意します。

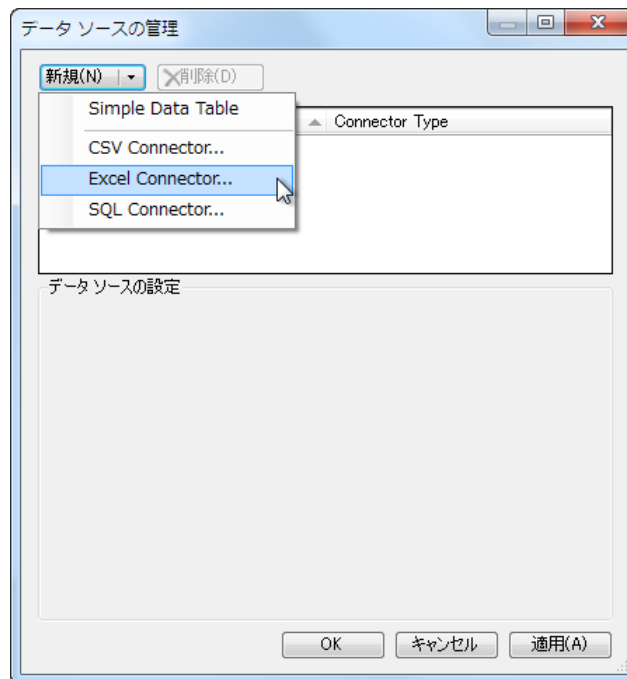


1 行目はヘッダーとして処理されます。1 行目の各セルは、列のキャプションとして解釈されます。



Excel ファイルを利用したデータ駆動型テスト

[データソースの管理]ダイアログで、[新規]-[Excel Connector]をクリックし、新規データコネクタを作成します。



新規 Excel コネクタの作成

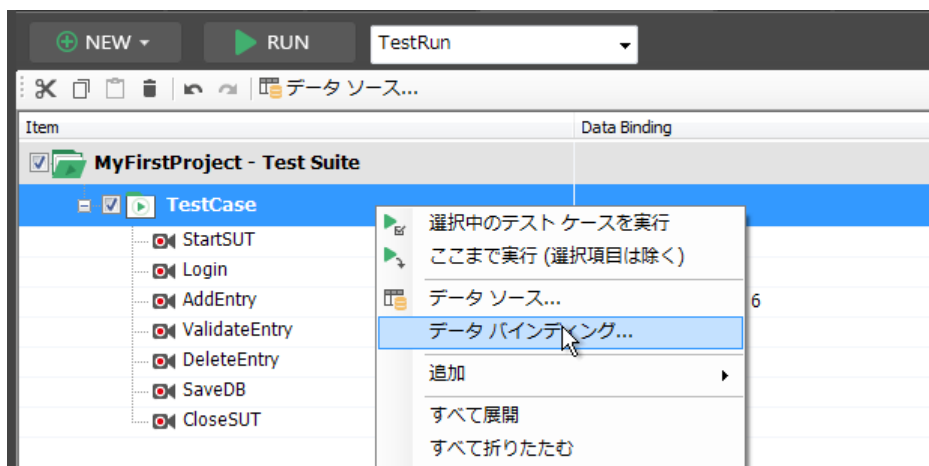
他のデータコネクタについては、別紙『Ranorex ユーザーガイド』の「データコネクタ」章を参照してください。



Excel ブック形式 xlsx を使用するかわりに、Excel バイナリブック形式 xlsb を使用することもできます。xlsb 形式は、Office 2007 以降でサポートされ、xlsx 形式よりも高速に処理がおこなわれます。

## テストデータと変数のバインド

現在のテストケース（最終的には、テストケースで使用されている“AddEntry”レコーディング）と Excel ファイルをバインドするには、テストケースのコンテキストメニューから[データバインディング]をクリックし、データバインディングダイアログを開きます。

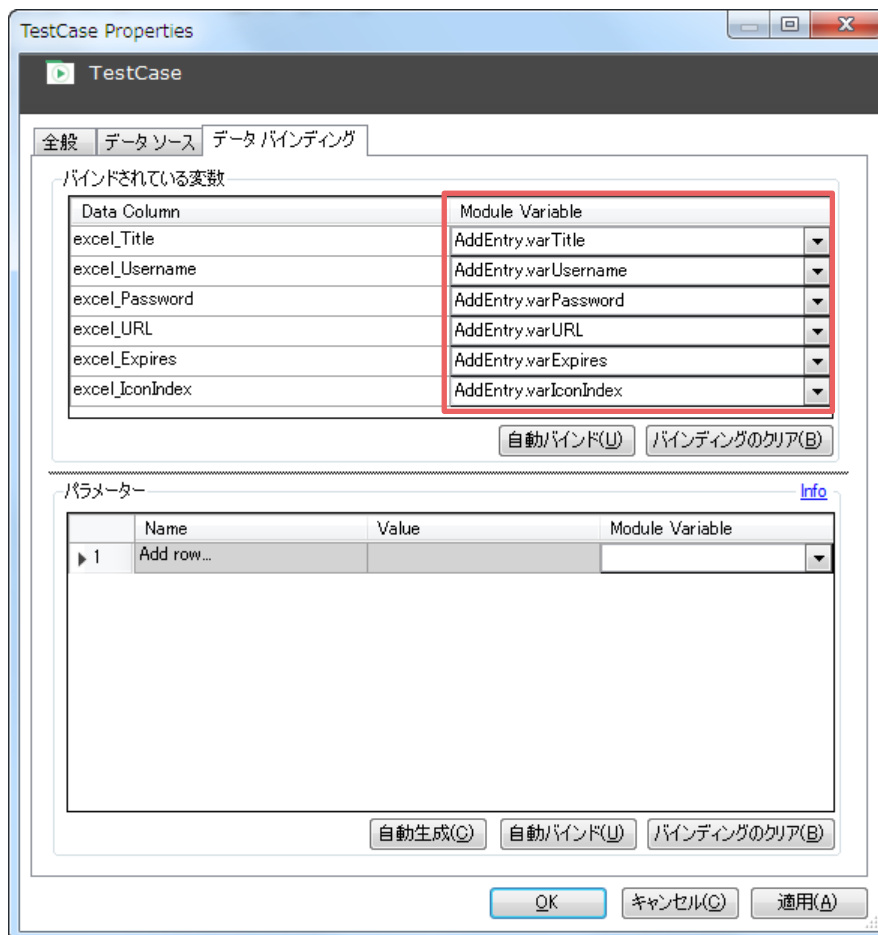


データバインディングダイアログのオープン



オープンしたダイアログの[データソース]タブで、先ほど作成した“ExcelConnector”が選択されていることを確認してください。

“Module Variable”列のリストから変数を選択することで、データテーブルの各カラムと、テストケースで使用されている変数をバインドできます。複数の変数を、ひとつのデータカラムにバインドすることもできます。



データカラムと変数のバインド



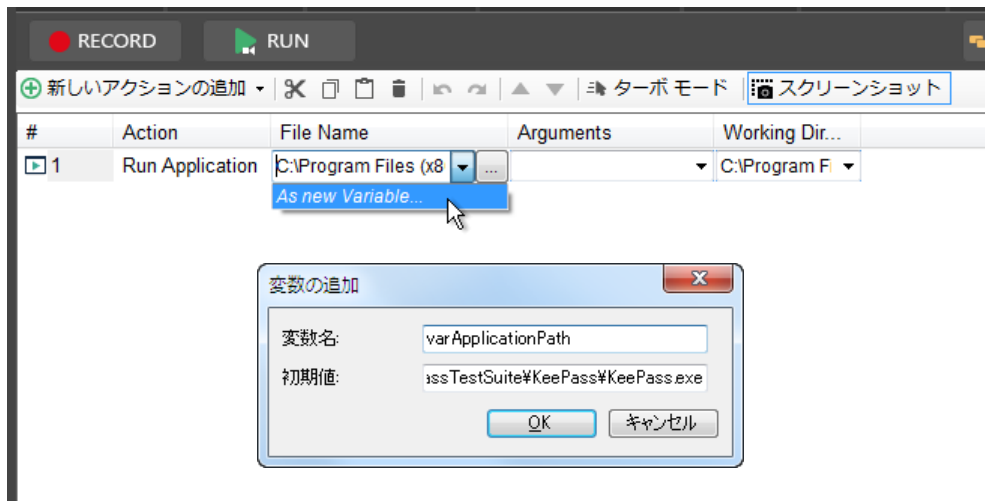
変数にバインドしたデータは、テストがテストスイートビューから実行される場合にのみ使用されます。モジュールを個別にテストする場合 (例: レコーダーから[Play]で実行) には、変数の初期値が使用されます。

## パラメーターと変数のバインド

パラメーターは、異なる環境下でテストを自動実行する場合に有用です。例えば、Windows 7 で自動テストをおこなう場合、Windows XP の場合とは異なるパス環境が必要です。パラメーターは、Ranorex Studio から作成し、変数とバインドすることができます。

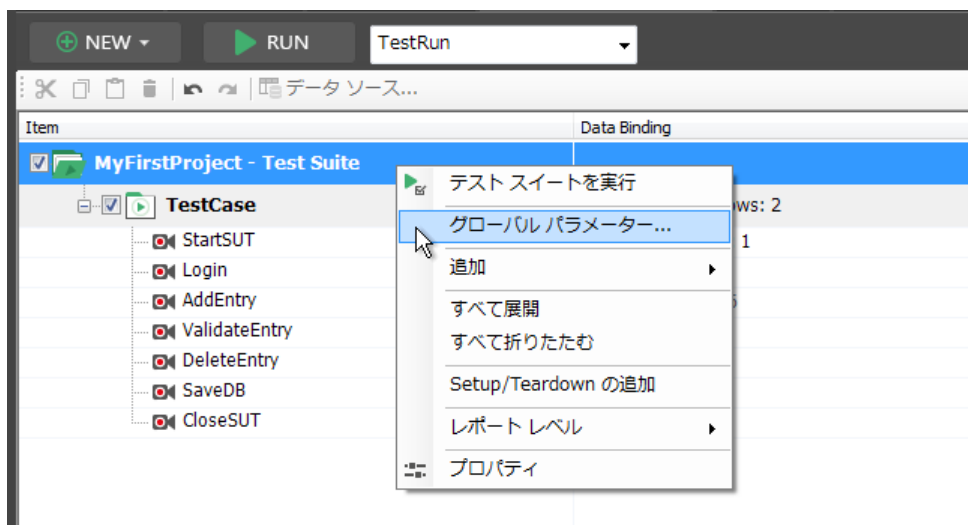
テストスイートとテストケースは、グローバルまたはローカルなパラメーターを宣言できます。これらのパラメーターは、モジュール変数にバインドできます。

テストケースは、“StartSUT”レコーディングモジュールから開始されます。このレコーディングは、KeePass アプリケーションの起動をおこなうアクションのみで構成されています。レコーディングをオープンし、新規に“varApplicationPath”変数を作成し、アプリケーションの実行パスを変数化します。

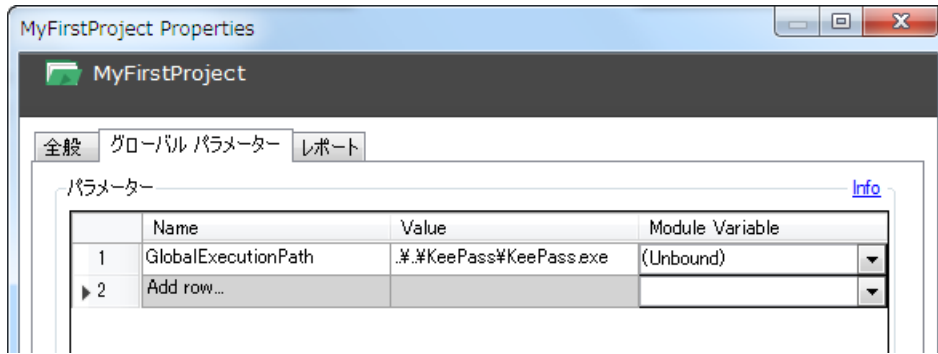


テスト対象システムの実行パスに関する変数の作成

テスト対象システムの実行パスは、テストスイートの複数のテストケースでバインドされる、テストスイートレベルのグローバルパラメーターとして宣言されます。テストスイートビューのコンテキストメニューから、パラメーターダイアログをオープンします。



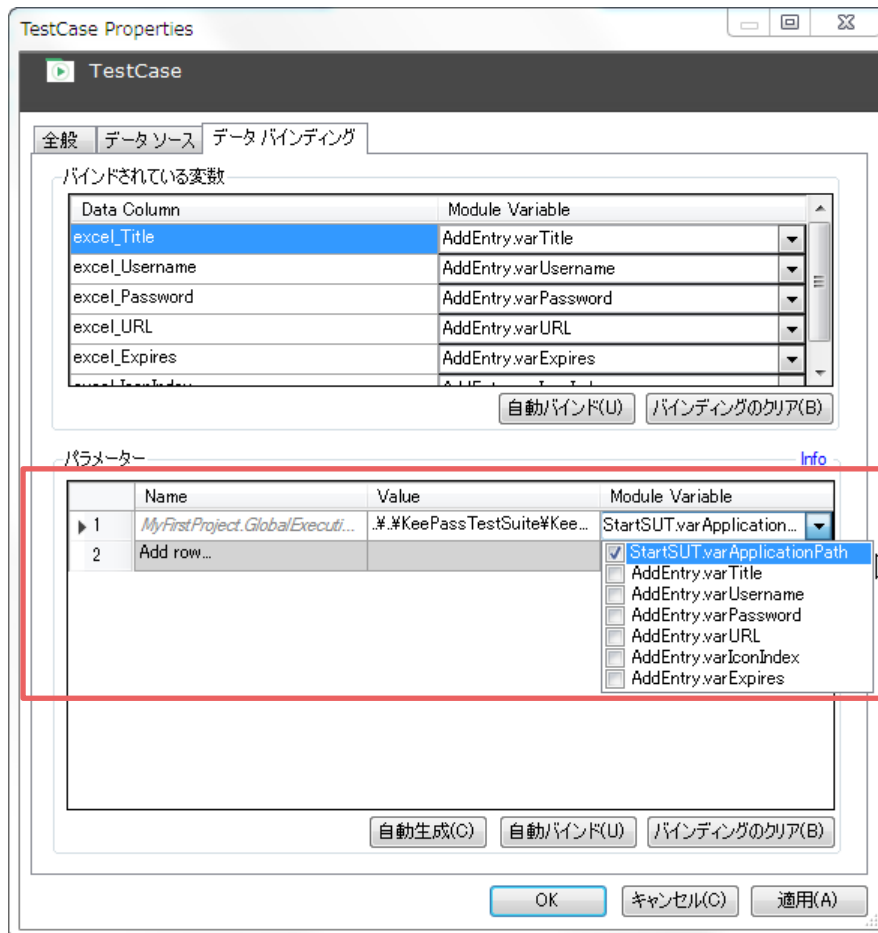
グローバルパラメーターダイアログのオープン



テストスイートに関するグローバルパラメーターの作成

“Name”列の最初のセルをクリックし、新規パラメーター名（例: GlobalExecutionPath）を入力します。次に、“Value”列に実行パスを入力します。ここでは、“StartSUT”レコーディングの変数ダイアログからデフォルトの実行パスをコピーし、ペーストします。[OK]ボタンをクリックし、変更を適用し、ダイアログを閉じます。

グローバルパラメーターと、以前のレコーディングで作成した変数のバインドの準備ができました。前節でおこなった手順とおなじように、テストケースのデータバインディングダイアログを開き、作成したパラメーターと変数のバインドをおこないます。



グローバルパラメーターと変数のバインド



テストスイートでモジュールにマウスオーバーすると、変数の一覧とバインドの有無がツールチップ表示されます。



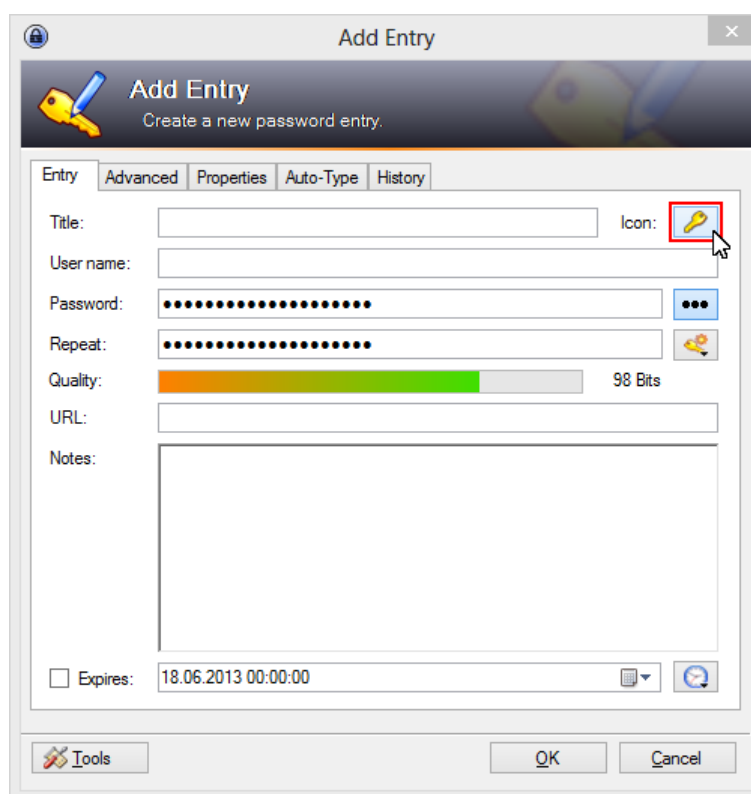
データ駆動テストケースを実行する前に、“AddEntry”レコーディングで、Excel ファイルのデータが正しくバインドされているか確認する必要があります。Title, Username, Password, URL, Expires については、テストの自動実行の問題となることはありませんが、IconIndex のように、リストからアイテムを選択するようなケースでは、初期状態ではこのリストが非表示であることによって、テストの自動実行で問題を起こすことが考えられます。このような状況の解決方法については、次節で学習します。



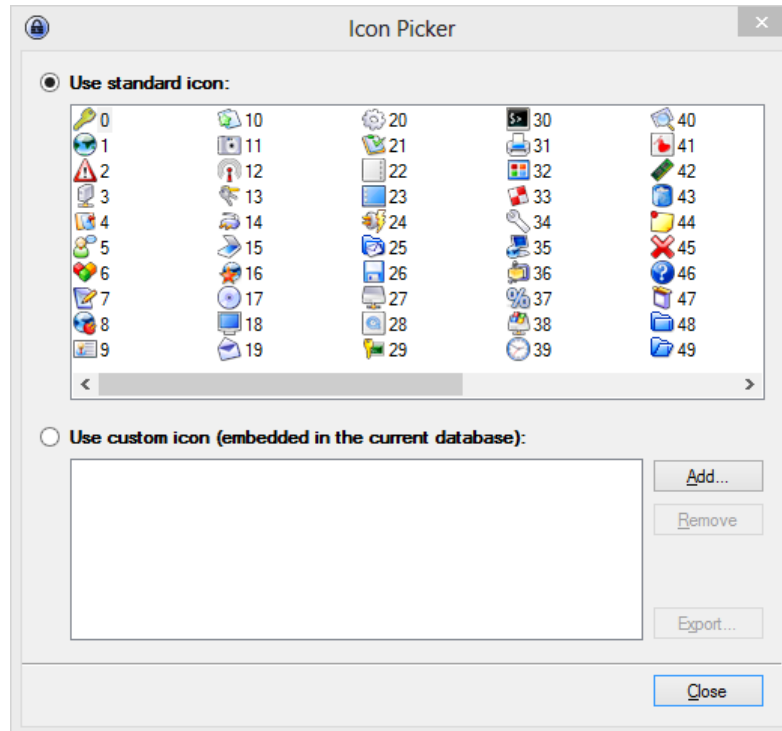
変数にバインドしたデータは、テストがテストスイートビューから実行される場合にのみ使用されます。モジュールを個別にテストする場合（例: レコーダーから[Play]で実行）には、変数の初期値が使用されます。

## 選択アクションの実行

KeePass アプリケーションには、“Icon”ラベルの横にあるボタンから、関連付けるアイコンをリスト選択する画面があります。



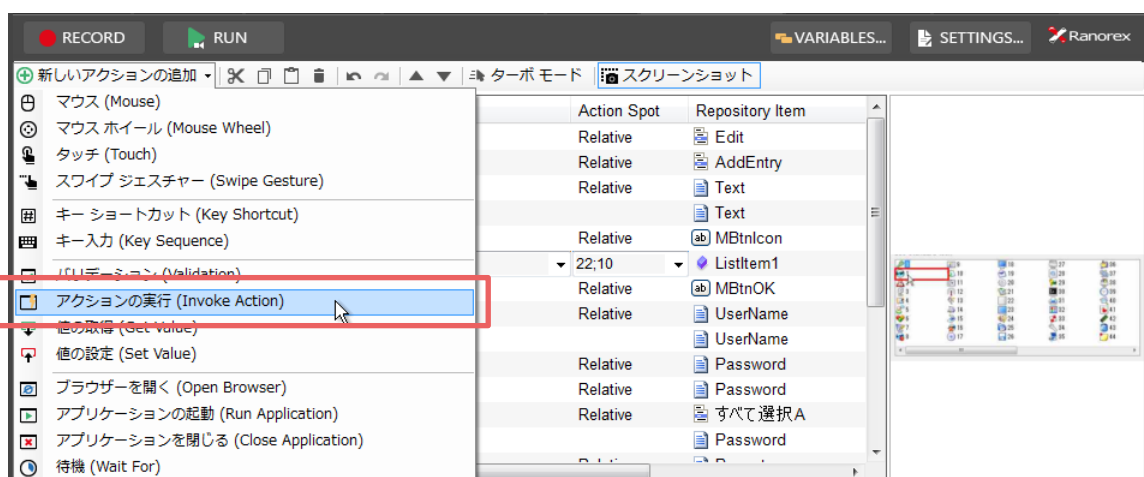
アイコンピッカーをオープンするボタン



アイコンの選択フォーム

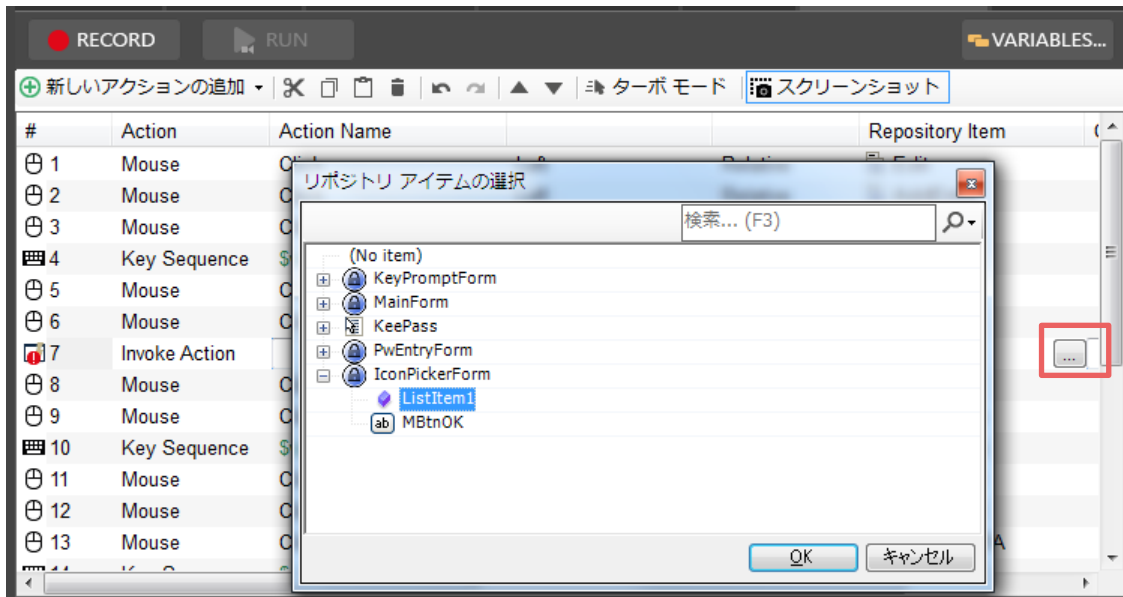
スクロールバーを使用しない場合、0 から 49 番目のアイコンまでが確認でき、これらのアイコンであれば、データ駆動テストで簡単に利用できます。49 番目以降のアイコンを自動テストで使用する場合、通常のクリックアクションでは、画面に表示されていないこれらのアイコンの選択は失敗します。リストでのアイコンの表示/非表示によらず、アイコンを選択するには、“**Invoke Action**”を使用します。

アクションテーブルからアイコンのクリックアクションを選択し、[新しいアクションの追加]ボタンをクリックし、新しいアクションを直後に追加します。



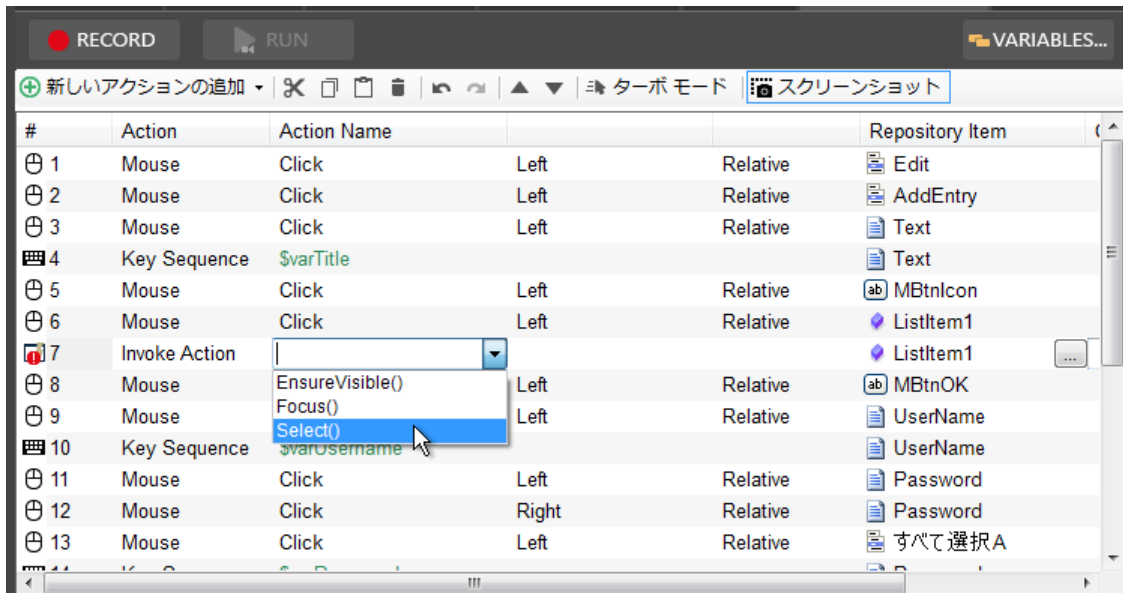
“Invoke Action”の追加

次に、Invoke Action で使用するリポジトリ要素を選択します。追加した Invoke Action の右側にあるボタンをクリックし、[リポジトリアイテムの選択]ダイアログを開き、“ListItem1”を選択します。



画面右のボタンをクリックして[リポジトリアイテムの選択]ダイアログをオープンし、リポジトリから“ListItem1”を選択

リポジトリアイテムを Invoke Action に割り当てると、そのリポジトリアイテムで選択可能なメソッドの一覧を確認できます。“ListItem1”では、“Select()”が選択できることが確認できます。

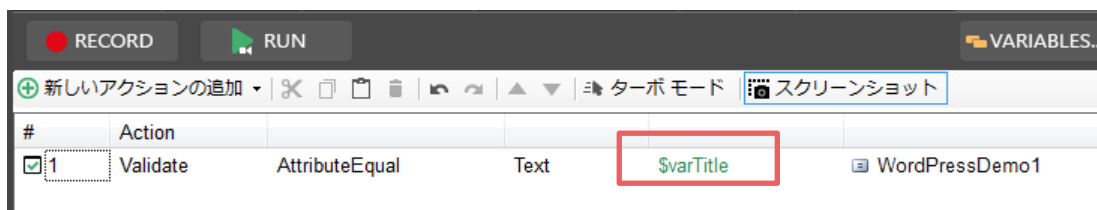


“Select”メソッドの選択

リストからアイテムを選択する Invoke Action を追加したので、直上にある、リストからアイテムを選択する最初のマウスクリックアクションを削除します。

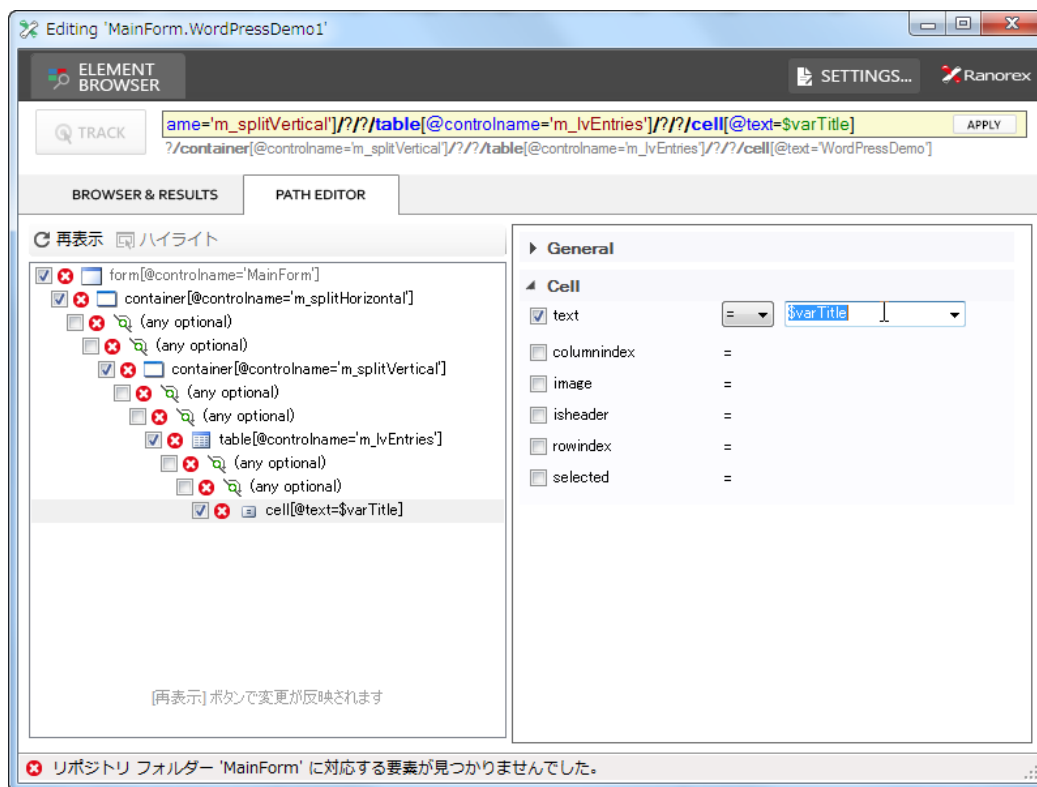
## データ駆動テストの実行

これまでの手順で、定数値“WordPressDemo”は変数で置き換えられているため、テストスイートの実行の前に、“ValidateEntry”でおこなう検証の期待値も変更する必要があります。最初に、検証アクションで使用される値について、変数 varTitle を作成します。



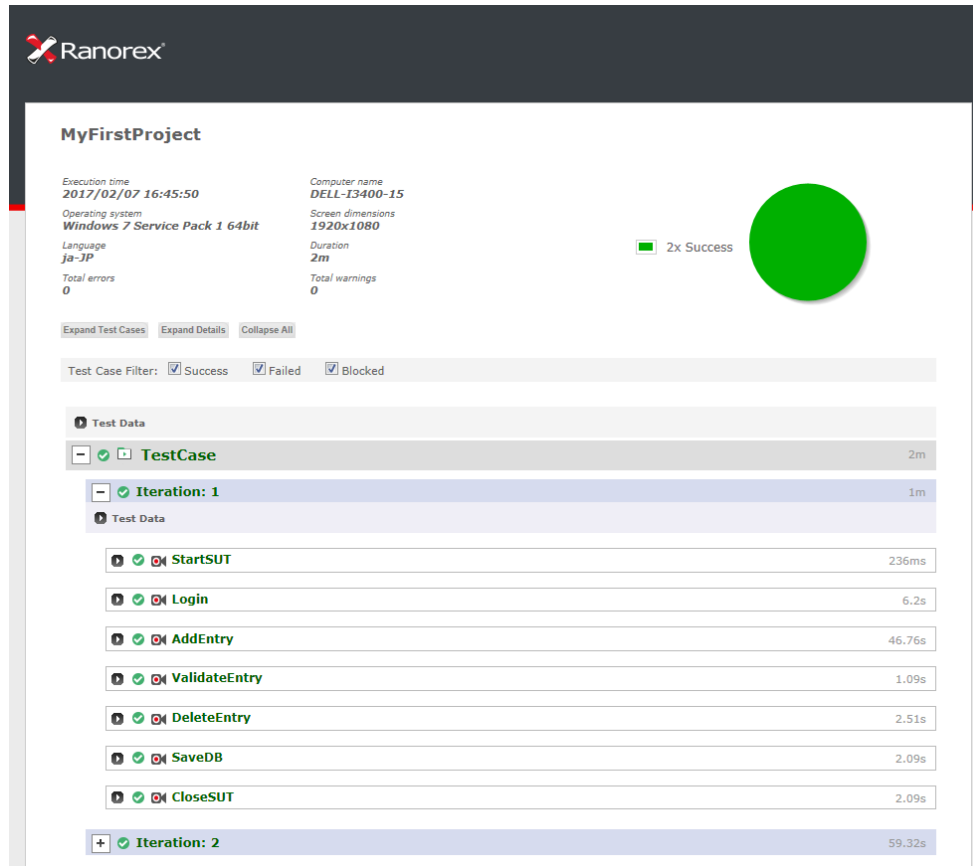
検証で使用される値の変数での置き換え

最後に、割り当てているリポジトリアイテムの RanoreXPath の Text 属性の値を、変数 **varTitle** で置き換えます。



RanoreXPath での変数での置き換え

テストスイートに戻り、[Run]ボタンをクリックし、データ駆動テストを実行します。テストの実行後、それぞれのイテレーションの結果がレポートされます。



**MyFirstProject**

Execution time: 2017/02/07 16:45:50  
Computer name: DELL-I3400-15  
Operating system: Windows 7 Service Pack 1 64bit  
Screen dimensions: 1920x1080  
Language: ja-JP  
Duration: 2m  
Total errors: 0  
Total warnings: 0

2x Success

Expand Test Cases | Expand Details | Collapse All

Test Case Filter:  Success  Failed  Blocked

**Test Data**

- TestCase** 2m
  - Iteration: 1** 1m
    - Test Data**
      - StartSUT** 236ms
      - Login** 6.2s
      - AddEntry** 46.76s
      - ValidateEntry** 1.09s
      - DeleteEntry** 2.51s
      - SaveDB** 2.09s
      - CloseSUT** 2.09s
  - Iteration: 2** 59.32s

テストケースのイテレーション結果のレポート

# おわりに

Windows アプリケーションのテストに関する詳細、Web アプリケーション、および、モバイルアプリケーションのテストの実施方法については、別紙『Ranorex ユーザーガイド』を参照してください。

## Ranorex に関するその他の情報

---

**Ranorex Blog** (英語サイト)

[www.ranorex.com/blog/](http://www.ranorex.com/blog/)

**Ranorex Forum** (英語サイト)

[www.ranorex.com/forum/](http://www.ranorex.com/forum/)

**FAQ** (英語サイト)

[www.ranorex.com/support/user-guide-20/faq.html](http://www.ranorex.com/support/user-guide-20/faq.html)

**Ranorex Blog** (日本語サイト)

[ranorex-jp.blogspot.jp](http://ranorex-jp.blogspot.jp)