Ranorex®

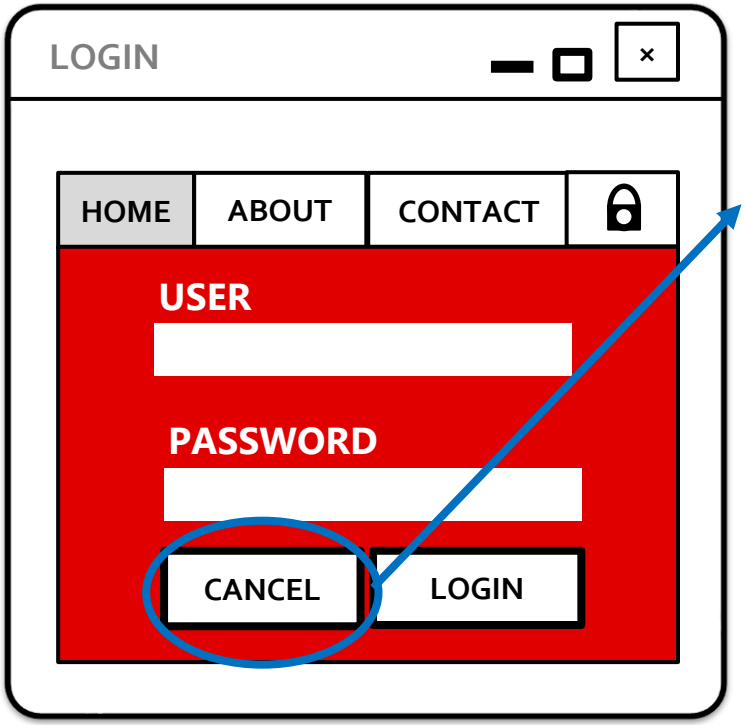
Self-Healing機能の概要

Self-Healing機能の概要

Self-Healing機能とは、1つのオブジェクトに対し、現在使用している
RanoreXPath とは別に、もう1つのパス情報を保持します。

テスト実行時、現在使用している RanoreXPath でオブジェクトが見つからない場合、
もう1つのパス情報によってオブジェクトの検索を行い、見つかったオブジェクトに対して
自動で操作が行われます。

※もう1つのパス情報 によるオブジェクト検索の結果、対象のオブジェクトが見つからない場合には、
オブジェクトが見つからないといったエラーが発生し、テストは中断されます。



例えば、CANCELボタンに対し、2つのRanoreXPathを保持します。

1 現在のRanoreXPath

/form/container/XXXX/XXXX/XXXX/button[@innertext='CANCEL']

2 もう1つのRanoreXPath

/form/XXXX//button[@innertext='CANCEL']

RanoreXPathの生成方法

RanoreXPathの生成は、**速度（Speed）**と**堅牢性（Robust）**の2軸を基に生成されます。

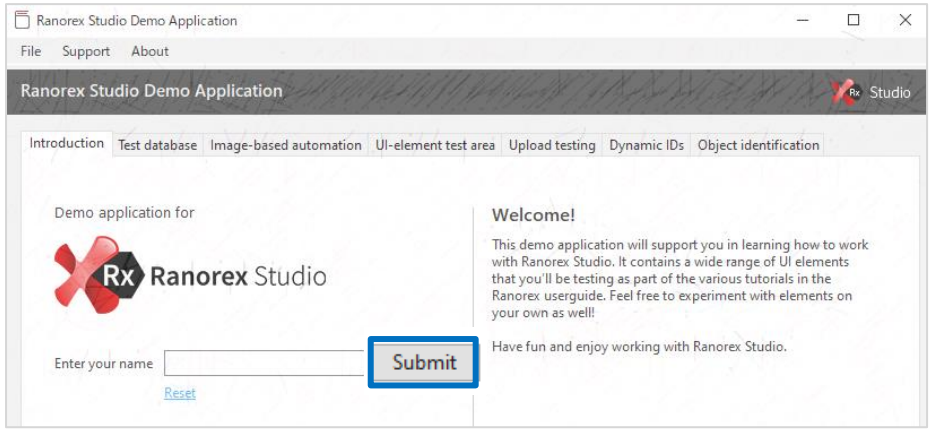
» **速度（Speed）**
速度（Speed）に重点を置いた場合、各階層の要素は省略されず、
多くの要素情報が含まれた RanoreXPath となります。
これにより、Ranorex が対象のオブジェクトをすぐに見つけ出すことができるため、
オブジェクトの検索速度が向上します。

» **堅牢性（Robust）**
堅牢性（Robust）に重点を置いた場合、多くの要素を使用せず、
対象のオブジェクトを認識できる最低限の要素のみを使用した RanoreXPath となります。
これにより、UI 要素の変更への耐性は高い（オブジェクトのヒット率が上がる）パスとなります。

速度/堅牢性については、以下のユーザーガイドに記載されています。
・RanoreXPath の設定 - 速度/堅牢性スライダー
https://www.ranorex.com/ja/help/latest/ranorex-studio-system-details/settings-configuration/advanced-settings-configurations/

RanoreXPathの具体例

下図のSubmit ボタンを例とし、速度重視型 と 堅牢性重視型 の RanoreXPath を確認します。



» **速度重視型のRanoreXPath**
/form[@controlname='RxMainFrame']/tabpagelist[@controlname='RxTabcontrol']/
tabpage[@controlname='RxTabIntroduction']/button[@controlname='btnSubmitUserName']

» **堅牢性重視型のRanoreXPath**
/form[@controlname='RxMainFrame']//button[@controlname='btnSubmitUserName']

上記の RanoreXPath はいずれもSubmitボタンを認識します。
速度重視型のRanoreXPath は多くの情報が使用され、堅牢性重視型のRanoreXPath は
必要最低限の情報のみが使用された RanoreXPath であることを確認できます。

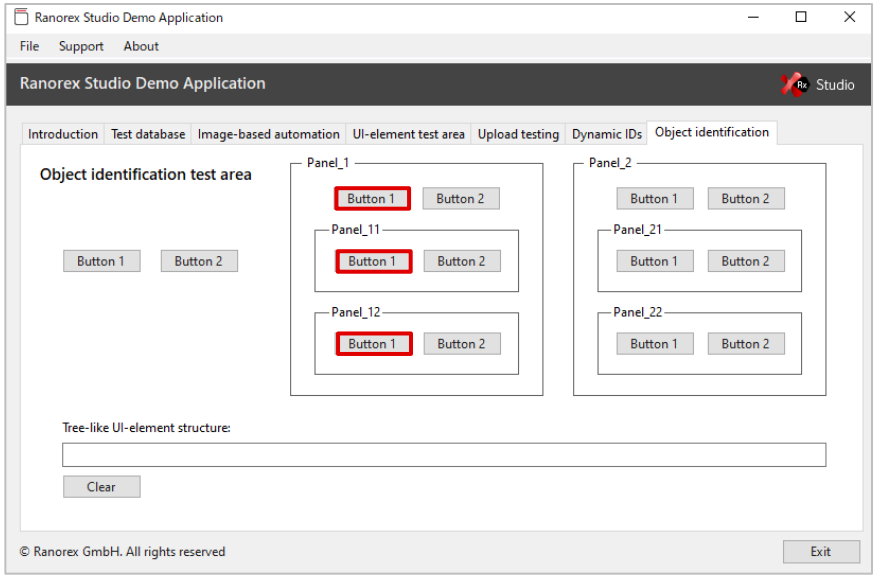
**Self-Healing機能では、もう1つの RanoreXPath として、
堅牢性重視型のRanoreXPath を保持しています。**

堅牢性重視型 RanoreXPath のメリット/デメリット

Self-Healing機能で使用される **堅牢性重視型のRanoreXPath** には、メリットとデメリットがあります。

» **メリット**
必要最低限の要素のみを使用した RanoreXPath となるため、
アプリケーション画面の構成が多少変更されても、そのままのパス情報でオブジェクトとして
認識できる可能性が高い RanoreXPath となります。

» **デメリット**
必要最低限の要素のみを使用した RanoreXPath となるため、オブジェクトを認識するために
必要な情報が少なく、オブジェクトの検索速度が低下します。
また、パスの中に含まれる情報が少ないことで、場合によっては1つのオブジェクトだけでなく、
複数のオブジェクトを認識してしまう RanoreXPath となる可能性があります。



１つの画面内に類似のオブジェクトが
複数存在する場合、堅牢性重視型のRanoreXPath
では複数のButton1を認識してしまう可能性があります。

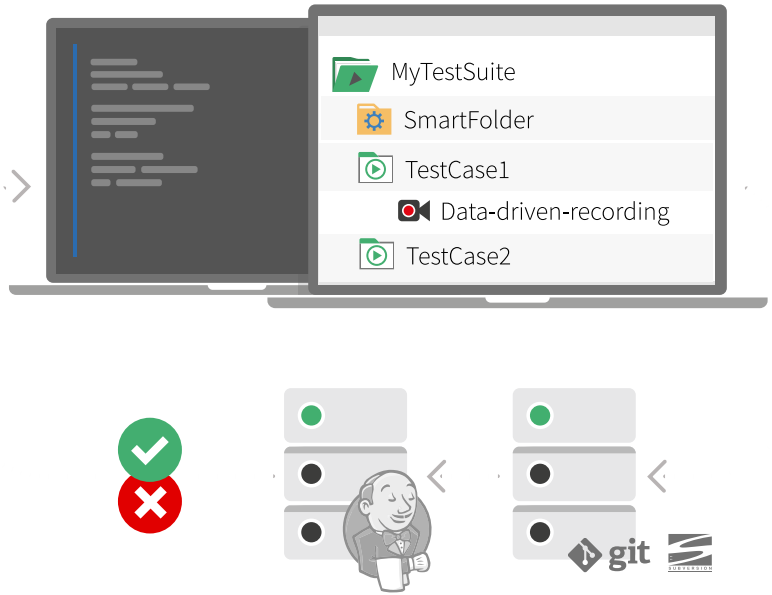
堅牢性が高すぎる RanoreXPath の場合、
オブジェクト認識に問題が発生する可能性もあるため、
最初に使用される RanoreXPath では、速度と堅牢性
のバランスがとれた RanoreXPath を使用します。

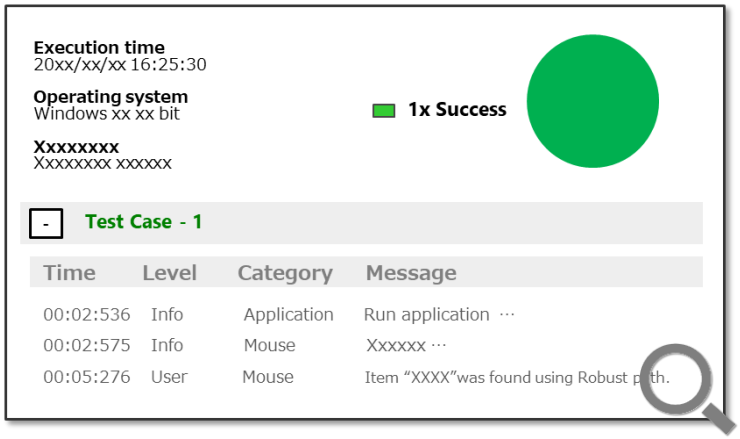
Self-Healing機能の目的

Self-Healing機能 を使用することで、テスト実行時にオブジェクトを検出できなかった場合、
より堅牢なオブジェクトパス（堅牢性重視のRanoreXPath）で再検出を試みることができます。

これにより、テスト対象アプリのUIの変更などにより、既存のオブジェクトパスで
認識できなくなった場合であっても、エラーでテストが中断されることを回避し、
より堅牢な RanoreXPath を使用してテストを完了させることを目的としています。

そのため、たとえば CIツールとの連携 により、定期的にテストを実行している場合にも
Self-Healing機能を使用することで、予期しないエラーでテストが中断されてしまうことを
回避することができます。





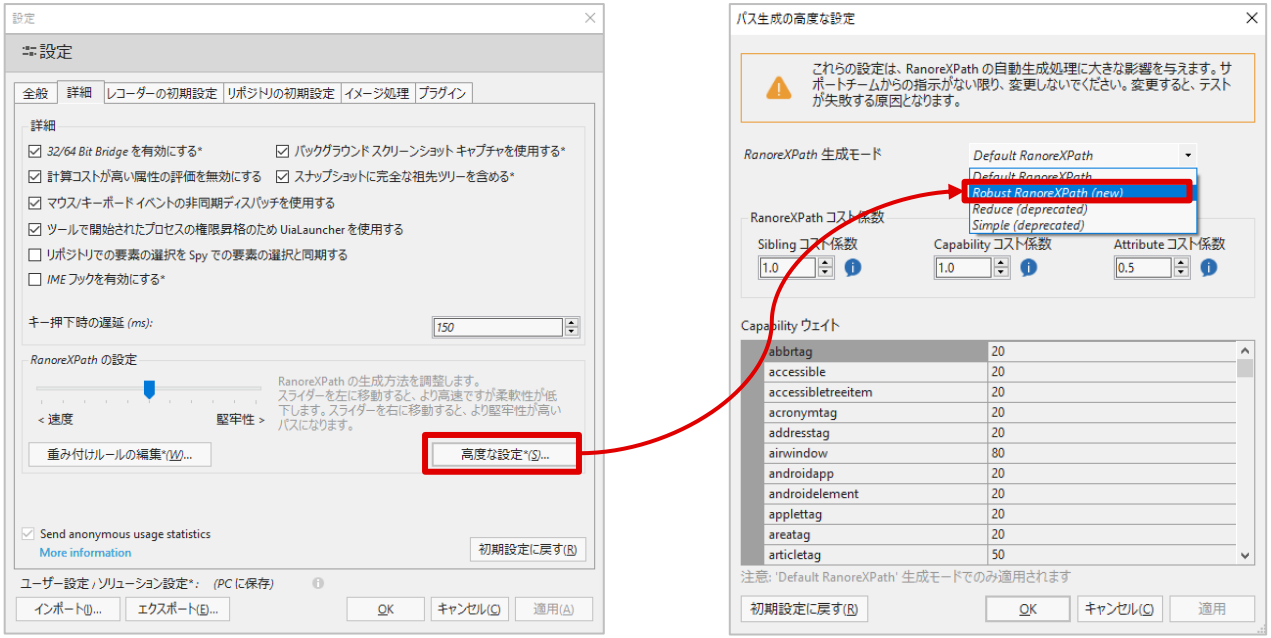
Item "XXXX"was found using Robust path.

Self-Healing機能の設定

以下の手順で設定を行います。

1. Ranorex Studio にて、アイコンのメニューから、 **設定(=)** をクリックします。

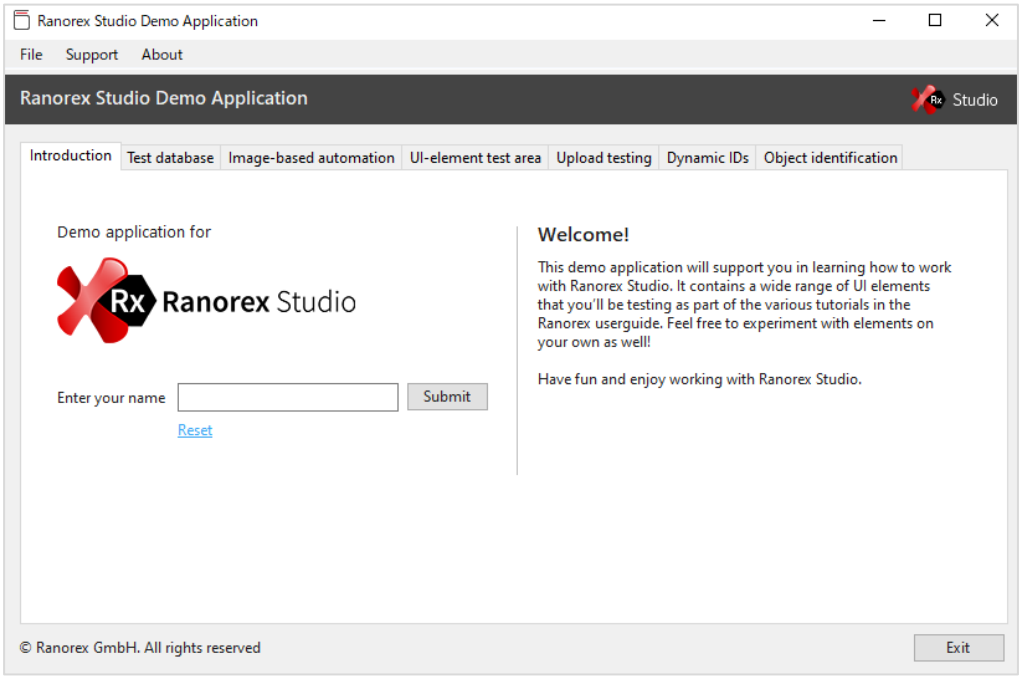
2. 詳細 タブを選択し、**高度な設定** をクリックします。
3. RanoreXPath 生成モード を **Robust RanoreXPath (new)** に変更します。



Self-Healing機能を使用した具体例

本資料では、開発元が提供しているサンプルのデスクトップアプリケーション（RxDemoApp.exe）を使用し、Self-Healing機能についてご説明します。

RxDemoApp.exe は、以下のユーザガイドからダウンロードできます。
https://www.ranorex.com/ja/help/latest/ranorex-studio-fundamentals/ranorize-20-minutes/3-plan-first-test/

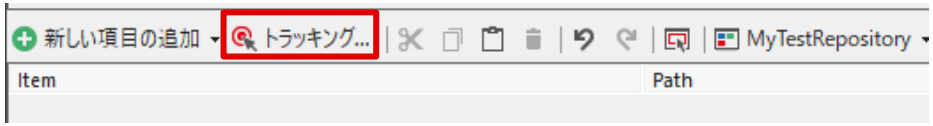


RxDemoApp.exe

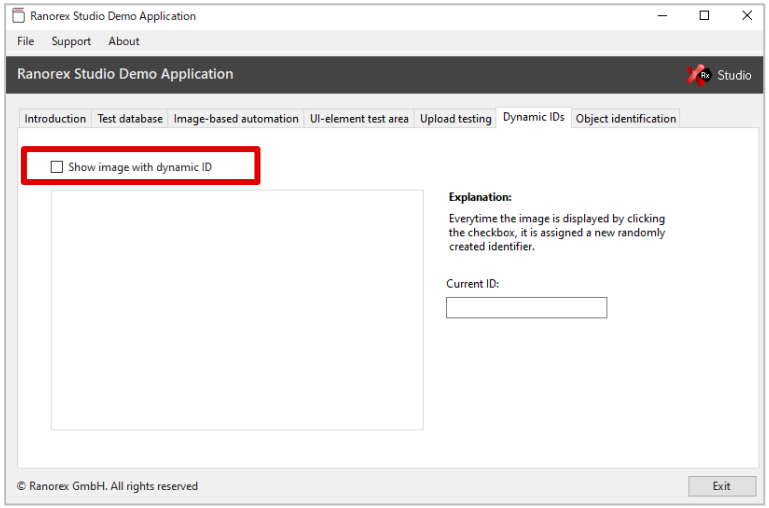
Self-Healing機能を使用した具体例

今回は、Dynamic IDs タブ内にある **Show image with dynamic ID** のチェックボックスを使用します。

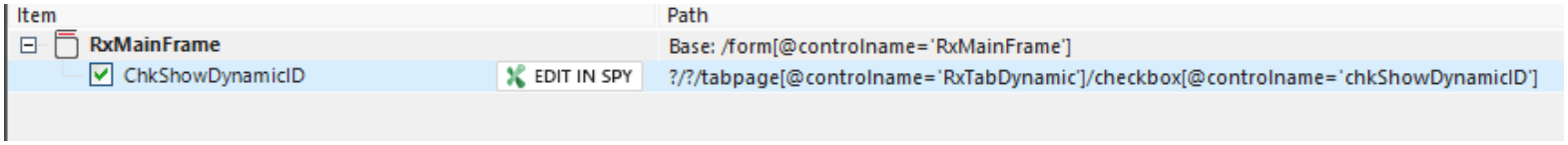
1. Ranorex Studioにて、任意のレコーディングモジュールを開きます。
2. 下側のリポジトリの上側にある **トラッキング** ボタンをクリックします。



3. **Show image with dynamic** ID のチェックボックスを赤枠で囲み、その上でクリックします。



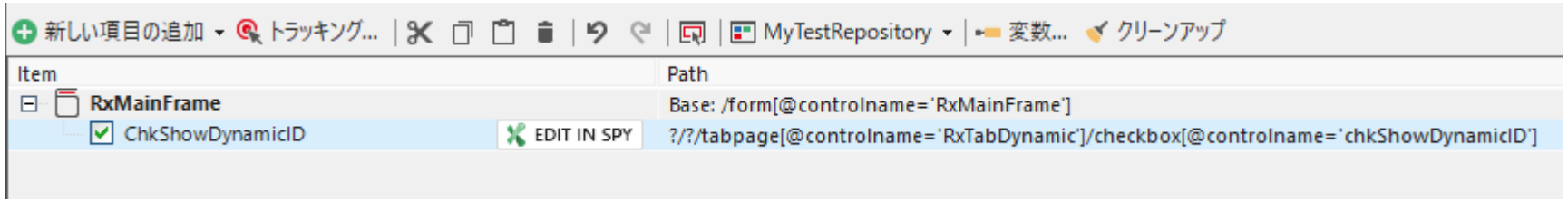
4. リポジトリ内に、**ChkShowDynamicID** オブジェクトが追加されます。



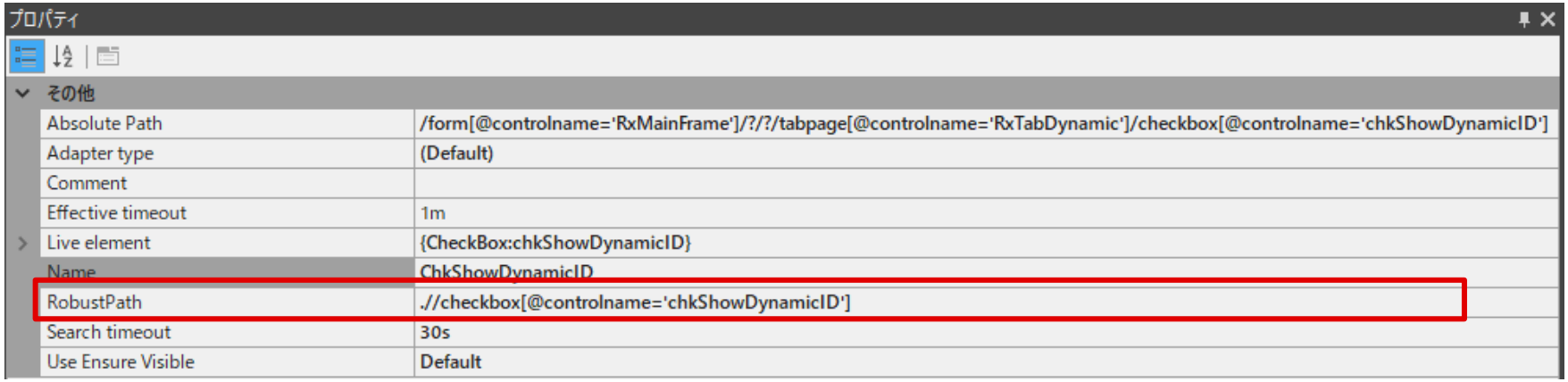
Self-Healing機能を使用した具体例

Show image with dynamic ID のチェックボックスを認識している
オブジェクト情報（ ChkShowDynamicID ）を確認します。

ChkShowDynamicID のプロパティ画面を開きます。
※プロパティ画面は、オブジェクトを選択し、F4キーをクリックすることで開かれます。



プロパティ画面の中には、 **RobustPath** が設定されています。
RobustPath は、ChkShowDynamicID を認識するために内部で保持している
もう1つの RanoreXPath となり、堅牢性を重視したRanoreXPath となります。

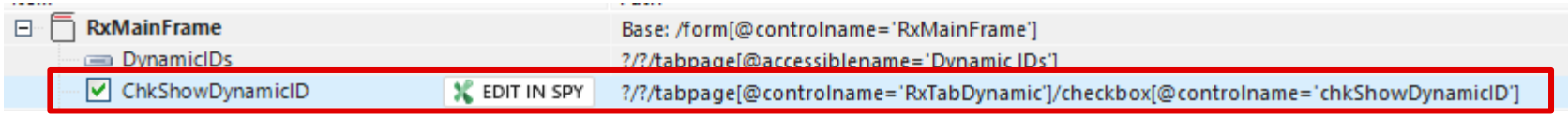


Self-Healing機能を使用した具体例

現在の RanoreXPath と、RobustPath について比較します。

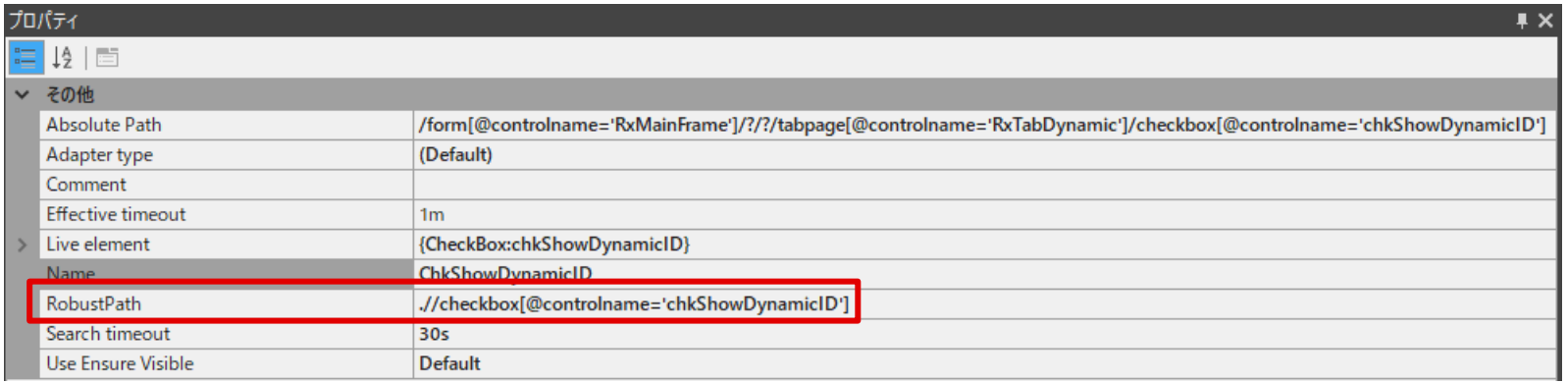
> 現在のRanoreXPath

/form[@controlname='RxMainFrame']/?/?/tabpage[@controlname='RxTabDynamic']/checkbox[@controlname='chkShowDynamicID']



> RobustPath（堅牢性重視型のRanoreXPath）

/form[@controlname='RxMainFrame'].//checkbox[@controlname='chkShowDynamicID']



2つのパスを比較すると、RobustPath は 現在の RanoreXPath よりも
少ない情報で生成された RanoreXPath となります。
現在の RanoreXPath ではオブジェクトが見つからない場合、RobustPath が使用されます。

まとめ

Ranorexでは、通常 **速度と堅牢性** を軸に、バランスのとれた RanoreXPath を生成します。

最初に使用する RanoreXPath においては、堅牢性重視型のRanoreXPath（RobustPath）ではなく、バランスのとれた RanoreXPath を使用することをお勧めします。

基本は バランスのとれた RanoreXPath を使用する一方で、
予期しないエラーによるテストの中断を防ぐために、Self-Healing機能を使用します。

これにより、現在使用している RanoreXPath ではオブジェクトが見つからない場合、
自動的に 堅牢性重視型のRanoreXPath（RobustPath）が使用され、
より安定したテストを実行することができます。

Self-Healing機能については、以下のユーザーガイドに記載されています。

・セルフヒーリング
https://www.ranorex.com/ja/help/latest/ranorex-studio-system-details/settings-configuration/self-healing/