

Kinaxis Custom Controls

Customization Documentation

Table of Contents

[Document Overview and Basics 3](#_Toc99361995)

[Purpose 3](#_Toc99361996)

[Delimitations 3](#_Toc99361997)

[Installation 4](#_Toc99361998)

[Overview and File List 4](#_Toc99361999)

[Preparations and preconditions 4](#_Toc99362000)

[Deployment procedure 4](#_Toc99362001)

[Usage 6](#_Toc99362002)

[Kinaxis RapidResponse Application 6](#_Toc99362003)

[Table Control 6](#_Toc99362004)

[TabControl 7](#_Toc99362005)

# Document Overview and Basics

## Purpose

This document describes the installation and usage of extensions made to the Tosca Tbox Engine.

## Delimitations

* This document does not cover any explanations concerning the setup and structure of the Tosca Workspace/Repository of the customer.
* For general information about the Tosca Tbox Engines or tool related automation questions please refer to the Tosca Manuals.

Change History:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Change Date | Changed By | Change Notes |
| 1.0 | 2022-03-15 | Marko Lipovac | Creation |
| 1.1 | 2022-03-24 | Marko Lipovac | Added TabControl |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Installation

## Overview and File List

The following components are part of the herein described Tricentis Tosca Extensions:

* Tricentis.Automation.Java.KinaxisControls.dll

## Preparations and preconditions

The following tools need to be installed and available to ensure proper functionality of the Tricentis Tosca Extensions. Please take particular notice of the corresponding application versions used to ensure proper functionality:

* Windows 10
* Tosca Testsuite 15.0
* Kinaxis RapidResponse H2112.2 Application

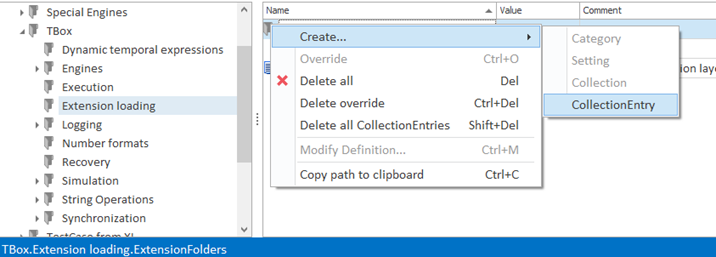
Important Note: Tricentis Tosca Custom Extensions are always tied to a particular SuT. Most of the time this SuT is equal to the environment used during development and test. Tosca Custom Extensions/Control will not always work for new environments. Supporting than the ones on hand during development is considered an additional requirement and needs to be covered with specific customizations otherwise it will not be considered during implementation.

Important Note: Tricentis Tosca Custom Extensions always need to be in sync with their supported versions of the Tosca Testsuite. If the versions of the supported environment changes, the Custom Extensions may also have to be changed accordingly.

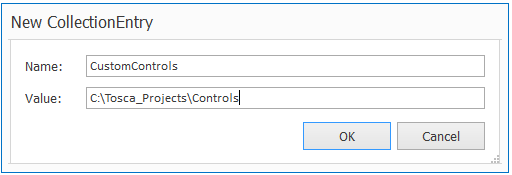
## Deployment procedure

The files delivered alongside with this document need to be installed on each TOSCA client to make sure every user can utilize them. Each client must have the following Tosca settings configured:

* TBox->Extension loading->ExtensionFolders
* RightClick and select Create…->CollectionEntry
  + Note: if this fails the first time try it again. Sometimes it throws an error the first time.



* Add name and file path in the dialog



* The file path must point to a local drive.
* Make sure you have placed the DLL file in the aforementioned filepath.
* Make sure the DLL is unblocked. (Right-click, properties, unblock).

(An alternative method is to simply drag and drop the DLL into “%TBOX\_HOME%”)

# Usage

## Kinaxis RapidResponse Application

### Table Control

|  |  |
| --- | --- |
| Scenario Description: | The user needs to steer the table in the application. The table can be found in the UNL Consensus Demand Planning screen for example. |
|  |  |
| Creating the Tosca Module: | Scan the control like a normal TBox table using the TBox Java Engine. Adapt the scanned Module by hand to add embedded controls underneath. Additionally adapt the HeaderRow Property to “3” since there are 3 rows of headers in the table. Also set the IgnoreInvisibleTableContent Property to “False”. |
|  |  |
| Using the Tosca Module: | Use standard TBox table steering to steer the scanned table control.    Input and verification of embedded controls can be done on the controls themselves. For example, to verify the contents of an embedded TextBox control use the embedded TextBox ModuleAttribute: |
|  |  |
|  |  |
| Sut Result: | The table will be steered. |
|  |  |

### TabControl

|  |  |
| --- | --- |
| Scenario Description: | The user needs to steer the TabControl in the application. |
|  |  |
| Creating the Tosca Module: | Scan the control like a normal TBox TabControl using the TBox Java Engine. The ValueRange will automatically be filled in. |
|  |  |
| Using the Tosca Module: | Use standard TBox TabControl steering to steer the scanned Tab control. |
|  |  |
|  |  |
| Sut Result: | The tab item will be steered. |
|  |  |