

Package: PhysioOpenSim (via r-universe)

October 3, 2026

Title Native OpenSim C++ Integration for PhysioExperiment

Version 0.1.4

Description Provides a native OpenSim C++ bridge for R via Rcpp. The package exposes OpenSim model-level operations to support biomechanics workflows in the Physio ecosystem. OpenSim linkage is optional at build time; when OpenSim is not detected, the package installs with informative runtime errors for OpenSim-dependent calls.

Depends R (>= 4.2)

Imports Rcpp

LinkingTo Rcpp

Suggests testthat (>= 3.2.0), withr, knitr, rmarkdown

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

SystemRequirements C++17, OpenSim (optional)

Config/testthat/edition 3

URL <https://github.com/x-biosignal/PhysioOpenSim>,
<https://x-biosignal.r-universe.dev/PhysioOpenSim>,
<https://x-biosignal.github.io/PhysioOpenSim/>

BugReports <https://github.com/x-biosignal/PhysioOpenSim/issues>

Repository <https://x-biosignal.r-universe.dev>

Date/Publication 2026-10-03 13:11:45 UTC

RemoteUrl <https://github.com/x-biosignal/PhysioOpenSim>

RemoteRef HEAD

RemoteSha 67346cb7706e834082ee9ea1e2e3f9ff187e34bd

Contents

opensimAvailable	2
opensimBuildConfig	3
opensimCLIAvailable	3
opensimCLIPath	4
opensimDiagnostics	4
opensimFinalizeConnections	5
opensimLoadModel	6
opensimModelComponents	6
opensimModelInitialize	7
opensimModelIsInitialized	8
opensimModelName	8
opensimModelSummary	9
opensimRunAnalyze	10
opensimRunCMC	11
opensimRunID	12
opensimRunIK	13
opensimRunRRA	14
opensimRunSO	15
opensimRunTool	16
opensimSaveModel	17
opensimSetModelName	17
opensimTemplatePath	18
opensimWriteAnalyzeSetupFromTemplate	19
opensimWriteCMCSetupFromTemplate	20
opensimWriteIDSetupFromTemplate	21
opensimWriteIKSetupFromTemplate	22
opensimWriteRRASetupFromTemplate	23
opensimWriteSOSetupFromTemplate	25
opensimWriteToolSetupFromTemplate	26
Index	28

opensimAvailable	<i>Check Whether Native OpenSim Support Is Available</i>
------------------	--

Description

Returns whether this package was compiled with OpenSim headers and libraries linked into the native code.

Usage

```
opensimAvailable()
```

Value

TRUE if OpenSim native support is available, otherwise FALSE.

Examples

```
opensimAvailable()
```

opensimBuildConfig	<i>Show Build Configuration for OpenSim Native Bridge</i>
--------------------	---

Description

Show Build Configuration for OpenSim Native Bridge

Usage

```
opensimBuildConfig()
```

Value

A list containing compilation status and build metadata.

Examples

```
cfg <- opensimBuildConfig()
cfg$package
cfg$opensim_enabled
```

opensimCLIAvailable	<i>Check Whether OpenSim CLI Is Available</i>
---------------------	---

Description

Check Whether OpenSim CLI Is Available

Usage

```
opensimCLIAvailable(cli = NULL)
```

Arguments

cli	Optional command or absolute path.
-----	------------------------------------

Value

TRUE if CLI command is resolvable, otherwise FALSE.

Examples

```
opensimCLIAvailable()
```

opensimCLIPath	<i>Resolve OpenSim CLI Command Path</i>
----------------	---

Description

Resolves the command path used to invoke OpenSim CLI.

Usage

```
opensimCLIPath(cli = NULL)
```

Arguments

cli	Optional command or absolute path. Defaults to OPENSIM_CLI environment variable, then "opensim-cmd".
-----	--

Value

A character scalar path to executable command.

Examples

```
## Not run:
opensimCLIPath()
opensimCLIPath("/usr/local/bin/opensim-cmd")

## End(Not run)
```

opensimDiagnostics	<i>Report the OpenSim Integration Environment</i>
--------------------	---

Description

Collects a single diagnostic snapshot of how this package can reach OpenSim: whether the native bridge was compiled in and by which detection method, whether the opensim-cmd CLI is on the search path, and the OpenSim / Simbody versions reported by whichever backend is available. Useful in CI logs and bug reports.

Usage

```
opensimDiagnostics(cli = NULL)
```

Arguments

cli	Optional opensim-cmd command or path passed to opensimCLIAvailable() / opensimCLIPath() .
-----	---

Value

An opensim_diagnostics list with elements package, native_available, detect_method, cli_available, cli_path, opensim_version, simbody_version, and backend (the backend that would be used by default: "native", "cli", or "none").

Examples

```
opensimDiagnostics()
```

```
opensimFinalizeConnections
```

Finalize OpenSim Model Connections

Description

Finalize OpenSim Model Connections

Usage

```
opensimFinalizeConnections(model)
```

Arguments

model A PhysioOpenSimModel external pointer.

Value

The input model handle (invisibly).

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimFinalizeConnections(model)

## End(Not run)
```

opensimLoadModel	<i>Load an OpenSim Model as a Native External Pointer</i>
------------------	---

Description

Load an OpenSim Model as a Native External Pointer

Usage

```
opensimLoadModel(path)
```

Arguments

path	Path to an OpenSim model file (.osim).
------	--

Value

A PhysioOpenSimModel external pointer handle.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimModelName(model)

## End(Not run)
```

opensimModelComponents	<i>List Core OpenSim Model Component Names</i>
------------------------	--

Description

List Core OpenSim Model Component Names

Usage

```
opensimModelComponents(model)
```

Arguments

model	A PhysioOpenSimModel external pointer.
-------	--

Value

Named list with vectors for body, joint, marker, muscle, and coordinate names.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
comps <- opensimModelComponents(model)
comps$bodies
comps$muscles

## End(Not run)
```

opensimModelInitialize

Initialize OpenSim Model System State

Description

Initialize OpenSim Model System State

Usage

```
opensimModelInitialize(model)
```

Arguments

model A PhysioOpenSimModel external pointer.

Value

The input model handle (invisibly).

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimModelInitialize(model)
opensimModelIsInitialized(model) # TRUE

## End(Not run)
```

opensimModelIsInitialized

Check Whether an OpenSim Model Handle Is Initialized

Description

Check Whether an OpenSim Model Handle Is Initialized

Usage

```
opensimModelIsInitialized(model)
```

Arguments

model A PhysioOpenSimModel external pointer.

Value

TRUE when model state has been initialized by OpenSim.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimModelIsInitialized(model)

## End(Not run)
```

opensimModelName

Get OpenSim Model Name

Description

Get OpenSim Model Name

Usage

```
opensimModelName(model)
```

Arguments

model A PhysioOpenSimModel external pointer.

Value

Character scalar model name.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimModelName(model)

## End(Not run)
```

opensimModelSummary *Summarize an OpenSim Model via Native C++ API*

Description

Loads an .osim model (or uses a loaded model handle) and returns structural summary information.

Usage

```
opensimModelSummary(path)
```

Arguments

path	Path to an OpenSim model file (.osim) or a PhysioOpenSimModel external pointer.
------	---

Value

A named list with model metadata (model_name, n_bodies, n_joints, n_markers, n_muscles, n_coordinates, total_mass, initialized).

Examples

```
## Not run:
info <- opensimModelSummary("gait2392.osim")
info$model_name
info$n_muscles

# Also works with a loaded model handle
model <- opensimLoadModel("gait2392.osim")
opensimModelSummary(model)

## End(Not run)
```

opensimRunAnalyze	<i>Run OpenSim Analyze Tool</i>
-------------------	---------------------------------

Description

Run OpenSim Analyze Tool

Usage

```
opensimRunAnalyze(  
  setup_file,  
  workdir = NULL,  
  cli = NULL,  
  extra_args = character(),  
  timeout_sec = 0L,  
  fail_on_error = TRUE,  
  execution = c("auto", "native", "cli")  
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".

Value

See [opensimRunTool\(\)](#).

Examples

```
## Not run:  
result <- opensimRunAnalyze("analyze_setup.xml")  
result$status  
  
## End(Not run)
```

opensimRunCMC

Run OpenSim Computed Muscle Control Tool

Description

Run OpenSim Computed Muscle Control Tool

Usage

```
opensimRunCMC(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli")
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".

Value

See [opensimRunTool\(\)](#).

Examples

```
## Not run:
result <- opensimRunCMC("cmc_setup.xml")
result$status

## End(Not run)
```

`opensimRunID`*Run OpenSim Inverse Dynamics Tool*

Description

Run OpenSim Inverse Dynamics Tool

Usage

```
opensimRunID(  
  setup_file,  
  workdir = NULL,  
  cli = NULL,  
  extra_args = character(),  
  timeout_sec = 0L,  
  fail_on_error = TRUE,  
  execution = c("auto", "native", "cli")  
)
```

Arguments

<code>setup_file</code>	Path to OpenSim tool setup XML.
<code>workdir</code>	Optional working directory.
<code>cli</code>	Optional command or absolute path (CLI mode only).
<code>extra_args</code>	Optional extra CLI args appended after setup file.
<code>timeout_sec</code>	Timeout in seconds for CLI execution (0 disables timeout).
<code>fail_on_error</code>	If TRUE, stop on non-zero exit status.
<code>execution</code>	Execution backend: "auto" (default), "native", "cli".

Value

See [opensimRunTool\(\)](#).

Examples

```
## Not run:  
result <- opensimRunID("id_setup.xml")  
result$status  
  
## End(Not run)
```

`opensimRunIK`*Run OpenSim Inverse Kinematics Tool*

Description

Run OpenSim Inverse Kinematics Tool

Usage

```
opensimRunIK(  
  setup_file,  
  workdir = NULL,  
  cli = NULL,  
  extra_args = character(),  
  timeout_sec = 0L,  
  fail_on_error = TRUE,  
  execution = c("auto", "native", "cli")  
)
```

Arguments

<code>setup_file</code>	Path to OpenSim tool setup XML.
<code>workdir</code>	Optional working directory.
<code>cli</code>	Optional command or absolute path (CLI mode only).
<code>extra_args</code>	Optional extra CLI args appended after setup file.
<code>timeout_sec</code>	Timeout in seconds for CLI execution (0 disables timeout).
<code>fail_on_error</code>	If TRUE, stop on non-zero exit status.
<code>execution</code>	Execution backend: "auto" (default), "native", "cli".

Value

See [opensimRunTool\(\)](#).

Examples

```
## Not run:  
result <- opensimRunIK("ik_setup.xml")  
result$status  
  
## End(Not run)
```

 opensimRunRRA

Run OpenSim Residual Reduction Algorithm Tool

Description

Run OpenSim Residual Reduction Algorithm Tool

Usage

```
opensimRunRRA(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli")
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".

Value

See [opensimRunTool\(\)](#).

Examples

```
## Not run:
result <- opensimRunRRA("rra_setup.xml")
result$status

## End(Not run)
```

opensimRunSO*Run OpenSim Static Optimization Tool*

Description

Run OpenSim Static Optimization Tool

Usage

```
opensimRunSO(  
  setup_file,  
  workdir = NULL,  
  cli = NULL,  
  extra_args = character(),  
  timeout_sec = 0L,  
  fail_on_error = TRUE,  
  execution = c("auto", "native", "cli")  
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".

Value

See [opensimRunTool\(\)](#).

Examples

```
## Not run:  
result <- opensimRunSO("so_setup.xml")  
result$status  
  
## End(Not run)
```

opensimRunTool

Run an OpenSim Tool Setup XML

Description

Executes a setup XML through native OpenSim bindings (when available) or through OpenSim CLI.

Usage

```
opensimRunTool(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli"),
  tool_type = c("tool", "ik", "id", "so", "analyze", "cmc", "rra")
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".
tool_type	Tool flavor token: "tool" (generic), "ik", "id", or "so", "analyze", "cmc", "rra".

Value

A named list with command metadata and logs.

Examples

```
## Not run:
result <- opensimRunTool("ik_setup.xml")
result$status
result$stdout

# Force CLI backend with extra arguments
opensimRunTool("ik_setup.xml", execution = "cli",
```

```

        extra_args = "--visualize")

## End(Not run)

```

opensimSaveModel	<i>Save OpenSim Model to File</i>
------------------	-----------------------------------

Description

Save OpenSim Model to File

Usage

```
opensimSaveModel(model, output_file)
```

Arguments

model	A PhysioOpenSimModel external pointer.
output_file	Output path for model XML.

Value

Normalized output path.

Examples

```

## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimSetModelName(model, "modified")
opensimSaveModel(model, "gait2392_modified.osim")

## End(Not run)

```

opensimSetModelName	<i>Set OpenSim Model Name</i>
---------------------	-------------------------------

Description

Set OpenSim Model Name

Usage

```
opensimSetModelName(model, name)
```

Arguments

model	A PhysioOpenSimModel external pointer.
name	New model name.

Value

The input model handle (invisibly).

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimSetModelName(model, "my_model")
opensimModelName(model) # "my_model"

## End(Not run)
```

opensimTemplatePath	<i>Path to a bundled OpenSim setup-template XML</i>
---------------------	---

Description

Returns the path of a setup-template XML shipped with **PhysioOpenSim**, suitable as the `template_file` argument of the `opensimWrite*SetupFromTemplate()` functions. Each template is a minimal OpenSim tool document whose fields (model, marker/coordinates, output, time range, external loads) are substituted by those writers.

Usage

```
opensimTemplatePath(
  tool = c("ik", "id", "so", "cmc", "rra", "analyze", "generic")
)
```

Arguments

tool	Which tool template to return: one of "ik", "id", "so", "cmc", "rra", "analyze", "generic".
------	---

Value

An absolute path to the bundled `<tool>_setup_template.xml`.

See Also

[opensimWriteIKSetupFromTemplate\(\)](#), [opensimWriteIDSetupFromTemplate\(\)](#), [opensimWriteSOSetupFromTemplate\(\)](#)

Examples

```
opensimTemplatePath("ik")
```

 opensimWriteAnalyzeSetupFromTemplate

Write Analyze Setup XML from Template

Description

Convenience wrapper around [opensimWriteToolSetupFromTemplate\(\)](#) for common Analyze-Tool tags.

Usage

```
opensimWriteAnalyzeSetupFromTemplate(  
    template_file,  
    output_file,  
    model_file,  
    coordinates_file = NULL,  
    external_loads_file = NULL,  
    time_range = NULL,  
    results_directory = NULL,  
    extra_fields = list(),  
    strict = TRUE  
)
```

Arguments

template_file	Path to Analyze template XML.
output_file	Path to output Analyze setup XML.
model_file	Path to .osim model.
coordinates_file	Optional path to coordinates/motion file.
external_loads_file	Optional path to external loads XML.
time_range	Optional numeric length-2 vector (c(start, end)).
results_directory	Optional output directory.
extra_fields	Optional named list of additional XML tag replacements.
strict	Passed to opensimWriteToolSetupFromTemplate() .

Value

See [opensimWriteToolSetupFromTemplate\(\)](#).

Examples

```
## Not run:
opensimWriteAnalyzeSetupFromTemplate(
  template_file = "analyze_template.xml",
  output_file = "analyze_setup.xml",
  model_file = "gait2392.osim",
  coordinates_file = "ik_output.mot"
)

## End(Not run)
```

```
opensimWriteCMCSetupFromTemplate
```

Write CMC Setup XML from Template

Description

Convenience wrapper around [opensimWriteToolSetupFromTemplate\(\)](#) for common CMCTool tags.

Usage

```
opensimWriteCMCSetupFromTemplate(
  template_file,
  output_file,
  model_file,
  desired_kinematics_file = NULL,
  external_loads_file = NULL,
  time_range = NULL,
  results_directory = NULL,
  extra_fields = list(),
  strict = TRUE
)
```

Arguments

template_file	Path to CMC template XML.
output_file	Path to output CMC setup XML.
model_file	Path to .osim model.
desired_kinematics_file	Optional path to desired kinematics file.
external_loads_file	Optional path to external loads XML.
time_range	Optional numeric length-2 vector (c(start, end)).
results_directory	Optional output directory.
extra_fields	Optional named list of additional XML tag replacements.
strict	Passed to opensimWriteToolSetupFromTemplate() .

Value

See [opensimWriteToolSetupFromTemplate\(\)](#).

Examples

```
## Not run:
opensimWriteCMCSetupFromTemplate(
  template_file = "cmc_template.xml",
  output_file = "cmc_setup.xml",
  model_file = "gait2392.osim",
  desired_kinematics_file = "ik_output.mot"
)

## End(Not run)
```

opensimWriteIDSetupFromTemplate

Write ID Setup XML from Template

Description

Convenience wrapper around [opensimWriteToolSetupFromTemplate\(\)](#) for common Inverse Dynamics tags.

Usage

```
opensimWriteIDSetupFromTemplate(
  template_file,
  output_file,
  model_file,
  coordinates_file,
  output_gen_force_file,
  external_loads_file = NULL,
  time_range = NULL,
  lowpass_cutoff_frequency_for_coordinates = NULL,
  results_directory = NULL,
  extra_fields = list(),
  strict = TRUE
)
```

Arguments

template_file	Path to ID template XML.
output_file	Path to output ID setup XML.
model_file	Path to .osim model.
coordinates_file	Path to coordinates/motion file (typically .mot).

output_gen_force_file Path to generalized force output file (.sto).
 external_loads_file Optional path to external loads XML.
 time_range Optional numeric length-2 vector (c(start, end)).
 lowpass_cutoff_frequency_for_coordinates Optional numeric cutoff.
 results_directory Optional output directory.
 extra_fields Optional named list of additional XML tag replacements.
 strict Passed to [opensimWriteToolSetupFromTemplate\(\)](#).

Value

See [opensimWriteToolSetupFromTemplate\(\)](#).

Examples

```
## Not run:
opensimWriteIDSetupFromTemplate(
  template_file = "id_template.xml",
  output_file = "id_setup.xml",
  model_file = "gait2392.osim",
  coordinates_file = "ik_output.mot",
  output_gen_force_file = "id_output.sto",
  time_range = c(0.5, 2.0)
)

## End(Not run)
```

opensimWriteIKSetupFromTemplate

Write IK Setup XML from Template

Description

Convenience wrapper around [opensimWriteToolSetupFromTemplate\(\)](#) for common Inverse Kinematics tags.

Usage

```
opensimWriteIKSetupFromTemplate(
  template_file,
  output_file,
  model_file,
  marker_file,
  output_motion_file,
```

```

    time_range = NULL,
    results_directory = NULL,
    extra_fields = list(),
    strict = TRUE
)

```

Arguments

template_file	Path to IK template XML.
output_file	Path to output IK setup XML.
model_file	Path to .osim model.
marker_file	Path to marker trajectory file (typically .trc).
output_motion_file	Path to IK output motion file (.mot).
time_range	Optional numeric length-2 vector (c(start, end)).
results_directory	Optional output directory.
extra_fields	Optional named list of additional XML tag replacements.
strict	Passed to opensimWriteToolSetupFromTemplate() .

Value

See [opensimWriteToolSetupFromTemplate\(\)](#).

Examples

```

## Not run:
opensimWriteIKSetupFromTemplate(
  template_file = "ik_template.xml",
  output_file = "ik_setup.xml",
  model_file = "gait2392.osim",
  marker_file = "walking.trc",
  output_motion_file = "ik_output.mot",
  time_range = c(0.5, 2.0)
)

## End(Not run)

```

opensimWriteRRASetupFromTemplate

Write RRA Setup XML from Template

Description

Convenience wrapper around [opensimWriteToolSetupFromTemplate\(\)](#) for common RRATool tags.

Usage

```

opensimWriteRRASetupFromTemplate(
    template_file,
    output_file,
    model_file,
    desired_kinematics_file = NULL,
    external_loads_file = NULL,
    output_model_file = NULL,
    time_range = NULL,
    results_directory = NULL,
    extra_fields = list(),
    strict = TRUE
)

```

Arguments

template_file	Path to RRA template XML.
output_file	Path to output RRA setup XML.
model_file	Path to .osim model.
desired_kinematics_file	Optional path to desired kinematics file.
external_loads_file	Optional path to external loads XML.
output_model_file	Optional path to adjusted model output (.osim).
time_range	Optional numeric length-2 vector (c(start, end)).
results_directory	Optional output directory.
extra_fields	Optional named list of additional XML tag replacements.
strict	Passed to opensimWriteToolSetupFromTemplate() .

Value

See [opensimWriteToolSetupFromTemplate\(\)](#).

Examples

```

## Not run:
opensimWriteRRASetupFromTemplate(
    template_file = "rra_template.xml",
    output_file = "rra_setup.xml",
    model_file = "gait2392.osim",
    desired_kinematics_file = "ik_output.mot",
    output_model_file = "gait2392_rra.osim"
)

## End(Not run)

```

`opensimWriteSOSetupFromTemplate`*Write SO Setup XML from Template*

Description

Convenience wrapper around [opensimWriteToolSetupFromTemplate\(\)](#) for common Static Optimization tags.

Usage

```
opensimWriteSOSetupFromTemplate(  
    template_file,  
    output_file,  
    model_file,  
    coordinates_file,  
    external_loads_file = NULL,  
    time_range = NULL,  
    results_directory = NULL,  
    extra_fields = list(),  
    strict = TRUE  
)
```

Arguments

<code>template_file</code>	Path to SO template XML.
<code>output_file</code>	Path to output SO setup XML.
<code>model_file</code>	Path to .osim model.
<code>coordinates_file</code>	Path to coordinates/motion file (typically .mot).
<code>external_loads_file</code>	Optional path to external loads XML.
<code>time_range</code>	Optional numeric length-2 vector (c(start, end)).
<code>results_directory</code>	Optional output directory.
<code>extra_fields</code>	Optional named list of additional XML tag replacements.
<code>strict</code>	Passed to opensimWriteToolSetupFromTemplate() .

Value

See [opensimWriteToolSetupFromTemplate\(\)](#).

Examples

```
## Not run:
opensimWriteSOSetupFromTemplate(
  template_file = "so_template.xml",
  output_file = "so_setup.xml",
  model_file = "gait2392.osim",
  coordinates_file = "ik_output.mot",
  time_range = c(0.5, 2.0)
)

## End(Not run)
```

```
opensimWriteToolSetupFromTemplate
```

Write OpenSim Tool Setup XML from Template

Description

Replaces tag values in an existing OpenSim setup XML template. This function is tool-agnostic and can be used for IK/ID/SO/RRA/CMC templates.

Usage

```
opensimWriteToolSetupFromTemplate(
  template_file,
  output_file,
  fields,
  strict = TRUE
)
```

Arguments

template_file	Path to template XML.
output_file	Path to output XML.
fields	Named list of replacement values keyed by XML tag name.
strict	If TRUE, error when a tag in fields is missing in template.

Value

A named list with output_file, applied_tags, and missing_tags.

Examples

```
# Create a minimal template in a temp file
tpl <- tempfile(fileext = ".xml")
writeLines(c(
  "<OpenSimDocument>",
  "  <model_file>Unassigned</model_file>",
  "  <time_range>0 1</time_range>",
  "</OpenSimDocument>"
), tpl)

out <- tempfile(fileext = ".xml")
result <- opensimWriteToolSetupFromTemplate(
  template_file = tpl,
  output_file = out,
  fields = list(model_file = "my_model.osim", time_range = "0.5 2.0")
)
result$applied_tags
readLines(result$output_file)
```

Index

`opensimAvailable`, [2](#)
`opensimBuildConfig`, [3](#)
`opensimCLIAvailable`, [3](#)
`opensimCLIAvailable()`, [4](#)
`opensimCLIPath`, [4](#)
`opensimCLIPath()`, [4](#)
`opensimDiagnostics`, [4](#)
`opensimFinalizeConnections`, [5](#)
`opensimLoadModel`, [6](#)
`opensimModelComponents`, [6](#)
`opensimModelInitialize`, [7](#)
`opensimModelIsInitialized`, [8](#)
`opensimModelName`, [8](#)
`opensimModelSummary`, [9](#)
`opensimRunAnalyze`, [10](#)
`opensimRunCMC`, [11](#)
`opensimRunID`, [12](#)
`opensimRunIK`, [13](#)
`opensimRunRRA`, [14](#)
`opensimRunSO`, [15](#)
`opensimRunTool`, [16](#)
`opensimRunTool()`, [10–15](#)
`opensimSaveModel`, [17](#)
`opensimSetModelName`, [17](#)
`opensimTemplatePath`, [18](#)
`opensimWriteAnalyzeSetupFromTemplate`,
[19](#)
`opensimWriteCMCSetupFromTemplate`, [20](#)
`opensimWriteIDSetupFromTemplate`, [21](#)
`opensimWriteIDSetupFromTemplate()`, [18](#)
`opensimWriteIKSetupFromTemplate`, [22](#)
`opensimWriteIKSetupFromTemplate()`, [18](#)
`opensimWriteRRASetupFromTemplate`, [23](#)
`opensimWriteSOSetupFromTemplate`, [25](#)
`opensimWriteSOSetupFromTemplate()`, [18](#)
`opensimWriteToolSetupFromTemplate`, [26](#)
`opensimWriteToolSetupFromTemplate()`,
[19–25](#)