

# Living Standards Measurement Study



*The World Bank's Flagship Household Survey Program*



**THE WORLD BANK**

IBRD • IDA | WORLD BANK GROUP

Development Economics • Data

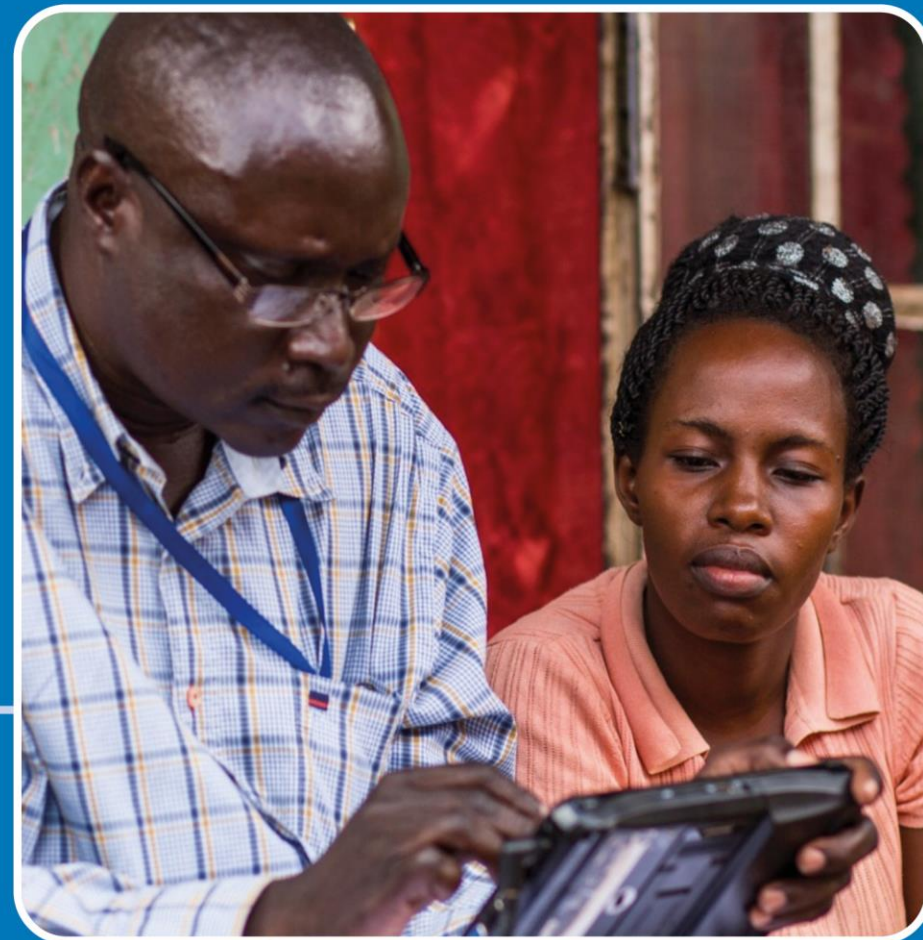
# adodown

Bringing down friction for  
package development

Kristoffer Bjärkefur

Arthur Shaw

LSMS & DIME Analytics  
World Bank



# adodown

The commands introduced in this presentation are all part of the package **adodown** which is available through SSC.

Installation : `ssc install adodown`

Documentation : <https://lsms-worldbank.github.io/adodown/>

Source code : <https://github.com/lsms-worldbank/adodown>

# Motivation

Getting started developing and publishing Stata packages was hard and slow.

*At least for me it was...*

Getting good at it and learning best practices took even longer time.

*At least for me it did...*

## Objectives

- **Automate manual tasks.** Start from a standardized template. Provide single-command workflows. Build bundle for SSC. Deploy docs to the web.
- **Make the docs easier to write, and to read.** For developers, simple Markdown. For users, help files and web doc.
- **Simplify development for community contributors.** Concentrate on coding. Worry less—or not at all—about pkg files, SSC rules, or SMCL.



# Set up the project

Make best practices the easiest option

# Don't do things manually

Command	Task
<b>ad_setup</b>	Set up folders for a new package
<b>ad_command</b>	Add command to package – creates an ado-file, a mdhlp-file, and a test file
<b>ad_sthlp</b>	Convert the mdhlp-files to sthlp-files
<b>ad_publish</b>	Prepare the package for publication on SSC

Humans are not good at doing repetitive manual tasks

–

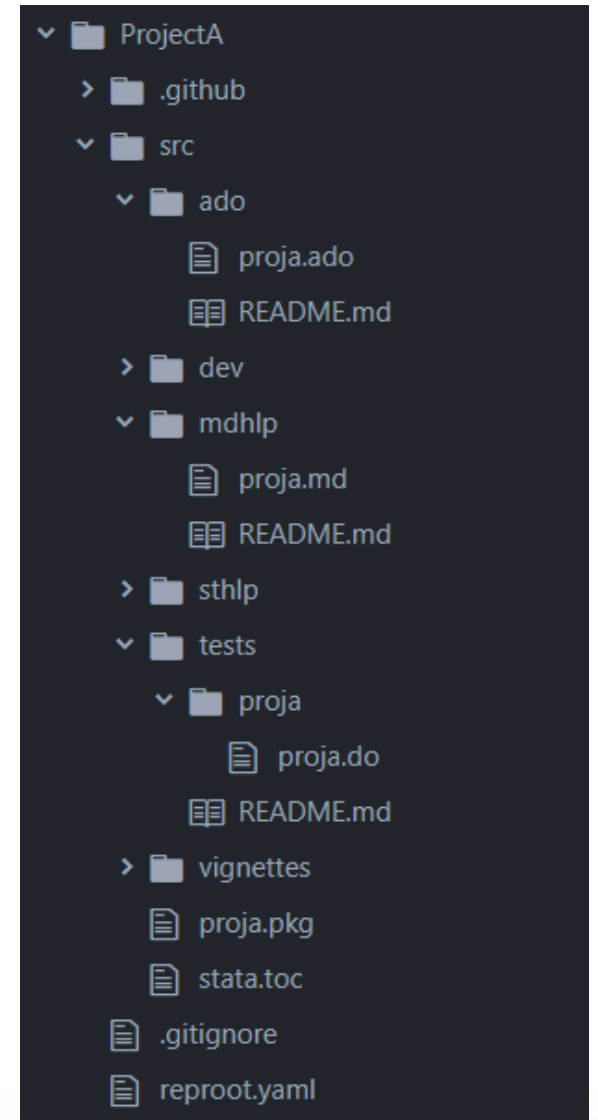
We make mistakes and cut corners

# Set up the package folder

\* Set up an new adodown-styled package

```
local myfolder "C:\Users\wb462869\GitHub\ProjectA"  
ad_setup, adfolder("`myfolder'") name("proja")
```

- Standardized folder structures with README files facilitating collaboration
- During setup, the first command named the same as the package is created
- Sets up the **.pkg** and the **stata.toc** file making self-hosted publication simple
- Adds optional files that might help
  - **reproot** root-file
  - **.github**





# Add a command to the package

Set up new required files and update existing files

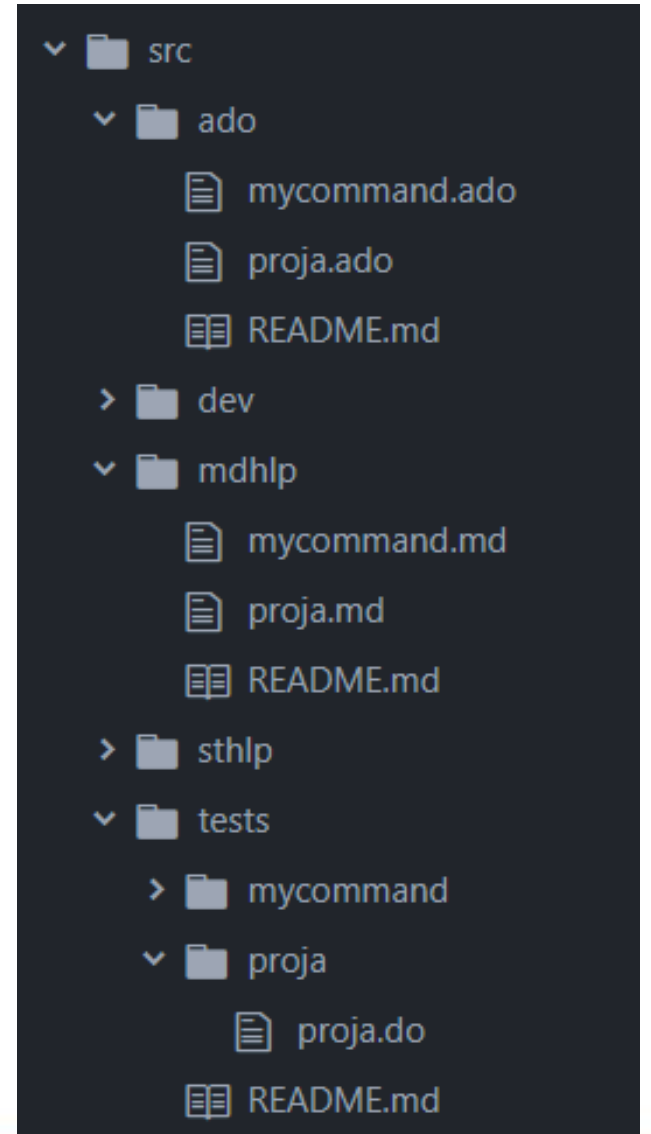
# Add a new command

\* Add a new command to the package

```
local myfolder "C:\Users\wb462869\GitHub\ProjectA"  
ad_command create mycommand, adfolder("`myfolder'") pkg("proja")
```

When adding a new command to the package:

- The .pkg file is updated
- The following files are created:
  - src/ado/mycommand.ado
  - src/mdhlp/mycommand.md
  - src/sthlp/mycommand.sthlp
  - src/test/mycommand/mycommand.do



# Create template files - ado-file

```
*! version XX XXXXXXXXX ADAUTHORNAME ADCONTACTINFO
```

```
cap program drop    mycommand  
    program define mycommand
```

```
qui {  
    version /* ADD VERSION NUMBER HERE */
```

```
    * Update the syntax. This is only a placeholder to make the command run  
    syntax [anything]
```

```
    //TODO : implement command here
```

```
}  
end
```

# Create template files – unit-test do-file

```
* Use reproject to manage root path
reproject, project("proja") roots("clone") prefix("proja_")

* Use locals for all non-root paths
local testfldr "${adwn_clone}/src/tests"

* Use the /dev-env folder as a dev environment
cap mkdir    "`testfldr'/dev-env"
reproject using "`testfldr'/dev-env"

* Make sure reproject is installed also in the dev environment
cap which reproject
if _rc == 111 ssc install reproject

* Make sure the version of proja in the dev environment
* is up to date with all edits.
cap net uninstall proja
net install proja, from("${proja_clone}/src") replace

*****
* Run tests
* Test basic case of the command mycommand
mycommand
```

Please share your best practices for how you test your commands and packages

# Create template files - mdhlp-file

```
# Title

__mycommand__ - This command is used for short description.

# Syntax
__mycommand__ , __**opt**ion1__(string)

| _options_ | Description |
|-----|-----|
| __**opt**ion1__(string) | Short description of option1 |

# Description
<!-- Longer description of the intended use of the command and best practices related to the usage -->

# Options
<!-- Longer description (paragraph length) of all options, their intended use case and best practices related to them -->
__**opt**ion1__(string) is used for xyz. Longer description (paragraph length) of all options, their intended use case and best practices related to them.

# Stored results
<!-- Document all results this command returns as either r-class, e-class or s-class macros -->

# Examples
<!-- A couple of examples to help the user get started and a short explanation of each of them -->

# Feedback, bug reports and contributions
<!-- Guidelines for how to provide feedback, bug reports or contributions for this package. Include an email or links to GitHub repo -->

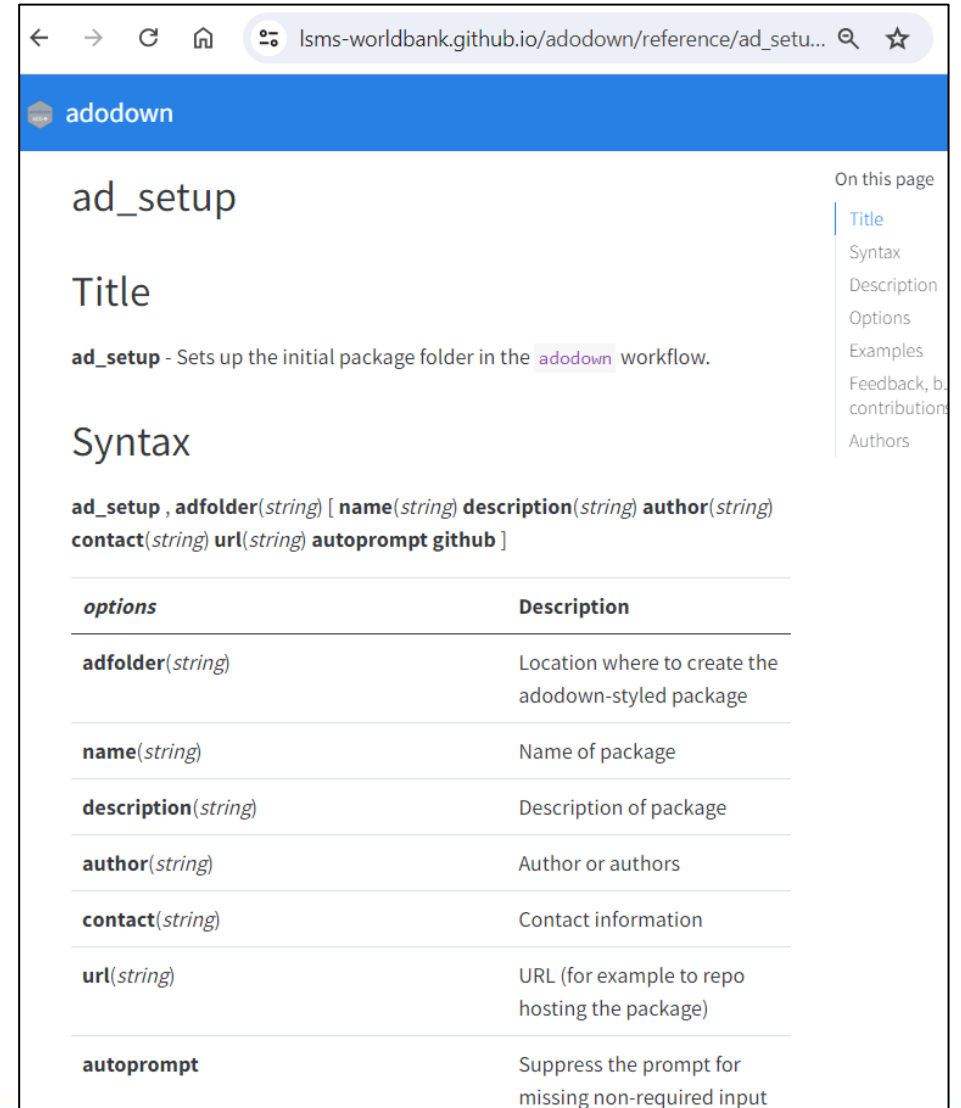
# Authors
<!-- A couple of examples to help the user get started and a short explanation of each of them -->
```

# Write documentation

Use mdhlp to document only once

# Write once in Markdown

- Across the data science community, **Markdown** is the standard tool for documentation
- In adodown, documentation is written in **mdhlp** – which is a flavor of markdown that **adodown** converts to **.sthlp** files
- Easy to learn and is compatible with other tools – such as web documentation tools



The screenshot shows a web browser window with the URL `lsms-worldbank.github.io/adodown/reference/ad_setu...`. The page title is "adodown" and the main heading is "ad\_setup". The content includes a "Title" section with the text "ad\_setup - Sets up the initial package folder in the adodown workflow." and a "Syntax" section with the command `ad_setup , adfolder(string) [ name(string) description(string) author(string) contact(string) url(string) autoprompt github ]`. A table of options is provided below, listing parameters like `adfolder`, `name`, `description`, `author`, `contact`, `url`, and `autoprompt` with their respective descriptions.

options	Description
<code>adfolder(string)</code>	Location where to create the adodown-styled package
<code>name(string)</code>	Name of package
<code>description(string)</code>	Description of package
<code>author(string)</code>	Author or authors
<code>contact(string)</code>	Contact information
<code>url(string)</code>	URL (for example to repo hosting the package)
<code>autoprompt</code>	Suppress the prompt for missing non-required input

# Deploy effortlessly to STHLP and HTML

Web Documentation



Documentation Source Code



Help File Documentation

→ ↻ 🏠 🔍 lsms-worldbank.github.io/adodown/reference/ad\_se

## adodown

### ad\_setup

#### Title

**ad\_setup** - Sets up the initial package folder in the `adodown` workflow.

#### Syntax

**ad\_setup**, **adfolder**(string) [ **name**(string) **description**(string) **author**(string) **contact**(string) **url**(string) **autoprompt** **github** ]

options	Description
<b>adfolder</b> (string)	Location where to create the adodown-styled package
<b>name</b> (string)	Name of package
<b>description</b> (string)	Description of package
<b>author</b> (string)	Author or authors

```
# Title
__ad_setup__ - Sets up the initial package folder in the `adodown` workflow.

# Syntax
__ad_setup__ , __adf__older__(string) [ __n__ame__(string)
__d__escription__(string) __a__uthor__(string) __c__ontact__(st
__u__rl__(string) __auto__prompt__ __git__hub__ ]

|_options_| Description |
|-----|-----|
| __adf__older__(string) | Location where to create the adodown-sty
| __n__ame__(string) | Name of package |
| __d__escription__(string) | Description of package |
| __a__uthor__(string) | Author or authors |
| __c__ontact__(string) | Contact information |
| __u__rl__(string) | URL (for example to repo hosting the pac
| __auto__prompt__ | Suppress the prompt for missing non-requ
| __git__hub__ | Add GitHub files without prompting ||

Read the `adodown` package's [web-documentation](https://lsms-worldbank.gi
) where you find all helpfiles for the commands in this package, as well a
guides and best-practices related to the commands in this package.

# Description

This command creates the initial folder template needed to write and docum
command packages in the `adodown` workflow.

This workflow makes it easier to create Stata command and packages both re
distribution on SSC and from a GitHub repository. This workflow also makes
```

help file for **ad\_setup**

---

#### Title

**ad\_setup** - Sets up the initial package folder in the `adodown` workflow.

#### Syntax

**ad\_setup**, **adfolder**(string) [ **name**(string) **description**(string) **author**(string) **contact**(string) **url**(string) **autoprompt** **github** ]

options	Description
<b>adfolder</b> (string)	Location where to create the adodown-styled package
<b>name</b> (string)	Name of package
<b>description</b> (string)	Description of package
<b>author</b> (string)	Author or authors
<b>contact</b> (string)	Contact information
<b>url</b> (string)	URL (for example to repo hosting the package)
<b>autoprompt</b>	Suppress the prompt for missing non-required input
<b>github</b>	Add GitHub files without prompting

Read the `adodown` package's `web-documentation` where you find all helpfiles for the commands in this package, as well as articles with guides and best-practices related to



# mdhlp syntax

Markdown syntax	Description	SMCL syntax	Comment
	Paragraph	Using <code>{pstd}</code> / <code>{p_end}</code> tags	In markdown a paragraph is not defined by a character. Instead, a paragraph is defined as text between empty lines with no other formatting (part from inline formatting). In the <i>Title</i> and <i>Syntax</i> sections the <code>{phang}</code> tag is used instead of <code>{pstd}</code> .
<code>#</code>	Header level 1	Using <code>{title:}</code> tag	
<code>##</code>	Header level 2	Using <code>{dltab:}</code> tag	No formatting applied if using more <code>#</code> . As in <code>###</code> , <code>####</code> etc.
<code>__</code>	Inline bold font	Using <code>{bf:}</code> tag	Ignored within code formatting
<code>** **</code>	Inline underlined font	Using <code>{ul:}</code> tag	Ignored unless used for text already in bold font
<code>__</code>	Inline italic font	Using <code>{it:}</code> tag	Ignored in bold font
<code>~ ~</code>	Inline code font	Using <code>{inp:}</code> tag	All other inline formatting is ignored within the <code>~</code> tags
<code>*** / ****</code>	Multiline code block	Using <code>{input}</code> / <code>{text}</code> tags	Ignores all formatting within the <code>***</code> tags
<code>[ ] ( )</code>	Hyperlinks	Using <code>{browse link:text}</code>	May not be combined with other

- **mdhlp** syntax is a subset of the markdown syntax
- **mdhlp** does not support everything you can do in markdown nor everything you can do in SMCL
- See full documentation at <https://lsms-worldbank.github.io/adodown/articles/mdhlp-syntax.html>

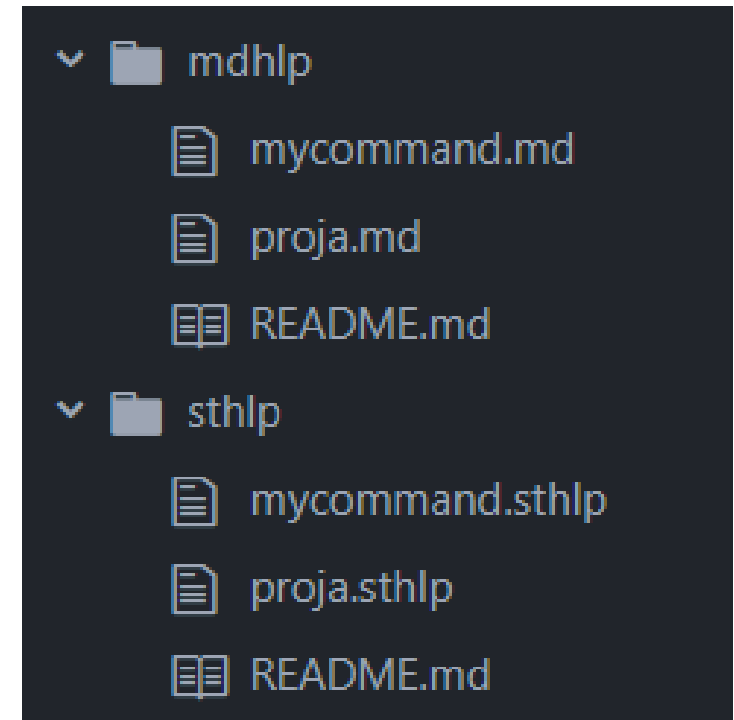
# Publishing documentation

Publishing **mdhlp** files both as helpfiles and as a website

# Publishing mdhlp files both as helpfiles and as a website

\* Add convert all mdhlp files to sthlp files  
local myfolder "C:\Users\wb462869\GitHub\ProjectA"  
ad\_sthlp, adfolder("`myfolder`")

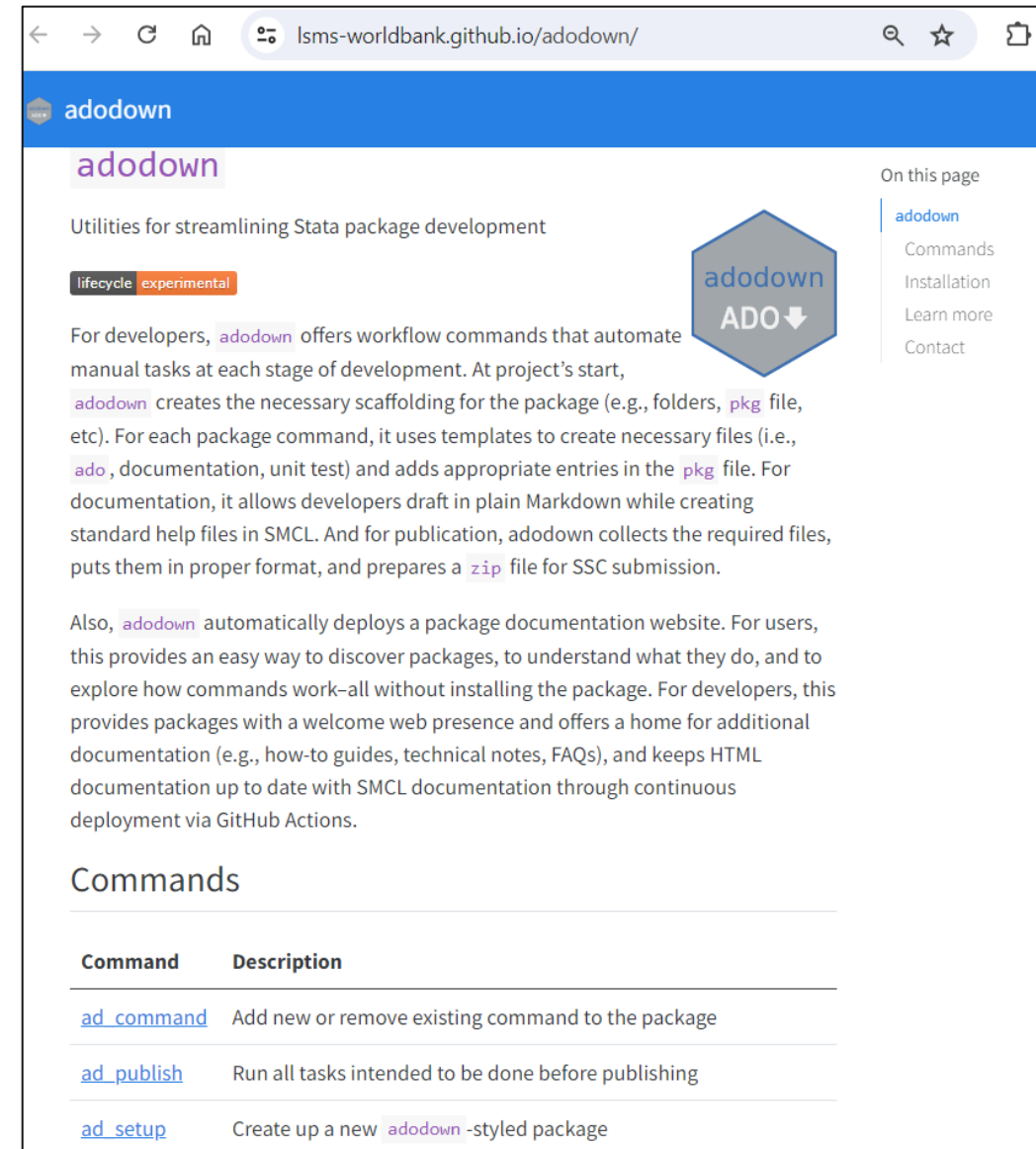
- Convert all mdhlp-files in markdown format to SMCL in sthlp-files



# Web documentation

- Very simple if the package is hosted on GitHub – but can be hosted elsewhere
- The root README.md becomes the landing page, each help file becomes a reference page and longer text vignette articles are supported

<https://lsms-worldbank.github.io/adodown>



The screenshot shows a web browser displaying the GitHub Pages site for the 'adodown' package. The page title is 'adodown' and the subtitle is 'Utilities for streamlining Stata package development'. There are two tabs: 'lifecycle' and 'experimental'. A hexagonal logo with 'adodown ADO↓' is visible. The main text describes the package's workflow, from scaffolding to documentation and deployment. A 'Commands' section lists three commands: 'ad\_command', 'ad\_publish', and 'ad\_setup'.

adodown

Utilities for streamlining Stata package development

lifecycle experimental

adodown ADO↓

On this page

- adodown
- Commands
- Installation
- Learn more
- Contact

For developers, `adodown` offers workflow commands that automate manual tasks at each stage of development. At project's start, `adodown` creates the necessary scaffolding for the package (e.g., folders, `pkg` file, etc). For each package command, it uses templates to create necessary files (i.e., `ado`, documentation, unit test) and adds appropriate entries in the `pkg` file. For documentation, it allows developers draft in plain Markdown while creating standard help files in SMCL. And for publication, `adodown` collects the required files, puts them in proper format, and prepares a `zip` file for SSC submission.

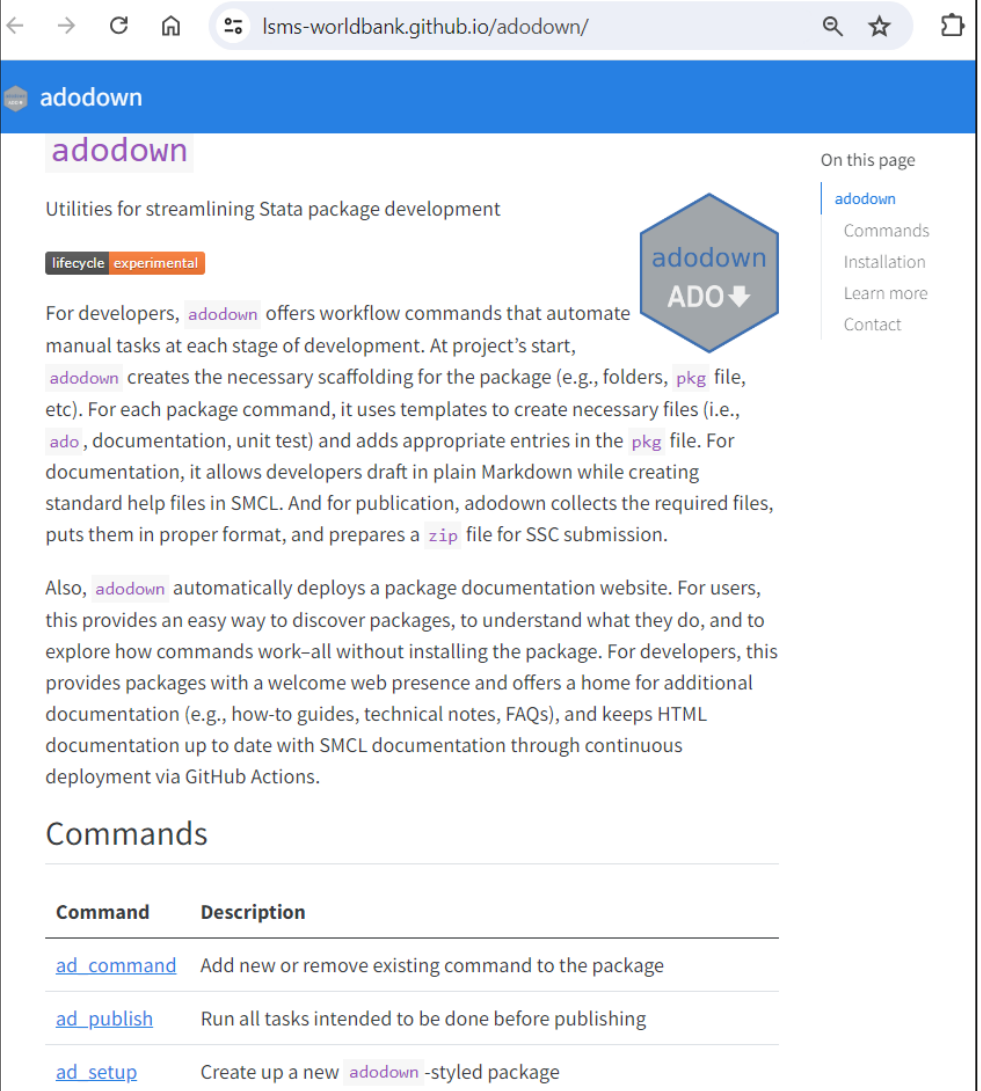
Also, `adodown` automatically deploys a package documentation website. For users, this provides an easy way to discover packages, to understand what they do, and to explore how commands work—all without installing the package. For developers, this provides packages with a welcome web presence and offers a home for additional documentation (e.g., how-to guides, technical notes, FAQs), and keeps HTML documentation up to date with SMCL documentation through continuous deployment via GitHub Actions.

### Commands

Command	Description
<a href="#">ad_command</a>	Add new or remove existing command to the package
<a href="#">ad_publish</a>	Run all tasks intended to be done before publishing
<a href="#">ad_setup</a>	Create up a new <code>adodown</code> -styled package

# Let GitHub do the work for you

- If opting in for GitHub-templates when using **ad\_setup**, then web docs are updated automatically when edits are pushed to the main branch
- The only thing required is to enable GitHub pages on your repository ([instructions](#))
- Possible to compile, render and host website outside GitHub (see [adodownr](#))

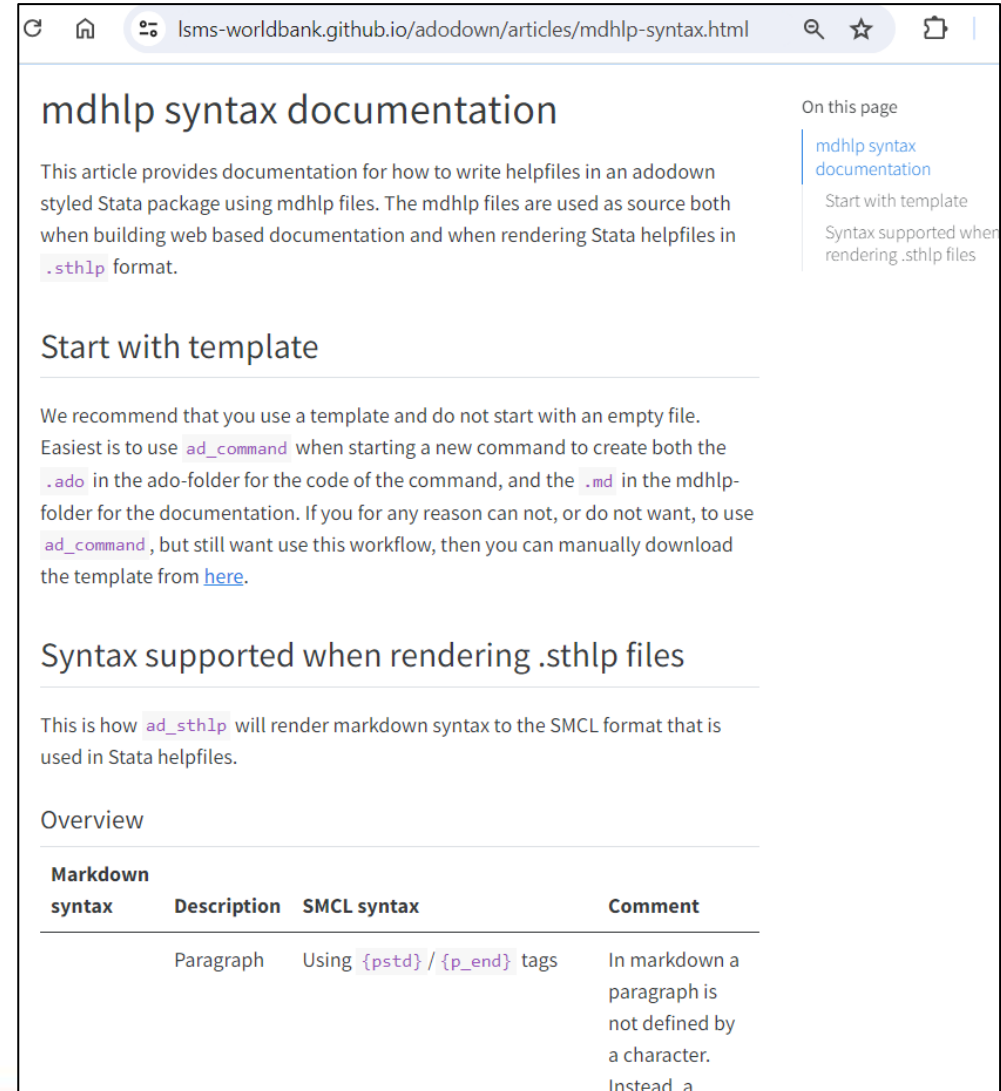


The screenshot shows the GitHub repository page for `adodown` at `lsms-worldbank.github.io/adodown/`. The page features a blue header with the repository name and a navigation menu on the right. The main content area includes a description of the project as "Utilities for streamlining Stata package development" and a "lifecycle experimental" badge. A large hexagonal logo with "adodown ADO" and a downward arrow is positioned on the right. The text describes how `adodown` automates workflow tasks, creates scaffolding, and handles documentation and publication. A "Commands" section at the bottom lists three commands: `ad_command`, `ad_publish`, and `ad_setup`, each with a brief description.

Command	Description
<a href="#">ad_command</a>	Add new or remove existing command to the package
<a href="#">ad_publish</a>	Run all tasks intended to be done before publishing
<a href="#">ad_setup</a>	Create up a new <code>adodown</code> -styled package

# Vignettes (optional but very useful)

- Make help-files quick and easy to use by keeping it to technical reference
- Use vignettes for longer reads for best practices, getting started guides, intended workflows, recommended practices etc.
- Any markdown file added to the vignettes folder are rendered into a vignette



mdhlp syntax documentation

This article provides documentation for how to write helpfiles in an adodown styled Stata package using mdhlp files. The mdhlp files are used as source both when building web based documentation and when rendering Stata helpfiles in `.sthlp` format.

### Start with template

We recommend that you use a template and do not start with an empty file. Easiest is to use `ad_command` when starting a new command to create both the `.ado` in the ado-folder for the code of the command, and the `.md` in the mdhlp-folder for the documentation. If you for any reason can not, or do not want, to use `ad_command`, but still want use this workflow, then you can manually download the template from [here](#).

### Syntax supported when rendering .sthlp files

This is how `ad_sthlp` will render markdown syntax to the SMCL format that is used in Stata helpfiles.

#### Overview

Markdown syntax	Description	SMCL syntax	Comment
	Paragraph	Using <code>{pstd}</code> / <code>{p_end}</code> tags	In markdown a paragraph is not defined by a character. Instead, a

# Publishing package

Utils for publication

# Manage package metadata

- **adodown** uses the .pkg file as the *source of truth* for package metadata
- Use **ad\_update** to update the metadata – possible to update the file directly, but keep the custom headers
- Most of this metadata is generated during package setup – package version most frequent update

```
*** version
v 2.1
*** title
d 'REPKIT': a module facilitating collaboration and computational reproducibility
*** description
d repkit is a package that aims to standardize best practices for
d reproducibility and collaboration as well as making them more accessible to
d the wider Stata community.
d
*** stata
d Requires: Stata version 14.1
d
*** author
d Author: DIME Analytics & LSMS Team, The World Bank
*** contact
d Contact: dimeanalytics@worldbank.org, lsms@worldbank.org
*** url
d URL: https://github.com/worldbank/repkit
d
*** date
d Distribution-Date: 20240516
d
*** adofiles
f ado/reproot.ado
f ado/reprun.ado
f ado/repado.ado
f ado/repkit.ado

*** helpfiles
f sthlp/reproot.sthlp
f sthlp/reprun.sthlp
f sthlp/repado.sthlp
f sthlp/repkit.sthlp
```



# Apply package metadata

- Use **ad\_publish** to apply meta information to package files that will be published
- Updates meta information for all .ado and .sthlp files listed in the pkg file
- Option to create zip-folder in the format **SSC** expects

```
* Add convert all mdhlp files to sthlp files
local myfolder "C:\Users\wb462869\GitHub\ProjectA"
ad_publish, adfolder("`myfolder'") ssczip
```

```
!* version XX XXXXXXXXX ADAUTHORNAME ADCONTACTINFO
cap program drop mycommand
program define mycommand

qui {
    version /* ADD VERSION NUMBER HERE */
    syntax [anything]
}
end
```

```
!* version 0.1 20240725 - Kristoffer Bjarkefur - kbjarkefur@worldbank.org
cap program drop mycommand
program define mycommand

qui {
    version 12.1
    syntax [anything]
}
end
```

# Thank you!

Kristoffer Bjärkefur – [kjarkefur@worldbank.org](mailto:kjarkefur@worldbank.org)

James Shaw – [jshaw@worldbank.org](mailto:jshaw@worldbank.org)

# Stay connected with LSMS

 [worldbank.org/lsms](https://worldbank.org/lsms)

 World Bank's Living Standards Measurement Study

 [lsms@worldbank.org](mailto:lsms@worldbank.org)

GET UPDATES

