



COMMAND LINE REINVENTED

Control of a Business Application via Guided Commands



PDM Use Cases

- Creating and managing data hierarchies (projects, folders, documents)
- Dealing with identities (organizations and persons) and their role-based relations
- Editing configuration data (how to view data etc.)
- All simple operations (selection etc.)



2009 U

Label	Last edited	Date	Status	Size
P6260076	PRichter	06 Jan 2020	Released	154.2 KB
P6260077	PRichter	06 Jan 2020	Released	533.8 KB
P6260078	PRichter	06 Jan 2020	Released	406.6 KB
P6260079	PRichter	06 Jan 2020	Released	378 KB
P6260080	PRichter	06 Jan 2020	Released	530.1 KB
P6260081	PRichter	06 Jan 2020	Released	650.5 KB
P6260082	PRichter	06 Jan 2020	Released	353.2 KB
P6260083	PRichter	06 Jan 2020	Released	327.3 KB
P6260084	PRichter	06 Jan 2020	Released	562.9 KB

2012 U

Label	Last edited	Date	Status	Size
Fotodokumentation Strecke 1503 Hude 15,131	PRichter	06 Jan 2020	Released	1.4 MB

2018 U

Label	Last edited	Date	Status	Size
DSCI2257	PRichter	06 Jan 2020	Released	269.2 KB
DSCI2258	PRichter	06 Jan 2020	Released	822.4 KB
DSCI2259	PRichter	06 Jan 2020	Released	727.5 KB
DSCI2260	PRichter	06 Jan 2020	Released	752.0 KB
DSCI2261	PRichter	06 Jan 2020	Released	245.4 KB
DSCI2262	PRichter	06 Jan 2020	Released	715.5 KB

2021 ff Erneuerung

Document save as release
 Document = Current single selection
 Contents from file = 'P6260077_zoom.jpg'

History note
Reference history

Commands are shown in a dedicated editor pane.

Each view has an own command editor.

View and editor are connected by a Command Context.

Commands are written with help of completion to be well-readable (8 keystrokes here).

Commands consist of a call type and call variables to be filled.



Project open

Project = '2019-130'

Project open

Project = 'pr63r'

Project open

Project = 8751

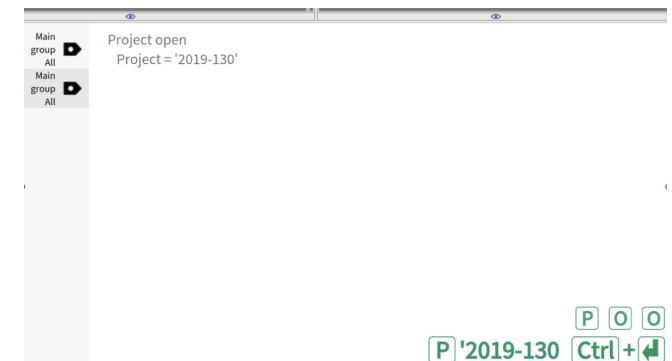
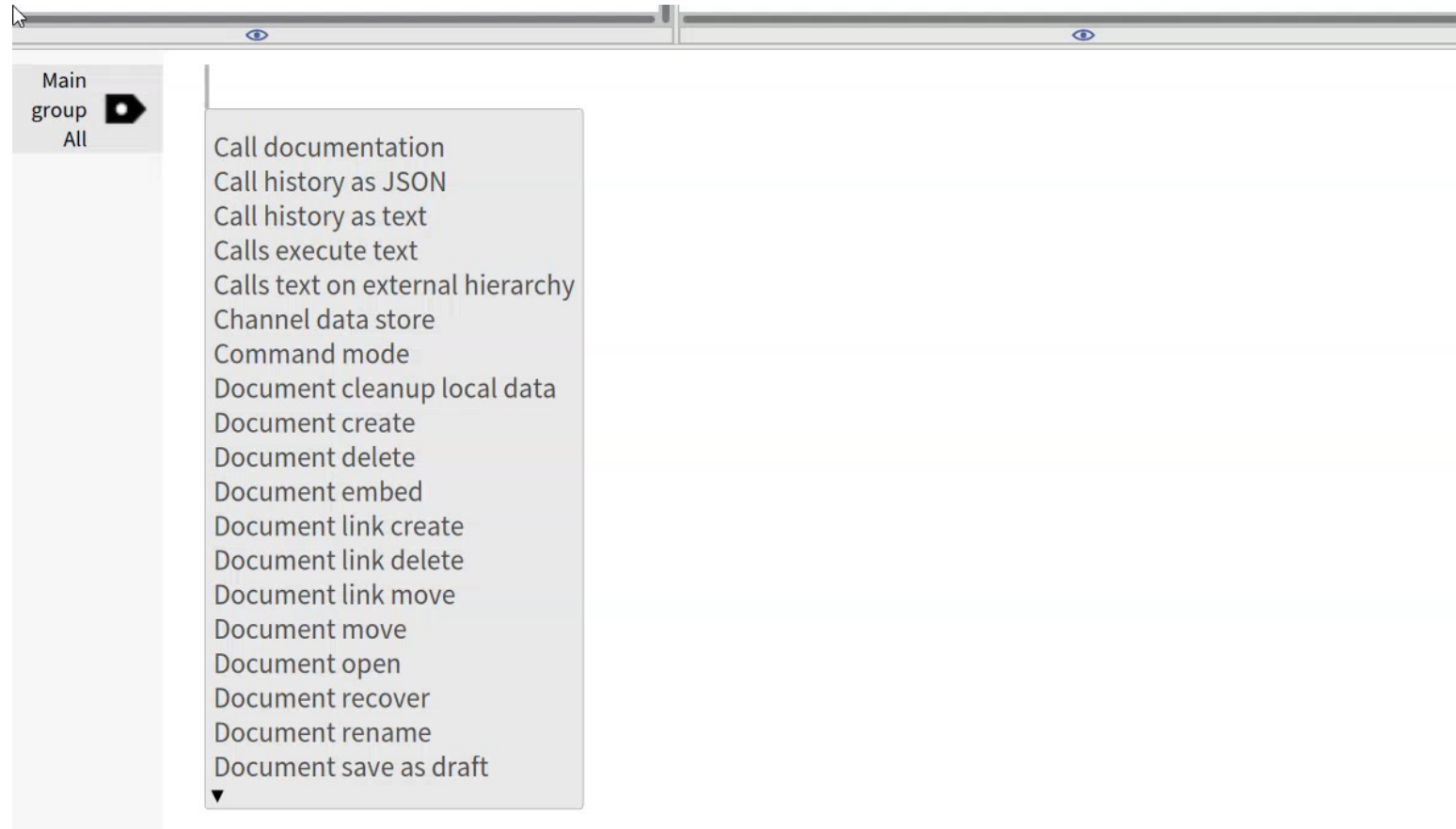
Command variable values are strongly typed, supporting automatic conversion.

Type Project can be converted from

- Project identification
- Item UID
- Primary key (numeric)



Example call: Project open



P '2019-130' [P] [O] [O]
Ctrl + [↓]



Example calls: Folder create, Document create

Main group All

- Call documentation
- Call history as JSON
- Call history as text
- Calls execute text
- Calls text on external hierarchy
- Channel data store
- Command mode
- Document cleanup local data
- Document create
- Document delete
- Document embed
- Document link create
- Document link delete
- Document link move
- Document move
- Document open
- Document recover
- Document rename
- Document save as draft

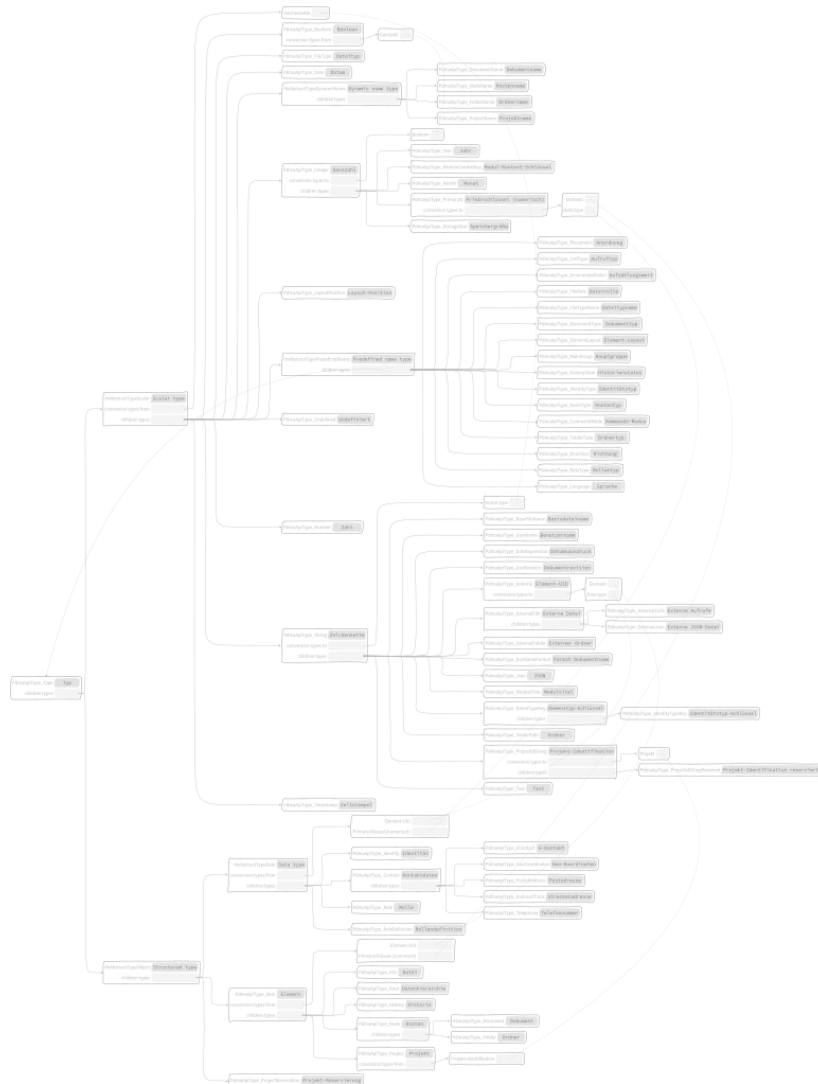
Main group All

Folder create
Type = 'Separator'
Name = 'Test'

Document create
Name = 'My doc'
File type = 'Microsoft Word document'
Placement = 'First below reference'
Reference item = Last call.Folder create.Result
Contents from file = Clipboard
Save immediately as = 'Released'

Main group All

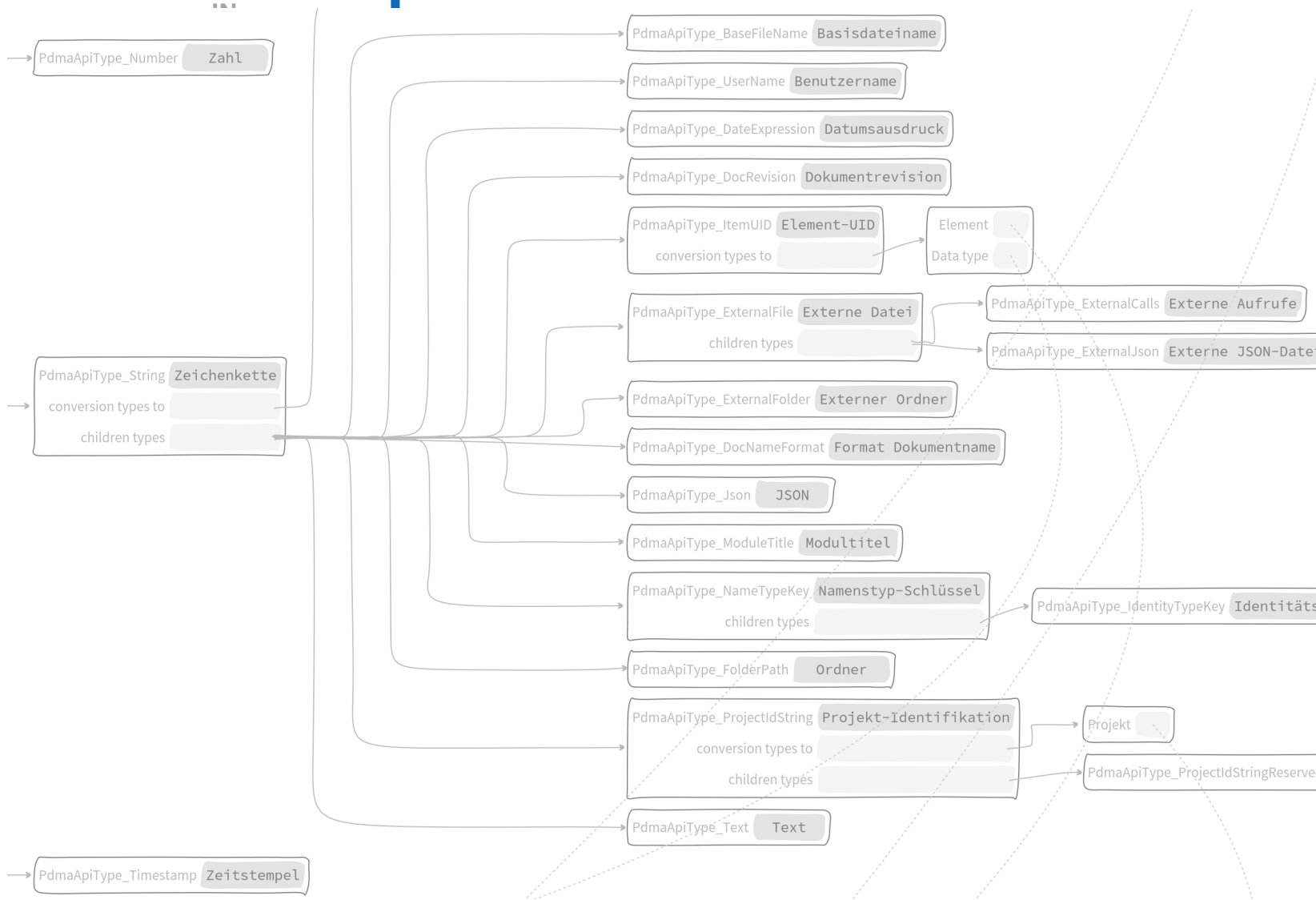
- Call documentation
- Call history as JSON
- Call history as text
- Calls execute text
- Calls text on external hierarchy
- Channel data store



Type Hierarchy as SVG

or optionally as text, always generated by command.

Future extension: Generate for subset of types.



Cutout of Types SVG

Showing type names in German, internal keys as labels.

Type keys are needed for the remote call API.

Diagram (as well as text documentation) shows references in both directions.



Label	Last edited	Date	Status	Size
Zeithonorar (2013):: Einleittrag nach Aufwand (AKönig)	AKönig	08 Sep 2020	Released	321.7 KB
Projektbuchhaltung2021_01	SStingl	19 Feb 2021	Released	7.5 MB

Label	Last edited	Date	Status	Size
Antrag nach §§8-10 WHG				
Antrag nach §§8-10 WHG Antragsformular	MMenne	26 Apr 2019	Released	134.9 KB
Antrag nach §§8-10 WHG_Erläuterungsbericht	MMenne	26 Apr 2019	Released	694.5 KB
DWA-A 153	MMenne	29 Apr 2019	Released	2.4 MB
Antrag nach §§8-10 WHG_Deckblatt	MMenne	29 Apr 2019	Revision created	126.2 KB
Mappenbeschriftung Antrag	MMenne	30 Apr 2019	Revision created	49.9 KB
2019-05-02-Versand Antrag	MMenne	02 May 2019	Revision created	610.5 KB

Running commands via the remote API, calls executed in testing mode

Label	Last edited	Date	Status	Size
Test doc1	Develop...	heute 14:44	Released	36 B
Test doc2	Develop...	heute 14:44	Released	72 B
Test doc3	Develop...	heute 14:44	Released	108 B
Test doc4	Develop...	heute 14:44	Released	144 B
Test doc5	Develop...	heute 14:44	Released	180 B
Test doc6	Develop...	heute 14:44	Released	216 B
Test doc7	Develop...	heute 14:44	Released	252 B
Test doc8	Develop...	heute 14:44	Released	288 B
Test doc9	Develop...	heute 14:44	Released	324 B
Test doc10	Develop...	heute 14:44	Released	360 B



Main advantage: Well-defined API

- Auto-completed input
- Pasted input
- Replayed input
- Input created by speech recognition
- Input as JSON via remote connection
- Well-defined unit tests for basics (parsing, assignments, conversion)
- High-level test runs based on computed calls
- All involved objects (calls and parameters) have a defined type



Next Steps

- Refining editor layout and make it configurable
- Connect input from speech recognition
- Dealing with different command modes:
 - Definition and execution of custom call templates
 - Define custom menus
 - Render direct actions and forms based on focus items like:

Correspondence		Last edited	Date	Status	Size
	Request 1234	Developer	today 12:27	Private draft	0 B
    					



Where We Want to Get to

Currently, we have more than 50 calls implemented, expecting several hundred.

Users of our document management system have different tasks to complete. So there is not one GUI approach well suitable for all purposes.

If the users finally can just process their individual workflows without thinking of „application features“, then we will have reached an important point.

Our technical base already exists in production quality. We want to reach the state of „emerging functionality“ mainly by

- making custom templates easily configurable for access via standard GUI (menues, buttons and hot keys)
- providing pre-configured sample scenarios for power users, where they can try out calls in dry runs.