

GridCraft ALPHA

clone of Microsoft Excel

Spreadsheets, formulas, PivotTables and charts. You describe the analysis; Claude builds the workbook through the gridcraft MCP server, tests formulas before writing them, and reads the results back.

v0.7.0

Driven from Claude Code via MCP
Richard Young • richard@deepneuro.ai
2026-10-15

Alpha software. GridCraft is a pre-1.0 ArtCraft app. Expect missing features, rough edges and the occasional crash; keep originals safe and let Claude work on copies.

21

MCP tools

341

engine commands

514

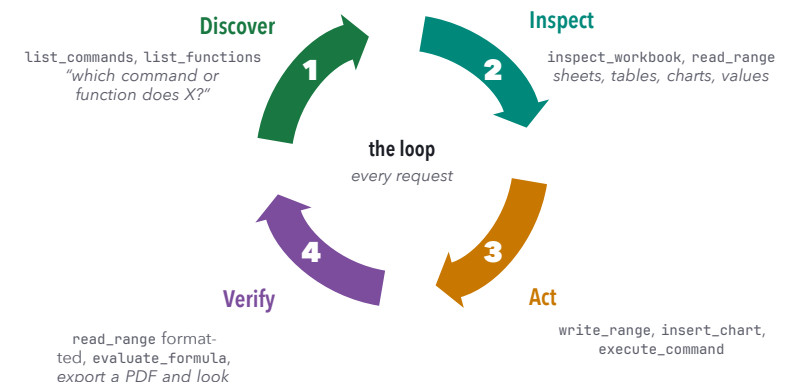
worksheet functions

3

sample workbooks

01 The idea in one picture

You never click a ribbon. You type a goal in Claude Code; Claude turns it into MCP tool calls; the server runs engine commands against a workbook held in memory; and `read_range` or an exported PDF shows the result before anything is saved.



i Check it is wired up

In Claude Code type `/mcp. gridcraft` should be *connected* with 21 tools. The workbook lives inside the server process: nothing reaches disk until Claude calls `save_workbook`, so always finish with “save it to ...”.

! There is no screenshot in headless mode

`screenshot` only works against a desktop window (`gridcraft-cli mcp --connect 7979`), which is not registered. To see a sheet, ask Claude to `read_range` with `formatted` true, or to export a PDF (`file.exportPdf`) and look at it (storyboard 1 does exactly that).

02 First five minutes

The image on this page is real output from GridCraft, captured through the MCP server while writing this guide. Paste the prompts as they are.

- 1 Open a sample** “Use gridcraft: open the sales sample and inspect the workbook.” You get sheet names, used ranges, tables and charts back as JSON. That proves the connection works.
- 2 Read the numbers** “Read B4:I11 formatted.” Values come back exactly as the sheet displays them (\$362,100, 55.8%).
- 3 Look at it** “Export the sheet to ~/reports/sales.pdf.” Claude can open the PDF and show you the page (storyboard 1).
- 4 Test a formula, then write it** `evaluate_formula` never changes the sheet; only `write_range` does (worked example below).
- 5 Save** “Save as ~/reports/sales-copy.xlsx.” The format follows the extension.

Storyboard 1: open a sample, read it, see it

YOU, IN CLAUDE CODE

"Use gridcraft. Open the sales sample, tell me what is in it, read the table formatted, then export the sheet as a PDF so I can see it."

- MCP CALLS CLAUDE MADE
- 1 new_workbook {"sample": "sales"} → "Quarterly Sales Dashboard"
 - 2 inspect_workbook sheet Sales, used range B2:I17, table Sales B4:I11 (TableStyleMedium2, totals row), charts "Revenue by Region" (column, K3) and "Share of Annual Total" (pie, K19), 2 conditional formats, 1 comment
 - 3 read_range {"range": "B4:I11", "formatted": true} → Region, Q1...Q4, Total, Growth; Online \$362,100 and 55.8%; grand total \$1,381,540
 - 4 execute_command file.exportPdf{"path": "~/craft-guides-demo/sales.pdf"} → 2 pages, 23 882 bytes

WHAT CAME BACK

Quarterly Sales Dashboard

Fiscal year 2026 - all figures in USD

Region	Q1	Q2	Q3	Q4	Total	Growth
North	\$48,200	\$51,340	\$56,010	\$61,870	\$217,420	28.4%
South	\$39,150	\$37,980	\$42,400	\$45,120	\$164,650	15.2%
East	\$55,300	\$58,910	\$57,230	\$63,480	\$274,920	14.8%
West	\$61,020	\$64,400	\$70,150	\$76,390	\$271,960	25.2%
Central	\$28,740	\$31,260	\$33,980	\$36,510	\$130,490	27.0%
Online	\$72,410	\$81,630	\$95,220	\$112,840	\$362,100	55.8%
Total	\$304,820	\$325,520	\$354,990	\$396,210	\$1,381,540	27.7%

Best region	Online
Average quarter	\$57,564
Online share	26.2%
Year-over-year	30.0%

Page 1 of the exported PDF, rendered with PdfCraft's page_render; the two charts are on page 2.

- 1 Table header row, TableStyleMedium2, from inspect_workbook.
- 2 Total column with its colour-scale conditional format.
- 3 Summary block: best region, average quarter, year-over-year.

Worked example: test a formula before writing it

YOU, IN CLAUDE CODE

"In the sales sample, add a column J "vs. avg" showing each region's Total as a percentage of the average Total. Test the formula first, then write it and read it back."

- MCP CALLS CLAUDE MADE
- 1 evaluate_formula {"formula": "=G5/AVERAGE(G5:G10)", "cell": "J5"} → 0.9443 (nothing changes)
 - 2 write_range {"range": "J4", "values": [{"vs. avg"}, {"=G5/AVERAGE(G\$5:G\$10)"}, ...]} (one undo step)
 - 3 format_range {"range": "J5:J10", "numberFormat": "0.0%"} (one undo step)
 - 4 read_range {"range": "J4:J10", "formatted": true}

● CLAUDE

Column J is in: North 94.4%, South 71.5%, East 102.0%, West 118.1%, Central 56.7%, Online 157.3%. Online is the only region above 150% of the average. Say “undo” if you want it gone; it was a single step.

Prompts to explore with

●●● warm-up ●●● real work ●●● stretch

●●●

“Use gridcraft: open the grades sample and inspect the workbook.”

`new_workbook` with `sample`;
`inspect_workbook` shows sheets, tables and charts.

●●●

“Which GridCraft functions handle business-day math? Show their signatures.”

`list_functions` with `search`; 514 functions to browse.

●●●

“Open ~/data/enrollments.csv as a table on a new sheet, sort by term, and tell me its shape.”

`data.fromTextCsv`, `create_table`,
`sort_range`.

●●●

“Pivot the sales sample by Region with quarters as columns, then add a column PivotChart.”

`insert.pivotTable` and `insert.pivotChart` through `execute_command`.

●●●

“List the commands that contain “conditional” and apply a 3-colour scale to the Growth column.”

`list_commands` first, then `home.conditionalFormat`.

●●●

“Open ~/grants/budget.xlsx (work on a copy), find every formula error, trace what feeds the grand total, and write a summary sheet.”

`formulas.errorChecking`, `formulas.tracePrecedents`, `read_range` with `formulas`.

●●●

“Forecast monthly enrollments 12 periods ahead with a 95% interval, chart it, export the chart sheet as a PDF and show me.”

`data.forecastSheet`, `file.exportPdf`, then look at the page.

03 Goals → how

Each card is a prompt you can paste, the tools Claude reaches for, and the command ids doing the work.

Data in, cleanup & formulas

1 Start a workbook and enter data

ASK CLAUDE

"New workbook: put this quarterly table in A1, add SUM totals, save as ~/reports/q3.xlsx."

TOOLS `new_workbook` → `write_range`
→ `save_workbook`

KEY `range` (top-left cell) + `values` (rows; = strings are formulas; "12%", "\$5", dates parse as typed). One undo step.

- `sample`: budget, sales, grades for practice data.

2 Import a CSV as a proper table

ASK CLAUDE

"Import ~/data/enrollments.csv into a new sheet Raw as a table and tell me its shape."

TOOLS `execute_command` → `inspect_workbook`

KEY `data.fromTextCsv` : `path`, `delimiter`, `sheet`.
`data.getData` also takes JSON arrays of objects.

- XLSB/ODS import cached values only (no formulas).

3 Clean a messy export

ASK CLAUDE

"Split Name on commas, drop duplicate rows by Email, blank out 'N/A', sort by Last."

TOOLS `execute_command` → `sort_range`

KEY `data.textToColumns` `data.removeDuplicates`
(`columns`) `edit.replace` (`what`, `with`, `all`)
`edit.flashFill`

4 Write formulas safely

ASK CLAUDE

"In F compute a 30/70 weighted grade and an IFS letter grade; test before writing."

TOOLS `evaluate_formula` → `write_range`
→ `read_range`

KEY `evaluate_formula` never changes the sheet; `cell` anchors relative refs. `range.fill` fills one input down a range.

- LET, LAMBDA, XLOOKUP and dynamic arrays are supported.

5 Audit an inherited workbook

ASK CLAUDE

"Open ~/grants/budget.xlsx, list formula errors, show what feeds H42, dump Summary's formulas."

TOOLS `open_workbook` → `execute_command`
→ `read_range`

KEY `formulas.errorChecking`
`formulas.tracePrecedents`
`formulas.nameManager` ; `read_range formulas`:
`true`.

6 Find the right function

ASK CLAUDE

"Which functions handle business-day math? Show signatures."

TOOLS `list_functions`

KEY `search` (name or description) or `category`. 514 functions on the CLI list.

Analysis

7 Build a PivotTable + PivotChart

ASK CLAUDE

"Pivot Sales: Region rows, months as columns, sum of Revenue as % of row total; add a column PivotChart."

TOOLS `execute_command`

KEY `insert.pivotTable` (`source`, `rows`, `columns`, `values`) `pivot.valueSettings` (`showAs`)
`insert.pivotChart`

- Group dates with {field, group: months} or pivot.group.

8 Sort and filter

ASK CLAUDE

"Sort by Department, then Salary descending; show only the top 10 in Engineering."

TOOLS `sort_range` → `filter`

KEY `keys`: [{"column": "C", "order": "desc"}]; `filter` `values`, `custom`, `top`, `clear`.

- Complex criteria: `data.advancedFilter`.

9 Subtotals and outlines

ASK CLAUDE

"Subtotal Units and Revenue at each change in Region, then collapse to totals."

TOOLS `sort_range` → `execute_command`

KEY `data.subtotal` (`groupBy`, `function`, `columns`)
`data.showLevels`

10 Regression & descriptive stats

ASK CLAUDE

"Regress Final on Midterm and Attendance with residuals; descriptives on a new sheet."

TOOLS `execute_command`

KEY `data.analysisToolPak` `tool`: regression (`inputY`, `inputX`), `descriptive`, `correlation`, `t-tests`, `ANOVA`, ...

- Writes static values; re-run when data changes.

11 Forecast a time series

ASK CLAUDE

"Forecast monthly enrollments 12 periods ahead with a 95% interval and a chart."

TOOLS `execute_command`

KEY `data.forecastSheet` `timeline`, `values`, `periods`, `confidence`, `seasonality`, `chart`

- New sheet with FORECAST.ETS columns and bounds.

12 What-if: Goal Seek & Solver

ASK CLAUDE

"What price in B2 makes profit B10 = \$50k? Then maximise B10 over B2:B4 with B5 <= 10."

TOOLS `execute_command`

KEY `data.goalSeek` (`set`, `to`, `changing`)
`data.solver` (`objective`, `variables`, `constraints`, `method`)

- Also `data.scenarioAdd`, `data.scenarioSummary`, `data.dataTable`.

Presentation

13 Chart the data

ASK CLAUDE

"Clustered column chart of A1:C13 titled 'Revenue vs Cost' at H2, with a linear trendline and R-squared."

TOOLS `insert_chart` → `execute_command`

KEY `type` (17 kinds), `subtype`, `title`, `at`; then `chart.addElement` (`trendline`, `displayRSquared`)

- `insert.chartRecommendations` ranks types for your data.

14 Make a report look finished

ASK CLAUDE

"Style A1:F40 as a table, currency in C:E, color scale on Margin, data bars on Units, freeze row 1."

TOOLS `create_table` → `format_range`
→ `execute_command`

KEY `home.conditionalFormat` `rule type`: `colorScale`, `dataBar`, `iconSet`, ...
`view.freezeTopRow`

15 Input form with dropdowns

ASK CLAUDE

"Dropdown of Pending/Approved/Rejected in B2:B50; protect the sheet but keep B editable."

TOOLS `execute_command`

KEY `data.validation` (`type list`, `formula1`)
`home.lockCell` `review.protectSheet`

16 Sparklines and in-cell trends**ASK CLAUDE**

"Add a line sparkline in N2 for each row's monthly values B2:M2."

TOOLS `execute_command`

KEY `insert.sparkline` `range`, `location`, `type`
(line/column/winLoss), `markers`

Output & automation**17 Print-ready PDF or CSV****ASK CLAUDE**

"Landscape, fit to one page wide, repeat row 1, export Summary to ~/reports/summary.pdf."

TOOLS `execute_command` → `save_workbook`

KEY `pageLayout.pageSetup` (`orientation`, `fitWidth`, `titleRows`) `file.exportPdf` (`path`, `sheets`)

● `save_workbook` to .csv saves the active sheet.

18 Save a repeatable recipe**ASK CLAUDE**

"Save those cleanup steps as a script WeeklyClean so I can rerun it next week."

TOOLS `execute_command`

KEY `automate.newScript` (`name`, `commands`)
`automate.runScript`

● Shell: `gridcraft-cli run --script steps.jsonl`.

✓ Always add: "read it back"

`read_range` with `formatted` true is cheap and shows exactly what a user would see. For layout (charts, conditional formats, column widths) ask for a PDF export and let Claude look at the page, as in storyboard 1.

04 The command map (341 ids behind `execute_command`)

Every ribbon button is a dotted command id. Darker tiles are bigger families. Ask Claude to "list gridcraft commands matching word" to open any tile.

64

home.

fonts, fills, borders, number
formats

37

data.

sort, filter, validation, solver,
forecast

27

formulas.

names, trace, error checking

26

edit.

find, replace, fill, flash fill

24

view.

freeze, zoom, gridlines, split

21

review.

comments, protection,
spelling

20

pivot.

fields, grouping, layout, re-fresh

16

insert.

pivot, sparkline, slicer, picture

16

pageLayout.

page setup, themes, print area

14

table.

style, totals row, resize

13

file.

exportPdf, exportCsv, proper-ties

12

chart.

set, addElement, changeType

10

sheet.

insert, delete, rename, move

7

selection.

select ranges, rows, columns

5

automate.

newScript, runScript, record

19

other

arrange, draw, slicer, cell, ob-ject

05 Tool reference (all 21 MCP tools)

MCP tool	What it does
Generic command layer	
list_commands	List commands: id, label, ribbon path, shortcut, params doc, runnable now. Filter with search.
execute_command	Run any command by id: command + params object.
Files and workbook state	
new_workbook	Blank workbook, or sample: budget, sales, grades.
open_workbook	Open path (xlsx, xlsxm, xls, xlsb, ods, csv, tsv, txt, json); check warnings.
save_workbook	Save to current file or path; format follows extension (.xlsx default).
inspect_workbook	Sheets, used ranges, tables, charts, merges, filters, freeze panes, names, selection, calc mode, undo history.
undo / redo	Step history; steps (default 1).
Cells and formulas	
read_range	2-D values for range (default used range); formulas, formatted, sheet. Errors come back as {"error": "#DIV/0!"}.
write_range	Write rows of values from the top-left of range. One undo step.
set_cell	Enter one cell exactly as typed (input: number, text or formula).
get_cell	Value, type, formula, display text, spill range, style, merge, comment, hyperlink.
evaluate_formula	Evaluate formula without changing anything; optional cell anchor.
list_functions	Worksheet functions with category, signature, description; search, category.
select	Set the selection (range, multi-area with commas; active).
Shaping and presenting	
format_range	Bold/italic/underline, font, colors, numberFormat, alignment, wrap, borders; one command per property.
create_table	Format range as a table: name, style, header.
insert_chart	Chart from range: type, subtype, title, at; returns chart id.
sort_range	Multi-key sort: keys, range, header.
filter	AutoFilter a column by values, custom, top, blanks, or clear.

MCP tool

What it does (cont.)

screenshot

PNG of the window to [path](#). Desktop-bridge mode only; fails headless.

06 Example tool calls

write_range: a data block with formulas

```
{
  "range": "A1",
  "values": [
    ["Quarter", "Revenue", "Cost", "Margin"],
    ["Q1", 120000, 80000, "=B2-C2"],
    ["Q2", 135000, 86000, "=B3-C3"],
    ["Q3", 150000, 91000, "=B4-C4"],
    ["Total", "=SUM(B2:B4)", "=SUM(C2:C4)", "=SUM(D2:D4)"]
  ]
}
```

execute_command: PivotTable by region and month

```
{
  "command": "insert.pivotTable",
  "params": {
    "source": "Sales",
    "destination": "new",
    "rows": ["Region"],
    "columns": [{
      "field": "Date",
      "group": "months"
    }],
    "values": [{
      "field": "Revenue",
      "func": "sum",
      "name": "Total Revenue"
    }]
  }
}
```

Storyboard 1: export the sheet as a PDF to look at it

```
{
  "command": "file.exportPdf",
  "params": {
    "path": "~/craft-guides-demo/sales.pdf"
  }
}
-> {
  "bytes": 23882,
  "pages": 2,
  "path": "~/craft-guides-demo/sales.pdf"
}
```

execute_command: color scale on a column

```
{
  "command": "home.conditionalFormat",
  "params": {
    "range": "E2:E40",
    "rule": {
      "type": "colorScale",
      "colors": ["#F8696B", "#FFEB84", "#63BE7B"]
    }
  }
}
```

- Exact shape of `colors` for a 3-color scale isn't documented; have Claude check `list_commands` first.

07 Limits & tips

! Alpha status

- About 89% of Excel's ribbon commands and 513 of 523 functions, but the team rates it **about 50% ready for real work**.
- Known gaps: XLSX fidelity on real-world files, very large workbooks, chart depth.
- **Work on copies of XLSX originals**; save under a new name and check with `gridcraft-cli check`.

✓ Habits that work

- **Save explicitly.** The headless workbook lives in the server process; ask Claude to `save_workbook` to a path when done.
- **Inspect first.** "Inspect the workbook and read the header row" lets Claude learn sheet names, tables and used ranges before editing.
- **Test, then write.** `evaluate_formula` is free; `undo` reverts a bad step. Formula text and APIs use English syntax (commas), whatever the UI language.
- **Static outputs.** Analysis ToolPak, data tables and Solver reports write values, not formulas; re-run after data changes. PivotTables refresh with `pivot.refresh / data.refreshAll`.
- **No dialogs.** Commands that open a dialog in the UI need all their params over MCP; a missing required param is an error, never a popup. Ask Claude to call `list_commands` when a call errors.
- **Shell shortcuts.** Claude can also use the CLI directly: `gridcraft-cli cat FILE --range A1:D20,eval,convert in.xlsx out.csv,info FILE --json`.

i Paths and scope

The registered `gridcraft` server has no `--root` flag, so there is no file-access restriction to work around; still, give absolute paths inside your home folder. `screenshot` needs the desktop bridge (`--connect 7979`), which is not registered, so it will fail in the default setup. Export a PDF instead.