

## docs/N706B/source/2/AT+CMGD.rst

.. \_cmd-at+cmgd:

AT+CMGD : 删除短消息

从当前存储器中删除短消息。

命令格式

命令格式

~~~~~

\*\*执行命令\*\*

~~~~~

\*\*命令 : \*\*

::

AT+CMGD

\*\*响应 : \*\*

::

<CR><LF>OK<CR><LF>

参数

~~~~

- \*\*<service>\*\* :

短信服务模式选择

- \*\*<mt>,<mo>,<bm>\*\* :

下行/上行/广播支持

说明

~~~~

说明示例41

## 示例命令

~~~~~

:::

AT+CMGD

OK

**docs/N706B/source/command\_index.rst**

## Neoway AT 命令索引

=====

| 序号 | 命令                               | 命令标题         | 所属章节     |
|----|----------------------------------|--------------|----------|
| 1  | :doc:`AT+CCID <1/AT+CCID>`       | 获取SIM卡标识     | 第1章 基础命令 |
| 2  | :doc:`AT+CCID <1/AT+CCID>`       | 获取SIM卡标识     | 第1章 基础命令 |
| 3  | :doc:`AT+CCLK <1/AT+CCLK>`       | 时钟管理         | 第1章 基础命令 |
| 4  | :doc:`AT+CCLK <1/AT+CCLK>`       | 时钟管理         | 第1章 基础命令 |
| 5  | :doc:`AT+CEREG <1/AT+CEREG>`     | 获取EPS网络注册状态  | 第1章 基础命令 |
| 6  | :doc:`AT+CEREG <1/AT+CEREG>`     | 获取EPS网络注册状态  | 第1章 基础命令 |
| 7  | :doc:`AT+CESQ <1/AT+CESQ>`       | 扩展信号强度       | 第1章 基础命令 |
| 8  | :doc:`AT+CESQ <1/AT+CESQ>`       | 扩展信号强度       | 第1章 基础命令 |
| 9  | :doc:`AT+CFUN <1/AT+CFUN>`       | 设置模组功能       | 第1章 基础命令 |
| 10 | :doc:`AT+CFUN <1/AT+CFUN>`       | 设置模组功能       | 第1章 基础命令 |
| 11 | :doc:`AT+CGATT <1/AT+CGATT>`     | 设置GPRS附着和分离  | 第1章 基础命令 |
| 12 | :doc:`AT+CGATT <1/AT+CGATT>`     | 设置GPRS附着和分离  | 第1章 基础命令 |
| 13 | :doc:`AT+CGDCONT <1/AT+CGDCONT>` | 设置PDP格式      | 第1章 基础命令 |
| 14 | :doc:`AT+CGDCONT <1/AT+CGDCONT>` | 设置PDP格式      | 第1章 基础命令 |
| 15 | :doc:`AT+CGMM <1/AT+CGMM>`       | 查询模组型号       | 第1章 基础命令 |
| 16 | :doc:`AT+CGMM <1/AT+CGMM>`       | 查询模组型号       | 第1章 基础命令 |
| 17 | :doc:`AT+CGSN <1/AT+CGSN>`       | 获取通信模组IMEI号  | 第1章 基础命令 |
| 18 | :doc:`AT+CGSN <1/AT+CGSN>`       | 获取通信模组IMEI号  | 第1章 基础命令 |
| 19 | :doc:`AT+CIMI <1/AT+CIMI>`       | 查询国际移动用户识别码  | 第1章 基础命令 |
| 20 | :doc:`AT+CIMI <1/AT+CIMI>`       | 查询国际移动用户识别码  | 第1章 基础命令 |
| 21 | :doc:`AT+CLCK <1/AT+CLCK>`       | PIN使能与查询功能指令 | 第1章 基础命令 |
| 22 | :doc:`AT+CLCK <1/AT+CLCK>`       | PIN使能与查询功能指令 | 第1章 基础命令 |
| 23 | :doc:`AT+CMUX <1/AT+CMUX>`       | 串口多路复用指令     | 第1章 基础命令 |
| 24 | :doc:`AT+CMUX <1/AT+CMUX>`       | 串口多路复用指令     | 第1章 基础命令 |
| 25 | :doc:`AT+COPS <1/AT+COPS>`       | 网络选择         | 第1章 基础命令 |
| 26 | :doc:`AT+COPS <1/AT+COPS>`       | 网络选择         | 第1章 基础命令 |
| 27 | :doc:`AT+CPIN <1/AT+CPIN>`       | 输入PIN码       | 第1章 基础命令 |
| 28 | :doc:`AT+CPIN <1/AT+CPIN>`       | 输入PIN码       | 第1章 基础命令 |
| 29 | :doc:`AT+CPWD <1/AT+CPWD>`       | 修改密码指令       | 第1章 基础命令 |
| 30 | :doc:`AT+CPWD <1/AT+CPWD>`       | 修改密码指令       | 第1章 基础命令 |
| 31 | :doc:`AT+CREG <1/AT+CREG>`       | 查询网络注册状态     | 第1章 基础命令 |
| 32 | :doc:`AT+CREG <1/AT+CREG>`       | 查询网络注册状态     | 第1章 基础命令 |
| 33 | :doc:`AT+CSQ <1/AT+CSQ>`         | 获取信号强度       | 第1章 基础命令 |
| 34 | :doc:`AT+CSQ <1/AT+CSQ>`         | 获取信号强度       | 第1章 基础命令 |
| 35 | :doc:`AT+GMM <1/AT+GMM>`         | 查询模组型号       | 第1章 基础命令 |
| 36 | :doc:`AT+GMM <1/AT+GMM>`         | 查询模组型号       | 第1章 基础命令 |

```

| 37 | :doc:`AT+GMR <1/AT+GMR>` | 查询版本信息 | 第1章 基础命令 |
| 38 | :doc:`AT+GMR <1/AT+GMR>` | 查询版本信息 | 第1章 基础命令 |
| 39 | :doc:`AT+GSN <1/AT+GSN>` | 获取通信模组IMEI号 | 第1章 基础命令 |
| 40 | :doc:`AT+GSN <1/AT+GSN>` | 获取通信模组IMEI号 | 第1章 基础命令 |
| 41 | :doc:`AT+IPR <1/AT+IPR>` | 设置模组波特率 | 第1章 基础命令 |
| 42 | :doc:`AT+IPR <1/AT+IPR>` | 设置模组波特率 | 第1章 基础命令 |
| 43 | :doc:`AT+NSTGETRSSI <1/AT+NSTGETRSSI>` | 获取强收模式下模组的接收功率 (仅可用于测试) | 第1
章 基础命令 |
| 44 | :doc:`AT+NSTGETRSSI <1/AT+NSTGETRSSI>` | 获取强收模式下模组的接收功率 (仅可用于测试) | 第1
章 基础命令 |
| 45 | :doc:`AT+NWDNS <1/AT+NWDNS>` | 域名解析 | 第1章 基础命令 |
| 46 | :doc:`AT+NWDNS <1/AT+NWDNS>` | 域名解析 | 第1章 基础命令 |
| 47 | :doc:`AT+NWENPWRSAVE <1/AT+NWENPWRSAVE>` | 休眠 (Sleep) 设置 | 第1章 基础命令 |
| 48 | :doc:`AT+NWENPWRSAVE <1/AT+NWENPWRSAVE>` | 休眠 (Sleep) 设置 | 第1章 基础命令 |
| 49 | :doc:`AT+NWPWROFF <1/AT+NWPWROFF>` | 模组关机指令 | 第1章 基础命令 |
| 50 | :doc:`AT+NWPWROFF <1/AT+NWPWROFF>` | 模组关机指令 | 第1章 基础命令 |
| 51 | :doc:`AT+NWRFTEST <1/AT+NWRFTEST>` | 模组强发强收指令 (仅可用于测试) | 第1章 基础命令 |
| 52 | :doc:`AT+NWRFTEST <1/AT+NWRFTEST>` | 模组强发强收指令 (仅可用于测试) | 第1章 基础命令 |
| 53 | :doc:`AT+XGAUTH <1/AT+XGAUTH>` | 用户认证 | 第1章 基础命令 |
| 54 | :doc:`AT+XGAUTH <1/AT+XGAUTH>` | 用户认证 | 第1章 基础命令 |
| 55 | :doc:`ATD <1/ATD>` | *99# : GPRS | 第1章 基础命令 |
| 56 | :doc:`ATD <1/ATD>` | *99# : GPRS | 第1章 基础命令 |
| 57 | :doc:`ATE1 <1/ATE1>` | /ATE0 : 打开&关闭回显 | 第1章 基础命令 |
| 58 | :doc:`ATE1 <1/ATE1>` | /ATE0 : 打开&关闭回显 | 第1章 基础命令 |
| 59 | :doc:`ATI <1/ATI>` | 获得模组厂商信息 | 第1章 基础命令 |
| 60 | :doc:`ATI <1/ATI>` | 获得模组厂商信息 | 第1章 基础命令 |
| 61 | :doc:`AT+CMGD <2/AT+CMGD>` | 删除短消息 | 第2章 SMS命令 |
| 62 | :doc:`AT+CMGD <2/AT+CMGD>` | 删除短消息 | 第2章 SMS命令 |
| 63 | :doc:`AT+CMGF <2/AT+CMGF>` | 设置短消息模式 | 第2章 SMS命令 |
| 64 | :doc:`AT+CMGF <2/AT+CMGF>` | 设置短消息模式 | 第2章 SMS命令 |
| 65 | :doc:`AT+CMGL <2/AT+CMGL>` | 短信列表 | 第2章 SMS命令 |
| 66 | :doc:`AT+CMGL <2/AT+CMGL>` | 短信列表 | 第2章 SMS命令 |
| 67 | :doc:`AT+CMGR <2/AT+CMGR>` | 读短消息 | 第2章 SMS命令 |
| 68 | :doc:`AT+CMGR <2/AT+CMGR>` | 读短消息 | 第2章 SMS命令 |
| 69 | :doc:`AT+CMGS <2/AT+CMGS>` | 发送短消息 | 第2章 SMS命令 |
| 70 | :doc:`AT+CMGS <2/AT+CMGS>` | 发送短消息 | 第2章 SMS命令 |
| 71 | :doc:`AT+CMGW <2/AT+CMGW>` | 写短消息 | 第2章 SMS命令 |
| 72 | :doc:`AT+CMGW <2/AT+CMGW>` | 写短消息 | 第2章 SMS命令 |
| 73 | :doc:`AT+CMSS <2/AT+CMSS>` | 发送已保存的短消息 | 第2章 SMS命令 |
| 74 | :doc:`AT+CMSS <2/AT+CMSS>` | 发送已保存的短消息 | 第2章 SMS命令 |
| 75 | :doc:`AT+CNMI <2/AT+CNMI>` | 设置短信指示格式 | 第2章 SMS命令 |
| 76 | :doc:`AT+CNMI <2/AT+CNMI>` | 设置短信指示格式 | 第2章 SMS命令 |
| 77 | :doc:`AT+CPMS <2/AT+CPMS>` | 首选短信存储器 | 第2章 SMS命令 |
| 78 | :doc:`AT+CPMS <2/AT+CPMS>` | 首选短信存储器 | 第2章 SMS命令 |
| 79 | :doc:`AT+CSCA <2/AT+CSCA>` | 短信中心号码 | 第2章 SMS命令 |
| 80 | :doc:`AT+CSCA <2/AT+CSCA>` | 短信中心号码 | 第2章 SMS命令 |
| 81 | :doc:`AT+CSCS <2/AT+CSCS>` | 设置TE字符集 | 第2章 SMS命令 |
| 82 | :doc:`AT+CSCS <2/AT+CSCS>` | 设置TE字符集 | 第2章 SMS命令 |
| 83 | :doc:`AT+CSDH <2/AT+CSDH>` | 显示文本模式参数 | 第2章 SMS命令 |
| 84 | :doc:`AT+CSDH <2/AT+CSDH>` | 显示文本模式参数 | 第2章 SMS命令 |
| 85 | :doc:`AT+CSMP <2/AT+CSMP>` | 设置文本模式参数 | 第2章 SMS命令 |
| 86 | :doc:`AT+CSMP <2/AT+CSMP>` | 设置文本模式参数 | 第2章 SMS命令 |

```

| 87 | :doc:`AT+CSMS <2/AT+CSMS>` | 选择短信服务 | 第2章 SMS命令 |  
| 88 | :doc:`AT+CSMS <2/AT+CSMS>` | 选择短信服务 | 第2章 SMS命令 |

## docs/N706B/source/conf.py

```
import os
import sys
import platform
from datetime import datetime
from pathlib import Path

project = 'Neoway AT 命令手册'
author = 'Neoway 文档工程组'
release = 'v1.4'
today = datetime.now().strftime('%Y-%m-%d')
copyright = f'{datetime.now().year}, Neoway Technology'

extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.todo',
    'sphinx.ext.ifconfig',
    'sphinx.ext.viewcode',
]
templates_path = ['_templates']
exclude_patterns = ['_build', 'Thumbs.db', '.DS_Store']
language = 'zh_CN'
html_theme = 'sphinx_rtd_theme'
html_static_path = ['_static']
html_logo = '_static/logo.png'
html_show_sourcelink = False
html_last_updated_fmt = today

sys_name = platform.system().lower()
if "darwin" in sys_name or "mac" in sys_name:
    zh_font = "PingFang SC"
    mono_font = "Menlo"
elif "win" in sys_name:
    zh_font = "Microsoft YaHei"
    mono_font = "Consolas"
else:
    zh_font = "Noto Sans CJK SC"
    mono_font = "DejaVu Sans Mono"

latex_engine = 'xelatex'

latex_elements = {
    'papersize': 'a4paper',
    'pointsize': '11pt',
    'extraclassoptions': 'openany,oneside',
    'fontpkg': rf"""
    \usepackage{{xeCJK}}
    \setCJKmainfont{{{zh_font}}}
    """
}
```

```

\\setmainfont{{Times New Roman}}
\\setsansfont{{Arial}}
\\setmonofont{{{mono_font}}}
"",
'preamble': r"""
\usepackage{graphicx,tikz,eso-pic,xcolor,fancyhdr,titlesec,hyperref}
\graphicspath{{./}{_common/_static/}}
\setlength{\headheight}{26pt}
\setlength{\headsep}{12pt}
\usepackage[a4paper,top=22mm,bottom=22mm,left=25mm,right=25mm]{geometry}

\newcommand{\neowayheaderlogo}{\includegraphics[scale=0.25]{header-logo.png}}
\fancypagestyle{normal}{
\fancyhf{}
\fancyhead[L]{\neowayheaderlogo}
\fancyhead[R]{\nouppercase{\rightmark}}
\fancyfoot[L]{深圳市有方科技股份有限公司版权所有}
\fancyfoot[R]{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
}
\fancypagestyle{plain}{
\fancyhf{}
\fancyhead[L]{\neowayheaderlogo}
\fancyhead[R]{\nouppercase{\rightmark}}
\fancyfoot[L]{深圳市有方科技股份有限公司版权所有}
\fancyfoot[R]{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
}
"",
'maketitle': r"""
\thispagestyle{empty}
\pagenumbering{gobble}
\begin{titlepage}
\begin{tikzpicture}[remember picture, overlay]
\node[anchor=north west, inner sep=0pt] at (current page.north west)
{\includegraphics[width=\paperwidth,height=\paperheight]{_common/_static/background.png}};
\end{tikzpicture}
\vspace*{8cm}
\begin{flushleft}
\color[HTML]{70AD47}\fontsize{42}{48}\selectfont \textbf{N706B} \\ [0.8cm]
\fontsize{28}{32}\selectfont AT 命令手册 \\ [0.6cm]
\large 版本 V1.4 \hspace{1em} 日期 2025年11月06日
\end{flushleft}
\end{titlepage}
\clearpage
\pagenumbering{roman}
""",
}
latex_additional_files = [
'_common/_static/logo.png',
'_common/_static/background.png',
]

```

```

'_common/_static/header-logo.png'
]
latex_documents = [
    ('index', 'Neoway_N706B_Manual.tex', project, author, 'manual'),
]
'cm'>
# >>> BEGIN: NEOWAY_LATEX_BLOCK
'cm'># 自动注入时间: 2025-11-08 21:05:44
latex_engine = 'xelatex'
latex_additional_files = [
    '../../_common/_static/logo.png',
    '../../_common/_static/background.png',
    '../../_common/_static/header-logo.png'
]
latex_documents = [('index', 'Neoway_N706B_Manual.tex', 'Neoway N706B AT 命令手册',
'Neoway 文档工程组', 'manual')]

latex_elements = globals().get('latex_elements', {})

latex_elements.update({
    'papersize': 'a4paper',
    'pointsize': '11pt',
    'extraclassoptions': 'openany,oneside',
    'geometry':
        r'\usepackage[a4paper,top=22mm,bottom=22mm,left=22mm,right=22mm,headheight=18pt]
{geometry}',
    'fontpkg': r'''
\usepackage{xecjk}
\setCJKmainfont{PingFang SC}
\setmainfont{Times New Roman}
\setsansfont{Arial}
\setmonofont{Menlo}
''',
    'preamble': r'''
\usepackage{graphicx,tikz,eso-pic,xcolor,fancyhdr,titlesec,hyperref}
\graphicspath{{./}{../../_common/_static/}{_common/_static/}}
\setlength{\headheight}{24pt}
\setlength{\headsep}{12pt}

\hypersetup{
    pdftitle={ Neoway N706B AT 命令手册 },
    pdfauthor={ Neoway 文档工程组 },
    pdfsubject={ Neoway 机密 | N706B | V1.4 },
    colorlinks=true, linkcolor=blue, urlcolor=blue
}
\newcommand{\neowayheaderlogo}{\includegraphics[scale=0.25]{header-logo.png}}
\makeatletter
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}%
\renewcommand{\sectionmark}[1]{\markright{#1}}%
\makeatother

\fancypagestyle{normal}{%
    \fancyhf{}%
}

```

```

\fancyhead[L]{\neowayheaderlogo}%
\fancyhead[R]{\nouppercase{\rightmark}}%
\fancyfoot[L]{深圳市有方科技股份有限公司版权所有}%
\fancyfoot[R]{\thepage}%
\renewcommand{\headrulewidth}{0.4pt}%
\renewcommand{\footrulewidth}{0.4pt}%
}

\fancypagestyle{plain}{%
\fancyhf{}%
\fancyhead[L]{\neowayheaderlogo}%
\fancyhead[R]{\nouppercase{\rightmark}}%
\fancyfoot[L]{深圳市有方科技股份有限公司版权所有}%
\fancyfoot[R]{\thepage}%
\renewcommand{\headrulewidth}{0.4pt}%
\renewcommand{\footrulewidth}{0.4pt}%
}

\let\cleardoublepage\clearpage
'',
'maketitle': r"""
%% ----- Neoway 封面 -----
\thispagestyle{empty}
\pagenumbering{gobble}
\begin{titlepage}
\begin{tikzpicture}[remember picture, overlay]
\node[anchor=north west, inner sep=0pt] at (current page.north west)
{\includegraphics[width=\paperwidth,height=\paperheight]{_common/_static/background.png}};
\end{tikzpicture}
\vspace*{8cm}
\begin{flushleft}
\color[HTML]{70AD47}\fontsize{42}{48}\selectfont \textbf{N706B}\|[0.8cm]
\fontsize{28}{32}\selectfont AT 命令手册}\|[0.6cm]
\large 版本 V1.4 \hspace{1em} 日期 2025年11月08日
\end{flushleft}
\end{titlepage}
\clearpage
\pagenumbering{roman}
"""
,
})
'cm'># <<< END: NEOWAY_LATEX_BLOCK

```

## docs/N706B/source/index.rst

Neoway AT 命令手册 v1.4

=====

:Author: 文档工程组

:Date: 2025-11-08

:Version: v1.4

```
.. toctree:::  
    :maxdepth: 1  
    :titlesonly:  
  
    1/index  
  
    2/index
```

## 附录

```
----  
.. toctree:::  
    :maxdepth: 1  
    :titlesonly:  
  
    appendix/abbreviations  
    appendix/index
```

**docs/N725/conf.py**

```
from docs._common.conf_common import *  
  
project = "Neoway N725 AT 命令手册"  
model = "N725"  
pdf_documents = [  
    ("index", f"Neoway_{model}_AT_Commands_Manual", project, author)  
]
```

**docs/\_common/conf\_common.py**

```
'cm'># =====  
'cm'># 📄 Neoway Docs Common Config (共享基础配置)  
'cm'># 统一跨项目的 Sphinx/LaTeX 公共配置  
'cm'># =====  
from pathlib import Path  
import os  
import sys  
import platform  
from datetime import datetime  
'cm'>  
# === 路径定义 ===  
COMMON_ROOT = Path(__file__).resolve().parent  
COMMON_STATIC_PATH = COMMON_ROOT / "_static"  
'cm'>  
# 确保 _static 在路径中可访问  
sys.path.insert(0, str(COMMON_ROOT))  
'cm'>  
# === 通用信息 ===  
author = "Neoway 文档工程组"
```

```

copyright = f"{{datetime.now().year}}, Neoway Technology"
language = "zh_CN"
'cm'>
# === HTML 静态资源 ===
html_static_path = [str(COMMON_STATIC_PATH)]
html_logo = str(COMMON_STATIC_PATH / "logo.png")
'cm'>
# === LaTeX 资源路径 (公共静态引用) ===
latex_engine = "xelatex"
latex_additional_files = [
    str(COMMON_STATIC_PATH / "logo.png"),
    str(COMMON_STATIC_PATH / "header-logo.png"),
    str(COMMON_STATIC_PATH / "background.png"),
]
'cm'>
# === 跨平台字体自动识别 ===
sys_name = platform.system().lower()
if "windows" in sys_name:
    zh_main, zh_sans, zh_mono = "SimSun", "SimHei", "FangSong"
    en_main, en_sans, en_mono = "Times New Roman", "Arial", "Consolas"
elif "darwin" in sys_name: # macOS
    zh_main, zh_sans, zh_mono = "PingFang SC", "STHeiti", "PingFang SC"
    en_main, en_sans, en_mono = "Times New Roman", "Arial", "Menlo"
else: # Linux
    zh_main, zh_sans, zh_mono = "Noto Sans CJK SC", "Noto Sans CJK SC", "Noto Sans Mono
CJK SC"
    en_main, en_sans, en_mono = "Times New Roman", "Arial", "DejaVu Sans Mono"
'cm'>
# === 公共 LaTeX 元素 (在各子 conf.py 里继承 update() 即可) ===
latex_elements = {
    'papersize': 'a4paper',
    'pointsize': '11pt',
    'extraclassoptions': 'openany,oneside',
    'geometry':
r'\usepackage[a4paper,top=22mm,bottom=22mm,left=25mm,right=25mm,headheight=25pt]
{geometry}',

    'fontpkg': rf'''
\usepackage{{xeCJK}}
\setCJKmainfont{{zh_main}}
\setCJKsansfont{{zh_sans}}
\setCJKmonofont{{zh_mono}}
\setmainfont{{en_main}}
\setsansfont{{en_sans}}
\setmonofont{{en_mono}}
\linespread{{1.3}}
''',
    'preamble': r'''
\usepackage{fancyhdr}
\usepackage{titlesec}
\usepackage{tocloft}
\usepackage{hyperref}
\usepackage{setspace}

```

```
\usepackage{graphicx}
\usepackage{xcolor}
\usepackage{tikz}

% ===== 页眉页脚 =====
\pagestyle{fancy}
\fancyhf{}
\fancyhead[L]{\includegraphics[scale=0.25]{../../_common/_static/header-logo.png}}
\fancyhead[R]{\leftmark}
\fancyfoot[L]{深圳市有方科技股份有限公司版权所有}
\fancyfoot[R]{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
\setlength{\headheight}{25pt}

% ===== 中文目录与章节 =====
\renewcommand{\contentsname}{\centering 目~录}
\titleformat{\chapter}{\Huge\bfseries}{第\,\,\,\thechapter\,章}{1em}{}
\let\cleardoublepage\clearpage
...
}
```

## docs/en/conf.py

```
from docs._common.conf_common import *

language = "en"
project = "Neoway AT Commands Manual"
pdf_language = "en"
```

## docs/zh\_CN/conf.py

```
from docs._common.conf_common import *

language = "zh_CN"
project = "Neoway AT 命令手册"
pdf_language = "zh_CN"
```

## tools/build\_pdf.py

```
'cm'># =====
'cm'># Neoway Build PDF v7.5 - 统一封面注入 + 平台字体 + 自动导出
'cm'># =====
import os, re, shutil, subprocess, platform, sys
from pathlib import Path
from datetime import datetime
'cm'>
# ✅ 确保能导入 tools 内的模块
```

```

CURRENT_DIR = Path(__file__).resolve().parent
PROJECT_ROOT = CURRENT_DIR.parent
sys.path.insert(0, str(PROJECT_ROOT))
'cm'>
# ✅ 导入 CSV → RST 构建脚本
from tools.render_rst import *
'cm'>
# === 基础信息 ===
LANG = "zh"
MODEL_NAME = "N706B"
VERSION = "v1.4"

DOC_TYPE_CN = "AT 命令手册"
AUTHOR_CN = "Neoway 文档工程组"

DATE = datetime.now()
DATE_CN = DATE.strftime("%Y年%m月%d日")
VERSION_TAG = VERSION.strip().replace("v", "V")
'cm'>
# === 语言选择 ===
DOC_TYPE = DOC_TYPE_CN
AUTHOR = AUTHOR_CN
DATE_SHOW = DATE_CN
TITLE = f"Neoway {MODEL_NAME} {DOC_TYPE_CN}"
SUBJECT = f"Neoway 机密 | {MODEL_NAME} | {VERSION_TAG}"
'cm'>
# === 路径定位 ===
PROJECT_ROOT = Path.cwd()
for p in [PROJECT_ROOT] + list(PROJECT_ROOT.parents):
    if (p / "docs" / MODEL_NAME / "source" / "conf.py").exists():
        PROJECT_ROOT = p
        break
else:
    raise FileNotFoundError("❌ 未找到 conf.py, 请确认 docs/{MODEL_NAME}/source 目录结构正确。")

ROOT_DIR = PROJECT_ROOT / "docs" / MODEL_NAME / "source"
BUILD_DIR = PROJECT_ROOT / "docs" / MODEL_NAME / "build"
LATEX_DIR = BUILD_DIR / "latex"
PDF_DIR = BUILD_DIR / "pdf"
PDF_DIR.mkdir(parents=True, exist_ok=True)
conf_path = ROOT_DIR / "conf.py"
'cm'>
# ✅ 自动生成 RST 步骤
print("🛠 生成 RST 文件中 (CSV → RST) . . .")
subprocess.run(["python", "tools/render_rst.py"], check=True)
print("✅ RST 生成完成, 准备构建 PDF。")
'cm'>
# === 平台字体选择 ===
sys_name = platform.system().lower()
if "darwin" in sys_name or "mac" in sys_name:
    zh_font = "PingFang SC"
    mono_font = "Menlo"
elif "win" in sys_name:
    zh_font = "Microsoft YaHei"

```

```

    mono_font = "Consolas"
else:
    zh_font = "Noto Sans CJK SC"
    mono_font = "DejaVu Sans Mono"
'cm'>
# === 封面模板 ===
cover_block = rf"""
%% ----- Neoway 封面 -----
\thispagestyle{{empty}}
\pagenumbering{{gobble}}
\begin{{titlepage}}
    \begin{{tikzpicture}}[remember picture, overlay]
        \node[anchor=north west, inner sep=0pt] at (current page.north west)
            {{\includegraphics[width=\paperwidth,height=\paperheight]{{_common/_static/
background.png}}}};
    \end{{tikzpicture}}
    \vspace*{{8cm}}
    \begin{{flushleft}}
        {{\color[HTML]{{70AD47}}}\fontsize{{42}}{{48}}\selectfont \textbf{{{MODEL_NAME}}}}\\
[0.8cm]
        {{\fontsize{{28}}{{32}}\selectfont {DOC_TYPE}}}\\
        {{\large 版本 {VERSION_TAG} \hspace{{1em}} 日期 {DATE_SHOW}}}
    \end{{flushleft}}
\end{{titlepage}}
\clearpage
\pagenumbering{{roman}}
"""
'cm'>
# === 注入标记 ===
marker_begin = "# >>> BEGIN: NEOWAY_LATEX_BLOCK"
marker_end = "# <<< END: NEOWAY_LATEX_BLOCK"

conf_txt = conf_path.read_text(encoding="utf-8")
conf_txt = re.sub(rf"{re.escape(marker_begin)}.*?{re.escape(marker_end)}",
"", conf_txt, flags=re.DOTALL)

latex_block = f"""
{marker_begin}
# 自动注入时间: {datetime.now():%Y-%m-%d %H:%M:%S}
latex_engine = 'xelatex'
latex_additional_files =
    '../{_common/_static/logo.png}',
    '../{_common/_static/background.png}',
    '../{_common/_static/header-logo.png}'
]
latex_documents = [('index', 'Neoway_{MODEL_NAME}_Manual.tex', '{TITLE}', '{AUTHOR}', 'manual')]

latex_elements = globals().get('latex_elements', {})
"""

latex_elements.update({
    'papersize': 'a4paper',
    'pointsize': '11pt',
    'extraclassoptions': 'openany,oneside',
    'geometry': r'\usepackage[a4paper,top=22mm,bottom=22mm, left=22mm, right=22mm, headhei

```

```

ght=18pt]{{geometry}}',
  'fontpkg': r'''
\\usepackage{{xeCJK}}
\\setCJKmainfont{{zh_font}}
\\setmainfont{{Times New Roman}}
\\setsansfont{{Arial}}
\\setmonofont{{mono_font}}
  '',
  'preamble': r'''
\\usepackage{{graphicx,tikz,eso-pic,xcolor,fancyhdr,titlesec,hyperref}}
\\graphicspath{{{{.}}}{../../_common/_static/}{_common/_static/}}}
\\setlength{{\\headheight}}{{24pt}}
\\setlength{{\\headsep}}{{12pt}}


\\hypersetup{{
  pdftitle={{ {TITLE} }},
  pdfauthor={{ {AUTHOR} }},
  pdfsubject={{ {SUBJECT} }},
  colorlinks=true, linkcolor=blue, urlcolor=blue
}}


\\newcommand{{\\neowayheaderlogo}}{{\\includegraphics[scale=0.25]{{header-logo.png}}}}
\\makeatletter
\\renewcommand{{\\chaptermark}}[1]{{\\markboth{{#1}}{}}}
\\renewcommand{{\\sectionmark}}[1]{{\\markright{{#1}}}}
\\makeatother


\\fancypagestyle{{normal}}{%
  \\fancyhf{{}}
  \\fancyhead[L]{{\\neowayheaderlogo}}
  \\fancyhead[R]{{\\nouppercase{{\\rightmark}}}}
  \\fancyfoot[L]{{深圳市有方科技股份有限公司版权所有}}
  \\fancyfoot[R]{{\\thepage}}
  \\renewcommand{{\\headrulewidth}}{{0.4pt}}
  \\renewcommand{{\\footrulewidth}}{{0.4pt}}
}

\\fancypagestyle{{plain}}{%
  \\fancyhf{{}}
  \\fancyhead[L]{{\\neowayheaderlogo}}
  \\fancyhead[R]{{\\nouppercase{{\\rightmark}}}}
  \\fancyfoot[L]{{深圳市有方科技股份有限公司版权所有}}
  \\fancyfoot[R]{{\\thepage}}
  \\renewcommand{{\\headrulewidth}}{{0.4pt}}
  \\renewcommand{{\\footrulewidth}}{{0.4pt}}
}

\\let\\cleardoublepage\\clearpage
 '',
 'maketitle': r\"{{cover_block}\",\n}})\n{marker_end}
'''


conf_path.write_text(conf_txt.rstrip() + "\n\n" + latex_block + "\n", encoding="utf-8")
print(f"✓ 已更新 {conf_path} ({LANG.upper()} 版封面 + 页眉LOGO) ")
'cm'>

```

```

# === 构建 Sphinx LaTeX ===
subprocess.run(["sphinx-build", "-b", "latex", str(ROOT_DIR), str(LATEX_DIR)],
check=True)
'cm'>
# === 同步公共资源 ===
common_static = PROJECT_ROOT / "docs" / "_common" / "_static"
dest_common = LATEX_DIR / "_common" / "_static"
if common_static.exists():
    shutil.copytree(common_static, dest_common, dirs_exist_ok=True)
    print(f"✅ 已复制公共资源到 {dest_common}")

os.chdir(LATEX_DIR)
tex_main = next(LATEX_DIR.glob("*.tex"))
'cm'>
# === 编译两轮 XeLaTeX ===
for i in range(2):
    print(f"⌚ XeLaTeX 第 {i+1}/2 轮 ...")
    subprocess.run(["xelatex", "-interaction=nonstopmode", tex_main.name], check=True)
'cm'>
# === 输出 PDF 命名 ===
version_label = VERSION_TAG.lstrip("Vv")
output_filename = f"Neoway_{MODEL_NAME}_{DOC_TYPE}_V{version_label}.pdf".replace(" ", "_")
out_pdf = PDF_DIR / output_filename

pdfs = sorted(LATEX_DIR.glob("*.pdf"), key=lambda p: p.stat().st_mtime, reverse=True)
if pdfs:
    shutil.copy2(pdfs[0], out_pdf)
    print(f"🎉 成功生成 PDF : {out_pdf}")
else:
    print("❌ 未生成 PDF, 请检查 LaTeX 日志。")

```

## tools/render\_rst.py

```

'cm'># -----
'cm'># 📄 Neoway Render RST v8.0 – 模板外置版 (修复重复命令格式问题)
'cm'># -----
import pandas as pd
from jinja2 import Environment, FileSystemLoader
from datetime import datetime
from pathlib import Path
import json
'cm'>
# === 基本项目信息 ===
PROJECT_NAME = "Neoway AT 命令手册"
VERSION = "v1.4"
AUTHOR = "文档工程组"
DATE = datetime.now().strftime("%Y-%m-%d")
'cm'>
# === 定位项目根目录 ===
PROJECT_ROOT = Path.cwd()
for parent in [PROJECT_ROOT] + list(PROJECT_ROOT.parents):

```

```

if (parent / "csv-input").exists():
    PROJECT_ROOT = parent
    break
'cm'>
# === 路径定义 ===
CSV_PATH = PROJECT_ROOT / "csv-input" / "at_N706B.csv"
ROOT_DIR = PROJECT_ROOT / "docs" / "N706B" / "source"
OUTPUT_DIR = ROOT_DIR
TEMPLATE_DIR = PROJECT_ROOT / "docs" / "_common" / "templates"

OUTPUT_DIR.mkdir(parents=True, exist_ok=True)
'cm'>
# === 模板加载环境 ===
env = Environment(loader=FileSystemLoader(str(TEMPLATE_DIR)))
env.globals.update(max=max, min=min, len=len)
'cm'>
# === 加载模板 ===
main_index_tmpl = env.from_string("""
{{ project_name }} {{ version }}
{{ "=" * (project_name|length + version|length + 1) }}

:Author: {{ author }}
:Date: {{ date }}
:Version: {{ version }}

.. toctree::
   :maxdepth: 1
   :titlesonly:

{% for chap in chapters %}
   {{ chap[0] }}/index
{% endfor %}

附录
-----
.. toctree::
   :maxdepth: 1
   :titlesonly:

   appendix/abbreviations
   appendix/index
""".strip())

chapter_index_tmpl = env.from_string("""
第{{ chap_num }}章 {{ chap_name }}
{{ "=" * max((chap_name|length + 6), 10) }}

.. toctree::
   :maxdepth: 1
   :titlesonly:

{% for cmd in cmds %}
   {{ cmd }}
{% endfor %}
""")

```

```

    """.strip())
'cm'>
# === 外部命令页模板（修复重复命令格式） ===
cmd_tmpl = env.get_template("command_page.j2")
'cm'>
# === 读取 CSV 数据 ===
df = pd.read_csv(CSV_PATH, dtype=str).fillna("")
'cm'>
# === 按章节分组 ===
chapters = []
for chap, group in df.groupby("章节", sort=True):
    chap_name = group["章节名称"].iloc[0].strip() or f"第{chap}章"
    chapters.append((chap, chap_name, group))
'cm'>
# === 渲染每个章节 ===
for chap_num, chap_name, group in chapters:
    chap_dir = OUTPUT_DIR / str(chap_num)
    chap_dir.mkdir(parents=True, exist_ok=True)

    cmd_list = []
    for _, row in group.iterrows():
        cmd_name = row["命令"].strip()
        cmd_title = row["命令标题"].strip()
        cmd_file = chap_dir / f"{cmd_name}.rst"
        cmd_list.append(cmd_name)

    try:
        params = json.loads(row["参数json"])
    except Exception:
        params = {}

    rendered = cmd_tmpl.render(
        cmd_name=cmd_name,
        cmd_title=cmd_title,
        desc=row["功能描述"],
        cmd_format=row["命令格式"],
        cmd_response=row["响应"],
        params=params,
        note=row.get("备注", ""),
        example=row.get("示例命令", ""),
    )

    cmd_file.write_text(rendered.strip() + "\n", encoding="utf-8")
    print(f"✅ 已生成命令 : {cmd_name}")
'cm'>
# === 渲染章节 index ===
idx_path = chap_dir / "index.rst"
idx_render = chapter_index_tmpl.render(
    chap_num=chap_num,
    chap_name=chap_name,
    cmd_list=cmd_list
)
idx_path.write_text(idx_render.strip() + "\n", encoding="utf-8")
print(f"📘 第{chap_num}章 {chap_name} 生成完成 ({len(cmd_list)} 条命令)")

```

```
'cm'>
# === 渲染主 index.rst ===
main_rst = main_index_tmpl.render(
    project_name=PROJECT_NAME,
    version=VERSION,
    author=AUTHOR,
    date=DATE,
    chapters=chapters
)
(OUTPUT_DIR / "index.rst").write_text(main_rst.strip() + "\n", encoding="utf-8")
print(f"🔴 主 index.rst 生成完成 → {OUTPUT_DIR/'index.rst'}")
```

## tools/validate\_csv\_schema.py

```
'cm'>#!/usr/bin/env python3
'cm'># -*- coding: utf-8 -*-
"""

validate_csv_schema.py
验证 csv-input 目录下所有 CSV 文件结构一致性：
- 列名是否一致
- 字段数是否正确
- 引号、逗号配对问题
- 空行/缺值检测
"""

import csv
import os
from pathlib import Path
import sys

def validate_csv_schema(csv_dir: Path):
    csv_files = list(csv_dir.glob("*.csv"))
    if not csv_files:
        print("🔴 未找到 CSV 文件 :{csv_dir}")
        sys.exit(1)

    print("🔍 开始验证 CSV 文件结构, 共 {len(csv_files)} 个文件...")
    reference_header = None
    errors = []

    for csv_file in csv_files:
        print(f"📝 检查 {csv_file.name} ...")
        with open(csv_file, "r", encoding="utf-8-sig") as f:
            reader = csv.reader(f)
            try:
                header = next(reader)
            except Exception as e:
                errors.append(f"🔴 {csv_file.name} 无法读取表头 :{e}")
                continue

        # 初始化参考表头
        if reference_header is None:
```

```
reference_header = header
print(f"✅ 模板表头 : {reference_header}")
else:
    if header != reference_header:
        errors.append(
            f"⚠️ {csv_file.name} 表头不一致 :\n 预期: {reference_header}\n 实际: {header}"
        )
'cm'>
# 检查字段数一致性
for i, row in enumerate(reader, start=2):
    if len(row) != len(reference_header):
        errors.append(
            f"⚠️ {csv_file.name} 第 {i} 行字段数不符: {len(row)} != {len(reference_header)}"
        )
if errors:
    print("\n❌ 验证发现以下问题:")
    for e in errors:
        print(e)
    sys.exit(1)
else:
    print("\n✅ 所有 CSV 文件结构一致, 验证通过。")

if __name__ == "__main__":
    project_root = Path(__file__).resolve().parents[1]
    csv_dir = project_root / "csv-input"
    validate_csv_schema(csv_dir)
```