

## Table of Contents

|                                                                                                                   |   |
|-------------------------------------------------------------------------------------------------------------------|---|
| <b>Dragon 16.1 Cheat Sheet —When to Use SendKeys, Send Dragon Keys, or System Keys, and details about braces.</b> | 2 |
|  <b>The Three Input Methods</b>  | 3 |
| 1. When to Use SendDragonKeys                                                                                     | 3 |
| <b>✓ Use it when:</b>                                                                                             | 3 |
| <b>✗ Do NOT use it when:</b>                                                                                      | 3 |
| 2. When to Use SendKeys                                                                                           | 4 |
| <b>✓ Use it when:</b>                                                                                             | 4 |
| <b>✗ Avoid it when:</b>                                                                                           | 4 |
| 3. When to Use System-Level Keys                                                                                  | 4 |
| <b>✓ Use them when:</b>                                                                                           | 4 |
| <b>✗ Do NOT use them when:</b>                                                                                    | 4 |
| 4. Quick Decision Guide                                                                                           | 5 |
| <b>If you're typing text → use SendDragonKeys</b>                                                                 | 5 |
| <b>If Dragon's keys don't work → try SendKeys</b>                                                                 | 5 |
| <b>If you need OS-level shortcuts → use system keys</b>                                                           | 5 |
| 5. Practical Examples                                                                                             | 5 |
| <b>Text entry (best practice)</b>                                                                                 | 5 |
| <b>Navigation</b>                                                                                                 | 5 |
| <b>Literal braces</b>                                                                                             | 5 |
| <b>When Dragon fails (fallback)</b>                                                                               | 5 |
| <b>System-level</b>                                                                                               | 5 |
| 6. Golden Rules for Reliability                                                                                   | 6 |
| When do I know to sendDragonkeys as opposed to just sendkeys.                                                     | 6 |
| Use SendDragonKeys when:                                                                                          | 6 |
| 1. Timing matters                                                                                                 | 6 |
| 2. You need literal text output                                                                                   | 6 |

|                                                             |    |
|-------------------------------------------------------------|----|
| 3. The target app is slow, Electron-based, or UI-heavy..... | 7  |
| Use SendKeys when:.....                                     | 7  |
| The Practical Rule You Can Use Instantly.....               | 7  |
| What about Curly Braces .....                               | 9  |
| When You DO Need Curly Braces {}.....                       | 9  |
| When You do NOT Need Curly Braces {}.....                   | 9  |
| The One-Line Rule.....                                      | 10 |
| 1. Modifier Symbols (No Braces Needed).....                 | 11 |
| 2. When to Use SendKeys vs. SendDragonKeys .....            | 11 |
| 3. Keys That REQUIRE Curly Braces {} .....                  | 12 |
| Dragon basics .....                                         | 13 |

## **Dragon 16.1 Cheat Sheet —When to Use SendKeys, Send Dragon Keys, or System Keys, and details about braces.**

## The Three Input Methods

Dragon scripting gives you **three different ways** to send keystrokes:

| Method                                                | Purpose                                           | Strengths                            | Weaknesses                                      |
|-------------------------------------------------------|---------------------------------------------------|--------------------------------------|-------------------------------------------------|
| <b>SendKeys</b>                                       | Standard Windows keystroke injection              | Simple, universal                    | Slow, unreliable in modern apps, easily blocked |
| <b>SendDragonKeys</b>                                 | Dragon's internal, high-priority keystroke engine | Fast, reliable, respects text fields | Cannot send some system keys; timing-sensitive  |
| <b>System Keys</b> (e.g., AppBringUp, SendSystemKeys) | OS-level key events                               | Works when apps ignore normal keys   | Can bypass UI focus; dangerous if misused       |

### 1. When to Use SendDragonKeys

Use **SendDragonKeys** for **99% of text-entry and editing actions**.

#### ✓ Use it when:

- You are typing **text** into any field `SendDragonKeys "Hello world"`
- You need **fast, reliable navigation keys** `SendDragonKeys "{Left} {Left} {Ctrl+End}"`
- You are automating **Word, Notepad, browsers, editors, IDEs**
- You need **repeatable behavior** across machines
- You want **Dragon-timed keystrokes** that avoid double-sending
- You need **Unicode-safe** output (SendKeys sometimes corrupts characters)

#### ✗ Do NOT use it when:

- You need to send **system-level keys** like:
  - {Alt+Tab}
  - {Ctrl+Shift+Esc}
  - {Win+R}
- You need to interact with **non-standard UI elements** (Electron, Qt, Java apps) that ignore Dragon's keystrokes

- You need **literal curly braces** (use SendDragonKeys "{" or Spell commands)

## 2. When to Use SendKeys

Use **SendKeys** only when **SendDragonKeys fails** or when the target app **requires Windows-native key events**.

### ✓ Use it when:

- You're scripting **older Windows apps** that expect classic Win32 messages
- You need to send **literal characters** that Dragon interprets (e.g., braces)
- You're interacting with **dialogs** that ignore Dragon keys
- You need **slow, deliberate keystrokes** (SendKeys is slower by design)

### ✗ Avoid it when:

- Working in **modern apps** (VS Code, Chrome, Edge, Electron apps)
- You need **speed**
- You need **reliability**
- You need **modifier combos** (SendDragonKeys handles them better)

## 3. When to Use System-Level Keys

These include:

- SendSystemKeys
- AppBringUp
- Windows shell commands
- OS-level shortcuts

### ✓ Use them when:

- You need **true OS-level keystrokes** that bypass the app
- You need to trigger **global shortcuts**
  - {Win+R}
  - {Ctrl+Shift+Esc}
  - {Alt+Tab}
- You need to **launch apps** reliably
  - AppBringUp "notepad.exe"
- You need to interact with **secure desktops** (UAC prompts, lock screen)

### ✗ Do NOT use them when:

- You're typing text
- You're editing documents
- You're navigating inside an app
- You need predictable cursor behavior
- You're working in a **sandboxed** or **sensitive** environment

## 4. Quick Decision Guide

**If you're typing text → use `SendDragonKeys`**

**If Dragon's keys don't work → try `SendKeys`**

**If you need OS-level shortcuts → use system keys**

## 5. Practical Examples

### **Text entry (best practice)**

```
SendDragonKeys "Hello{Enter}"
```

### **Navigation**

```
SendDragonKeys "{Ctrl+Home}{Shift+End}"
```

### **Literal braces**

```
SendDragonKeys "\\{example\\}"
```

### **When Dragon fails (fallback)**

```
SendKeys "^a"
```

### **System-level**

```
SendSystemKeys "{Alt+Tab}"  
AppBringUp "notepad.exe"
```

## 6. Golden Rules for Reliability

- **Default to SendDragonKeys** — it's the most stable in Dragon 16.1
- **Use SendKeys only as a fallback**
- **Use system keys only for OS-level actions**
- **Never mix SendKeys and SendDragonKeys in the same line**
- **Add delays only when absolutely necessary**
- **Avoid SendKeys in Electron apps** (VS Code, Discord, Slack, Obsidian)

## When do I know to sendDragonkeys as opposed to just sendkeys.

Below is the version that will actually help you decide in the moment.

 The Real Difference: SendKeys vs. SendDragonKeys

### Use SendDragonKeys when:

#### 1. Timing matters

Dragon's internal engine queues and synchronizes SendDragonKeys with speech recognition.

This means:

- It waits for the target window to be ready
- It avoids dropped keystrokes
- It handles modifier timing better (Ctrl/Shift/Alt)
- It's less likely to "outrun" the UI
- If the shortcut is sensitive—like Ctrl+Shift+W in an app that reacts instantly—SendDragonKeys is safer.

#### 2. You need literal text output

SendDragonKeys is designed for text entry, not just keystrokes.

It handles:

- Unicode

- Special characters
- Dragon's own escape sequences

If you're typing text into an editor, `SendDragonKeys` is the correct tool.

### 3. The target app is slow, Electron-based, or UI-heavy

#### Use `SendKeys` when:

1. You need raw Windows keystrokes

`SendKeys` sends keystrokes directly through Windows messaging.

This is useful when:

- The app expects a literal Windows shortcut
- You're not typing text
- Timing is not critical
- The shortcut is simple (`Ctrl+C`, `Ctrl+W`, etc.)

2. You're launching a helper EXE

If you're using a UIA helper program (which you often do), then:

Dragon → `ShellExecute` → helper EXE → `SendKeys.SendWait()`

In that case, Dragon's own `SendKeys` is fine because the helper handles timing.

### The Practical Rule You Can Use Instantly

- If you're sending text → use `SendDragonKeys`.
- If you're sending a shortcut → try `SendKeys` first.
- If the shortcut is dropped or mis-timed → switch to `SendDragonKeys`.

This is exactly how most Dragon power-users end up working.

 A specific case: `Ctrl+Shift+W`

This shortcut is:

- A multi-modifier chord
- Timing-sensitive
- Often used in apps that don't like fast input

So, the best order is:

vb

`SendKeys "^+w"`

If it's unreliable, switch to:

vb

`SendDragonKeys "^+w"`

And if you want maximum resilience:

Use a helper EXE with UI Automation and call it via ShellExecute.

Are you **typing text**?

→ Yes → `SendDragonKeys`

Are you **sending a shortcut**?

→ Yes → Try `SendKeys`

Did the shortcut **drop, misfire, or race**?

→ Yes → Switch to `SendDragonKeys`

Is the target **app slow, Electron, or web-wrapped**?

→ Yes → Prefer `SendDragonKeys`

Are you using a **helper EXE**?

→ Yes → `SendKeys` is fine (helper handles timing)



# What about Curly Braces

## When You DO Need Curly Braces {}

Use curly braces when you want Dragon to send a named key, not a typed character.

Examples of keys that require braces:

- {ENTER}
- {TAB}
- {ESC}
- {DELETE}
- {HOME}
- {END}
- {LEFT} {RIGHT} {UP} {DOWN}
- {F1} through {F24}
- {PGUP} {PGDN}
- 

## When You do NOT Need Curly Braces {}

### 1. Regular characters

vb

```
SendDragonKeys "abc123"
```

```
SendKeys "hello"
```

### 2. Modifier shortcuts

(These use symbols, not braces)

^ = Ctrl

+ = Shift

% = Alt

Examples:

vb

```
SendKeys "^c" ' Ctrl+C
```

```
SendDragonKeys "^+w" ' Ctrl+Shift+W
```

```
SendKeys "%f" ' Alt+F
```

No braces needed because these are keystroke chords, not named keys.

## The One-Line Rule

Use **braces for named keys** ({ENTER}, {TAB}, {F5}),

use **symbols for shortcuts** (^+w, %s, ^c).

 A specific case: Ctrl+Shift+W

You want:

```
vb
```

```
SendKeys "^+w"
```

or, if timing matters:

```
vb
```

```
SendDragonKeys "^+w"
```

No curly braces needed, because w is a literal key and the modifiers are symbols.

## 1. Modifier Symbols (No Braces Needed)

| Modifier   | Symbol | Example | Meaning      |
|------------|--------|---------|--------------|
| Ctrl       | ^      | ^c      | Ctrl+C       |
| Shift      | +      | +a      | Shift+A      |
| Alt        | %      | %f      | Alt+F        |
| Ctrl+Shift | ^+     | ^+w     | Ctrl+Shift+W |
| Ctrl+Alt   | ^%     | ^%{TAB} | Ctrl+Alt+Tab |

Rule:

Use symbols for modifier chords. Braces only appear if the key itself is a named key (like {TAB}).

## 2. When to Use SendKeys vs. SendDragonKeys

✓ Use SendDragonKeys when:

- You are typing text
- Timing matters (slow UI, Electron apps, browsers)
- You need Unicode or special characters
- You want Dragon-synchronized, reliable output

✓ Use SendKeys when:

- Sending simple shortcuts (Ctrl+C, Ctrl+W, Alt+F)
- Timing is not critical
- A helper EXE will handle the keystroke
- You want raw Windows keystroke behavior

★ Practical rule:

Text → SendDragonKeys

Shortcuts → SendKeys (unless unreliable)

### 3. Keys That REQUIRE Curly Braces {}

Use braces for named keys:

#### Navigation & Editing

- {ENTER}
- {TAB}
- {ESC}
- {BACKSPACE}
- {DELETE}
- {INSERT}
- {SPACE}

#### Movement keys

- {LEFT} {RIGHT} {UP} {DOWN}
- {HOME} {END}
- {PGUP} {PGDN}

#### Function keys

{F1} {F2} ... {F24}

# Dragon basics

Turn mic on/off:

{wake up}, {go to sleep}, {microphone off}

Ask for help:

- {what can I say}
- {show navigation commands}
- {show correction commands}
- {show formatting commands}
- {show punctuation commands}
- {open help}

Dictation and punctuation

Plain dictation:

Just speak normally:

{this is a sample sentence}

Common punctuation:

- {period}, {comma}, {question mark}, {exclamation mark}
- {colon}, {semicolon}, {dash}, {hyphen}

Quotes and brackets:

- {open quote} ... {close quote}
- {open parenthesis} ... {close parenthesis}
- {open bracket} ... {close bracket}
- {open curly bracket} ... {close curly bracket}

(Useful when you need literal { and } in code or text.)

## Text formatting

### Capitalization:

- {cap} word → Capitalize next word
- Example: {cap} {dragon} → “Dragon”
- {all caps} word → ALL CAPS next word
- {caps on} ... {caps off} → Toggle capitalization mode

### Bold/italic/underline (in supported apps):

{bold that}, {italicize that}, {underline that}

### Font and size (app-dependent):

- {increase font size}, {decrease font size}
- {set font size} number

## Selecting and editing text

### Select text:

- {select} word/phrase
- {select that} → Select last dictated text
- {select line}, {select paragraph}

### Extend or shrink selection:

{extend selection}, {shrink selection}

### Delete and replace:

- {delete that}
- {delete line}, {delete paragraph}
- {scratch that} → Undo last utterance
- {scratch that} (repeated) → Undo multiple steps

## Correction commands

Correct recent text:

- {correct that}
- {correct} word/phrase

Choose from list:

After a correction list appears: {choose} 1–10

Spell it:

{spell that}

Then say letters, numbers, symbols, e.g.

{open curly bracket}, {close curly bracket}, {underscore}, {backslash}

## Navigation and basic editing keys

Arrow keys:

- {press left arrow}, {press right arrow}
- {press up arrow}, {press down arrow}

Line and document movement:

- {go to beginning of line}, {go to end of line}
- {go to top}, {go to bottom}

Enter, Tab, Backspace:

- {press enter}, {press tab}, {press backspace}
- {press delete}

Page navigation:

{press page up}, {press page down}

Working with applications and windows

Open programs (if known to Dragon/Windows):

{open} application name

Example: {open microsoft word}

Switch windows:

- {switch to} application/window name
- {press alt tab}

Close/minimize/maximize:

- {close window}
- {minimize window}, {maximize window}

File commands (via keystrokes):

- {press control s} → Save
- {press control o} → Open
- {press control p} → Print

Working with the web (Edge/Chrome with Dragon extension)

Address bar and navigation:

- {go to address bar} (or {press alt d})
- {press enter} to go to typed URL
- {refresh page} (or {press f5})



Tabs:

- {open new tab} (or {press control t})
- {close tab} (or {press control w})

Find on page:

{find on this page} (or {press control f})

Clicking elements:

- {click} link name
- {click link}
- {click text field} / {click edit box}
- {click button}, {click checkbox}, {click image}, {click list box}
- If there are multiple matches:

{show choices}

Numbers, dates, and special characters

Numbers and dates:

Dictate naturally:

{january fifteenth twenty twenty five}

Use {numeral} for digits if needed:

{numeral five} → “5”

Special characters (spoken forms):

- {underscore}, {slash}, {backslash}
- {asterisk}, {ampersand}, {at sign}
- {open curly bracket}, {close curly bracket}
- {less than sign}, {greater than sign}, {equal sign}

Curly braces in scripting and code contexts

When you’re dictating code, templates, or Dragon scripting:

Literal braces in text/code:

- {open curly bracket} → {
- {close curly bracket} → }

Combining with other symbols:

{open curly bracket} {cap if} {close curly bracket}

→ {If}

{open curly bracket} {underscore} {close curly bracket}

→ {\_}

For Dragon's own command syntax (like keystrokes), you typically type braces with the keyboard or use {spell that} and speak {open curly bracket} / {close curly bracket} to insert them precisely.

Quick “meta” commands

Show command sets:

- {show navigation commands}
- {show correction commands}
- {show formatting commands}
- {show punctuation commands}

Training and help (16.1):

Use Help menu or say {open help}, then follow on-screen links for “Get trained” and other topics.