Hints and Tips for the Lenovo Press CMS

Reference Information

Inserting an image

These are the steps you'd take to insert an image into a document.

Extract from Word

If the image you want is in Word, follow these steps to extract the image:

1. In Word, right click the image and click **Size and Position**
2. In the dialog box, click the **Reset** button to reset the size to 100% and click **OK**. If you don't reset to 100% then the image you extract will not be the original resolution.
3. Right click the image and click **Save Picture As**.
4. Select a folder and ensure the Save type is PNG.
5. Enter a meaningful name, eg the caption of the figure
6. Click **OK** to save the image.

Insert the image into the CMS

Now follow these steps to upload the image and insert it into the document:

1. In the CMS, put the cursor on a blank line (empty paragraph)
2. Click the **Image** icon in the editor icon bar
3. Copy paste the caption from the Word document into the Alternate Text field, or otherwise type in some summary text into this field. The alt text is used by vision-impared visitors to the web site and is required.
4. Click the **Browse Server** button
5. In the dialog that appears, navigate to the folder for the document (hint: use Ctrl-F and search for LPxxxx). If a folder doesn't exist, then right click on the Image parent directory to create a new folder. Name it the document number LPxxxx (uppercase LP).
6. With the document folder selected, click the **Upload** button
7. Click Browse and navigate to the folder where you have the PNG file, then click **OK**
8. Click **Upload Selected File**.
9. The file is now uploaded. Find the file in the amongst all the thumbnail. If you can't find the file visually, use Ctrl-F to search on the file name. (Hint: If you single click the image, you'll see the full filename at the bottom of the window.)
10. Double click the thumbnail to select the image
11. Back in the Image properties dialog, if the image is bigger than 615px, then change the Width to a number smaller than 615 (eg 600). The Height should automatically change to keep the ratio, provided the padlock icon is closed (click to close it if it is open)

12. Copy the URL to the **Link** tab - this will enable the image to be clickable
13. Click **OK** to close the dialog. The image should now appear in the document.
14. Optionally add a border around the image by double clicking the image to open the Image Properties dialog, go to the **Advanced** tab, and add the following text after the width and height styles:
    ```
    border:2px solid rgb(0,0,0);
    ```

Add the caption

Follow these steps to add the caption:

1. Select the image with a single click
2. Press the End key to move the cursor to the right edge of the image
3. Press Shift-Enter to add a line break. The cursor should now be immediately below the image. (Hint: This adds a <br /> to the code)
4. Enter the following text followed by the caption text. This ensures a figure caption with an automatic figure number.
   ```
   Figure {counter name="figure"}. Caption goes here
   ```

Enable Fancybox

Follow these steps to enable Fancybox - useful if the original image is larger than 615px wide:

1. If you didn't previously, double-click the image to open the Image properties dialog, and copy the URL to the field in the **Link** tab - this will enable the image to be clickable.
2. Select the image again and click the **Source** button
3. Find the <a> (anchor) tag that surrounds the <img> (image) tag and add the following parameters to the <a> tag - this enables FancyBox on the clickable image:
   ```
   data-fancybox="all" data-caption=""
   ```
4. In the data-caption parameter, paste/type in the figure caption - this text is displayed below the image when the visitor clicks the image.

CSS classes

Here are special classes that are useful for our documents.

Table 1. CSS classes

Class	Comment
`tight`	By default, first level lists have white space between each item. The class will remove the space. Use it on the ul or ol tag. `<ul class="tight">`
`flush`	Use with a list item when you want to reduce the left indent of the list item. Often used in conjunction with tight `<ul class="tight flush">`

Class	Comment
`loose`	By default, 2nd-level and 3rd-level (and beyond) lists do not have white space between each item. The class will add the space. Use it on the ul or ol tag. `<ul class="loose">`
`break-before`	Forces a page break in the PDF before this tag. Ignored for the web version. `<h3 class="break-before">`
`break-after`	Forces a page break in the PDF after this tag. Ignored for the web version.

`break-avoid`	Attempts to keep a block element (div, p, li, etc) from splitting between pages in the PDF. Ignored for the web version.
	Example:
	`<div class="break-avoid">`
	`<p>These two paragraphs should be kept together</p>`
	`<p>2nd paragraph</p>`
	`</div>`

`break-allow`	By default, the default for tables is`break-avoid`. To override this, use break-allow, which will allow the table to split across PDF pages, even if the table is small enough to fit on one page.

`withdrawn`	The content will be treated as withdrawn product information. The CMS has a feature where content marked as withdrawn is hidden by default. The user can show this information by clicking the checkbox on the web page, which then shows withdrawn content with a pink background. Withdrawn content is not shown in the PDF at all.
	Example (a row in a table of part numbers):
	`<tr class="withdrawn">`
	`<td>4XC7A08264</td>`
	`<td>B5SW</td>`
	`<td>ThinkSystem Marvell QL41232 10/25GbE 2-Port Adapter</td>`
	`<td>1</td>`
	`</tr>`
	Tip: There's also a button in the editor icon bar (look for the ⊘ symbol in the same row as bold and italics). If you just want a word or two marked as withdrawn (as opposed to an entire paragraph or list item), then you can use this. However, **don't** use this button for blocks of text (table cells, paragraphs, list items).
	Note: if a row is already "gray color" marking it withdrawn works, however background gray color overwrites pink in this scenario, so it stays grey.
	A smarty macro `option_withdrawn` has also been created to automate the marking of text as withdrawn based on the withdrawn date of one or more part numbers. See Smarty code for syntax.

Class	Comment

`callout`	Displays text in a grey box. Use in a div tag.
	Example:
	`<div class="callout">`
	`    <p><b>Riser 3 support</b>: The use of the onboard NVMe ports`
	`    is mutually exclusive with Riser 3.</p>`
	`</div>`
	Output:
	Riser 3 support: The use of the onboard NVMe ports is mutually exclusive with Riser 3.
	Tip: If possible, the text in bold should be something descriptive and not simply "Note". This will allow the reader to get a sense of purpose of the callout without having to read all of it.

`small`	Displays text in a smaller (8pt) font size. Applies to both PDF and web.
`pdf-small`	Displays text in the PDF in a smaller (8pt) font size. Ignored by the web version.
`img-left` `img-right`	Use these classes in img tags as alternatives to the standard`float:left` and `float:right` styles where images are aligned left or right. `img-right` will float an image right if the screen width is above the smallest screen breakpoint (>768px), and smaller than that, the image will be centered on its own. The `img-left` class floats the image left and provides the same styling below 768px. `<img class="img-right" ... >`

`nowrap`	Used with <pre> to prevent lines in a pre block from wrapping in the web version. Regardless of this class, the text in a <pre> block will always wrap in a PDF.
	`<pre class="nowrap">`
	`This text will wrap if needed when displayed as a web page</pre>`

`no-check`	use in either a <a> or <span> tag around a URL. Used to tell the automated broken link checker to ignore this URL. This is needed because some web sites will not allow bots like ours to check their pages - they return 500 or 403 even when the URL is valid.

Table formatting

Some useful code for tables.

- Table starter code
- Shading in table cell
- Table heading with rotated text
- Tables with small text
- Table footnote

Table starter code

Use this code to start a table.

```
<div class="scroll-y">
<table class="table table-bordered">
<caption>Table {counter name="table"}. Table code</caption>
<thead>
<tr>
<th width="15%">Part number</th>
<th width="15%">Feature code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
... rows here... use the TBLR Chrome tool
</tbody>
</table>
</div>
```

Tips:

- The TBLR tool ("tabler") is an excellent way to create the rows of a table from Excel. There is also a Chrome app which allows you to run tblr as a standalone Windows program.
- The width parameters in the heading row are good on part number tables to ensure that the column spacing is consistent throughout a product guide. This looks better in my opinion. For other tables, consider removing/adjusting these width parameters. The browser will figure out the widths of the other columns to make everything fit on page, so you don't have to hard-code all 100%.
- The `scroll-y` class is to ensure that a wide table is viewable even on a small screen such as a phone. Make sure you have this as a div.
- The use of `{$block_title}` for the caption is a timesaver if your block only has one table in it. Otherwise, you should change it.

Shading in table cell

Use this for heading rows throughout a table, or use it to highlight support (eg all cells with "Yes" or "Supported")

```
<td bgcolor="#c0c0c0">Yes</td>
```

Colors I use for table cells:

- Grey: #c0c0c0
- Green: lightgreen
- Blue: lightskyblue
- Yellow: #FFFF75

Storage tables:

- SATA (blue): #87CEEB
- AnyBay (purple): #af7ac5
- NVMe (red): #FF6347

Support tables:

- All supported (green): #00FF00
- Partial (yellow): #FFFF00
- Partial (orange): #FFA500

If you need to use 2 colors in a table, consider using green and blue

Table heading with rotated text

Use this code if you want text to be rotated. Note that the text has to be wrapped inside a div. The default height is 75px which is the minimum and the height will automatically adjust in the browser (via Javascript) to fit the text.

Use a style override to force the height. You can specify a larger height but not a smaller height that 75px.

use `rotate-tight` to force columns with a narrower width (16px width).

```
<th class="rotate"><div>VGA port</div></th>
<th class="rotate" style="height:210px;"><div>VGA port</div></th>
<th class="rotate-tight"><div>VGA port</div></th>
```

Tables with small text

Tables use 10pt font by default. For really complex tables, you might want to switch to a small font size so that more text can fit without extending the length of the table. Do this by adding the `table-smalltext` class to the table tag. This only affects the PDF; ignored by the web version.

```
<table class="table table-bordered table-smalltext">
```

Table footnote

A table footnote is a paragraph that is flush up against the bottom of a table. Normally, a paragraph after a table has space between them, so this class removes the space.

The trick is that the <p> tag has to be placed directly after the </table> end tag.

```
... table code
</table>
<p class="table-footnote">* Not supported in Slot 3</p>
</div>
```

Other useful HTML

Here are HTML snippets that are sometimes useful.

- Keeping words together on the same line
- Special characters
- Space between list item and indented p
- Convert a link to a button
- Tooltips
- Code block

Keeping words together on the same line

Use `&nbps;` (non-breaking space) or use a style override.

```
Keep this 1&nbsp;Gb unit together
<span style="white-space:nowrap">These words stay together</span>
<td style="white-space:nowrap">This cell won't break</td>
```

Special characters

Note that they all start with an & and end in a semicolon ;.

- &micro; = µ
- &deg; = ° degree symbol
- &dagger; = †
- &Dagger; = ‡
- &sect; = §
- &curren; = ¤
- &nbsp; = non-breaking space
- &#8209; = non-breaking hyphen ‑

If you're looking for more symbols, &what; ("amp-what") is a good resource

Space between list item and indented p

By default, if you have a paragraph inside a list item, there won't be a space between the two. Also, the two might break badly in the PDF. Add the following styles to fix:

```
<li class="break-avoid">Here is our list item
<p style="margin-top:10px;">This is the indented paragraph</p>
</li>
```

Convert a link to a button

Bootstrap provides these buttons:
https://getbootstrap.com/docs/4.0/components/buttons/

`class="btn btn-primary"` on the <a> tag turns the content into a blue button

Clickable blue button:

```
<p><a class="btn btn-primary" href="https://lenovopress.com/ds0091">Read the
datasheet</a></p>
```

Output:

Read the datasheet

Tooltips

Tooltips are messages that appear when you hover over a link or text. Add to a span or td or anchor tag.

For details see this page:
https://getbootstrap.com/docs/4.0/components/tooltips/

Example:

```
<span data-toggle="tooltip" data-placement="top" title="This is the tooltip text">Hover over me!</span>
```

Output:

Hover over me!

Direction choices are top, right, bottom, and left. Top is the default.

The title can include HTML code but you need to add `data-html="true"`.

Code block

To display lines of text as code use `<pre><code>...</code></pre>`

```
#include
int main() {
   // printf() displays the string inside quotation
   printf("Hello, World!");
   return 0;
}
```

Smarty comments

Use smarty comments instead of HTML comments, because smarty comments don't appear in the HTML stream and therefore can't be viewed by the end user who happens to be snooping around in the HTML code of the page.

HTML comments can only be added in Source mode and are not visible in Normal mode.

```
<!-- this a comment in HTML -->
```

Smarty comments can be added in either Source or Normal mode, so they are always visible in the CMS (but not published).

```
{* this is a smarty comment *}
```

No comments within comments: You can't have a smarty comment that includes another smarty comment. You'll need to add a = or some other character between the * and the } of the inner comment.

You can use smarty comments to hide an entire row. The trick is the placement of the opening and closing comment markup - either both in cell 1 (row to be hidden + the following row) or both in the last cell (preceding row + the row to be hidden).

Example 1 - Hides the first row:

Part number	**Feature code**	**Description**
{*7XB7A00036	AUUE	ThinkSystem 2.5" 1TB 7.2K SATA 6Gb Hot Swap 512n HDDHIDDEN ROW
*}7XB7A00037	AUUJ	ThinkSystem 2.5" 2TB 7.2K SATA 6Gb Hot Swap 512e HDD

Example 2 - Hides the last row:

Part number	**Feature code**	**Description**
7XB7A00036	AUUE	ThinkSystem 2.5" 1TB 7.2K SATA 6Gb Hot Swap 512n HDD{*
7XB7A00037	AUUJ	ThinkSystem 2.5" 2TB 7.2K SATA 6Gb Hot Swap 512e HDDHIDDEN ROW *}

Tip: I will often put some additional text in the smarty comment as a visual reminder that the row is hidden. I mark that reminder in red text (using the red A button in the editor).

Smarty functions

Add the following code to the Smarty block in the Document Info tab in any product guide, especially server product guides

```
{assign var=system_name value="ThinkSystem SR650 V2" scope="root"}
{assign var=system_name_short value="SR650 V2" scope="root"}
{assign var=machinetype_1year value="7Z72" scope="root"}
{assign var=machinetype_3year value="7Z73" scope="root"}
{assign var=machinetype_hana value="7D15" scope="root"}
{assign var=machinetype_nowarranty value="" scope="root"}
```

The following smarty code snippets are useful

Table 2. Smarty code

Smarty	Description
`{$system_name_short}`	Displays the system_name_short variable. Used in conjunction with the Smarty code definitions above. Example: `The {$system_name_short} is a new 1U server.`
`{$system_name}`	Displays the system_name variable. Used in conjunction with the Smarty code definitions above.
`{counter name="figure"}`	An automatically incremented counter for figures. Example: `Figure {counter name="figure"}. This is the figure caption`
`{counter name="table"}`	An automatically incremented counter for tables. Example: `Table {counter name="figure"}. This is the table caption`
`{$block_title}`	Returns the text of the current block title. Useful in a table caption if you want the table to match the block name.
`{$block_title|slugify}`	If you want to use the block_title variable in the id parameter in a heading tag, you can make it uniform and URL-friendly by piping the variable through our slugify modifier. Example: `<h3 id="internal-drives-{$block_title\|slugify}">Internal drives for {$block_title}</h3>`
`{link href=""}`	Use this as a way to display a URL. Useful because it ensures that the displayed URL and the actual URL behind the click are identical. Example: `{link href="https://lenovopress.com"}` Add parameter `no-check=true` to instruct the CMS not to check this URL if it is broken.
`{$minitoc}`	Inserts of bullet list of all the <h3> headings in the current section. The list items are clickable to allow the user to jump down to that heading. It assumes (requires) that the headings are unique in the document and does not test for that.
`{option_desc part=4XC7A08316}`	Returns the description of the specified part number or feature code. This data comes from the Server Support file that is uploaded into the Server Support section of the CMS. Returns nothing if the part number is missing.

Smarty	Description

`class="{option_withdrawn part=4XC7A08316}"`
`class="{option_withdrawn part='4XC7A08316 4XC7A08317'}"`
	Returns the string "withdrawn" if the part number is withdrawn, based on today's date and the date of withdrawal of this part. Use this macro inside a class parameter. This withdrawal date comes from the Server Support file that is uploaded into the Server Support section of the CMS. Returns "" if the part number is not withdrawn.
	Optionally, can contain multiple part numbers and will return "withdrawn" only if ALL parts (logical AND) are withdrawn. This is useful for table heading rows - the heading can be withdrawn once all the parts in the subsection are withdrawn. The list of parts must be wrapped in quotes. The delimiter between parts can be a space, or any of comma, pipe, colon, semicolon.

`{mbrgo}`	Used to hide text before or after a specific date and time (embargo). If time isn't specified, then defaults to 00:00. Times are in EST timezone.
	`{mbrgo bf="yyyy-mm-dd hh:00"}text to show before embargo{/mbargo}`
	`{mbrgo af="yyyy-mm-dd hh:00"}text to show after embargo{/mbargo}`
	the af and bf parameters can be used together to specify that text should appears between two dates.
	`{mbrgo af='2022-05-31' bf='2022-06-03'}This content only displays between 5/31 12:00am and 6/2 11:59pm{/mbrgo}`

`{author_bio slug='david-watts'}`	Inserts the author biography for a given author. **Tip**: the slug parameter is unique to the author. You can copy the code by going to the Author page for the author.
`{tour_qr form_id=lp1234 size=120}`	Inserts the QR code for the Augmented Reality (AR) Viewer. If the macro is run in the 3D Tour document, then no need to include the form_id (not case sensitive), otherwise specify the document number of the 3D Tour where the AR viewer is included. Default size of the QR code is 245px square; use the size parameter to override.

Smarty code

The use of if/then Smarty code is very powerful and a real time saver, especially in conjunction with shared library blocks. You can display content based on the `$system_name_short` variable, for example, which means that a library block can be shared across multiple documents, even if that content needs to change slightly from one document to another.

Smarty help: https://www.smarty.net/docsv2/en/language.function.if.tpl

Hint: Wrap the smarty code in HTML comments `<!-- ... -->` to ensure the smarty code doesn't get interpreted as HTML. This is especially important in tables.

If statements

```
<!--{if in_array($system_name_short, ['ST50 V2','ST250 V2','SR250 V2']) }-->
Display something only for these three servers.
<!--{else}-->
Display something for all other servers.
<!--{/if}-->
```

For the above, you can add `not` after the if to to the logical opposition of the boolean expression.

Use `<!--{elseif}-->` to add additional if clauses.

Logical and / or

Use `&&` for **logical and** and `||` for **logical or**. Also: **and** and **or** work too.

isset - checks if a variable is set

Use along with an and to only do something if the variable is set and matches a certain value:

```
{if isset($no_emr_support) and in_array($system_name_short, $emr_support)}
Do something if variable no_emr_support is set and system_name_short is in th
e array
{/if}
```

String operations

Use **strstr** to do a needle/haystack check. The code below checks if variable $system_name_short contains the string "V3":

```
<!-- {if $system_name_short|strstr:"V3"} -->
Display something only if the server name has "V3" in it.
<!--{/if}-->
```

Use **substr** to extract characters from a given position in a string. The first number is the starting position (0 = character 1) and the 2nd number is the length of the string to be returned. Note that "4" as the starting position means the character in the 5th position.

```
{if $system_name_short|substr:4:1 eq "5"}
This an AMD server, since the 1 character in the 5th position is a 5 (1st cha
racter = 0)
{/if}
```

To get the first n characters from a string, use **truncate**. Eg to get the first 5 characters:

```
{$system_name_short|truncate:5}
```

Array variable

Use the following code to assign a set of strings to an array variable:

Note: variables are case sensitive

```
{assign var=feature_BFYA value=['SR635','SR655','SR645','SR665'] scope="root"
}
...
{if in_array($system_name_short, $feature_BFYA)}
```

Combining array variables

Use the **array_merge** function to combine array variables:

```
{assign var=tatlow value=['SR250 V2', 'ST250 V2', 'ST50 V2']}
{assign var=catlow value=['SR250 V3', 'ST250 V3', 'ST50 V3']}
{assign var=all_servers value=array_merge($tatlow, $catlow)}
```

2D and 3D array variables

Use the following to build an array of data.

Note: These assign vars need to be in HTML comments otherwise the => will get escaped to &gt;

Lookup with single result:

```
{assign var=tour_doc value=['SE350 V2'=>'LP1751','SE360 V2'=>'LP1752']}
...
<p>The tour_doc is {$tour_doc[$system_name_short]}
```

Lookup with multiple data output:

```
{assign var=shipping_dimensions value=[
'ST50 V2'=>['depth'=>420,'width'=>280,'height'=>540],
'SR250 V2'=>['depth'=>860,'width'=>610,'height'=>196]
]}
...
<p>Width: {$shipping_dimensions[$system_name_short].width} mm
```

Fancybox

These two parameters are added to the <a> tag to enable Fancybox, our lightbox image viewer widget.

```
<a data-fancybox="all" data-caption=""></a>
```

Once you add this code to the anchor tag wrapped around an image tag, when the user clicks the image in the document, the image will display centered on the page with the rest of the browser window greyed out (like a photographer's lightbox). Only applies to web pages; Ignored by the PDF generator.

Example:

```
<a data-caption="Lenovo ThinkSystem {$system_name_short}" data-fancybox="all"
href="https://lenovopress.lenovo.com/assets/images/LP1607/ThinkSystem-SR645-V
3.jpg"><img alt="Lenovo ThinkSystem {$system_name_short}" src="https://lenovo
press.lenovo.com/assets/images/LP1607/ThinkSystem-SR645-V3.jpg" style="width:
600px;height:116px;" /></a>
```

Output:

For other Fancybox hints, see https://fancyapps.com/docs/ui/fancybox/

Bootstrap alerts

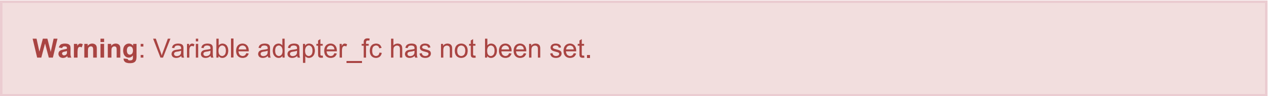
To display an alert banner use the predefined classes, `alert` and `alert-xxxxxx`.

- blue = `alert-primary`
- grey = `alert-secondary`
- green = `alert-success`
- red = `alert-danger`
- yellow = `alert-warning`

For example:

```
<p>{if not isset($adapter_fc)}</p>
<div class="alert alert-danger">
<p><strong>Warning</strong>: Variable adapter_fc has not been set.</p>
</div>
<p>{/if}</p>
```

This produces the following:



Warning: Variable adapter_fc has not been set.

Anchor links within the text should be coded with `class="alert-link"` so that they are colored properly.

For more information, see:
https://getbootstrap.com/docs/4.0/components/alerts/

Hiding email addresses

Use the mailto smarty function to obfuscate the email address from spammers:

```
{mailto address="example@example.com" encode="javascript_charcode"}
```

However, the use of javascript won't work in exports to Word. If you need the email address to appear in the Word export then change the encoding to hex:

```
{mailto address="example@example.com" encode="hex"}
```

See this page for other parameters:
https://www.smarty.net/docs/en/language.function.mailto.tpl

E-commerce URL management

Use the CMS to activate the blue **Configure and Buy** buttons in product guides and other documents. The process is two high-level steps:

1. Define the URLs for each product, one per e-commerce enabled market. This is done in the **E-COMMERCE URLS** section of the CMS
2. In the product guide, select the product in the **Enable e-commerce** field in the Document Info tab of the document in the CMS.

Note: Only CMS users with the **Buylink Administrator** admin role as specified in **Admin Users** section of the CMS will have the authority to define URLs in step 1 above.

In Step 1: To define the URLs for multiple products, the easiest way is to import them in an XLS file. You can use the Export function to get a complete listing of all URLs - you can use that as a template to add additional products.

Tips:

- The XLS file you use as an import does not have to be a complete list of all products. Only the products listed in column A are updated; all other products are ignored.
- If you import a product without any URLs, that product is deleted
- Once you click the button to Import, wait on that screen for 5-10 seconds. You should wait until a Success or Failure popup message appears.
- You can also delete a single product by opening that product up in the **Products** section and clicking the **Delete** button
- If you are updating URLs to an existing product via an XLS import, make sure the name of the product listed in column A matches the existing product. If not, a second product will be created (which will probably be confusing and not recommended)
- To add additional markets, simply add additional columns.

In Step 2: To select the products you want associated with a product guide, simply enter the product name(s) in the **Enable e-commerce** field in the Document Info tab. You can enter more than one product.

Useful web sites

These are web tools I use regularly:

- HTML-G - An easy way to strip out formatting and extra CSS from web pages or weird formatting from Word documents. Will also correctly process Tracked Changes from a word document. Requires a license for longer text.
- tblr - Converts Excel to HTML table code.
- CharacterCount - will display the number of characters and words in text you type/paste in.
- PDF unlocker - This tool will remove the protections from a PDF, such as preventing the ability to copy text.
- &what; ("amp-what") - a tool to show you all possible symbols & emojis you can use in text. Good for symbols like µ (also &micro;) , and for footnote symbols like daggers († and ‡, &dagger; and &Dagger;) and section (§ or &sect;), and of course the poop emoji

Useful Browser extensions

Firefox:

- Clippings: Lets you keep a library of code snippets for easy pasting into a field. Works in Source mode or in Normal mode (hold the Ctrl key down then right-click). https://addons.mozilla.org/en-US/firefox/addon/clippings/
- FoxReplace: Global search and replace in Source mode. https://addons.mozilla.org/en-US/firefox/addon/foxreplace

Chrome:

- tblr: Create HTML table code from Excel. https://chrome.google.com/webstore/detail/tblr/hfpbmgjmknhcakmgmfofmjloiecbocjj
- Copytables: Allows you to select and copy cells of a table from a web page. https://chrome.google.com/webstore/detail/copytables/ekdpkppgmlalfkphpibadldikjimijon

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION ”AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document, TIPS4CMS, was created or updated on April 30, 2024.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/TIPS4CMS
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/TIPS4CMS.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®
AnyBay®
ThinkSystem®

The following terms are trademarks of other companies:

AMD is a trademark of Advanced Micro Devices, Inc.

Excel® and Windows® are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.