displayas( ) — Set display level

Description

displayas(level) sets whether and how subsequent output is to be displayed.

Syntax

void displayas(string scalar level)

where level may be

<table>
<thead>
<tr>
<th>level</th>
<th>Minimum abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;result&quot;</td>
<td>&quot;res&quot;</td>
</tr>
<tr>
<td>&quot;text&quot;</td>
<td>&quot;txt&quot;</td>
</tr>
<tr>
<td>&quot;error&quot;</td>
<td>&quot;err&quot;</td>
</tr>
<tr>
<td>&quot;input&quot;</td>
<td>&quot;inp&quot;</td>
</tr>
</tbody>
</table>

Remarks and examples

If this function is never invoked, then the output level is result. Say that Mata was invoked in such a way that all output except error messages is being suppressed (for example, quietly was coded in front of the mata command or in front of the ado-file that called your Mata function). If output is being suppressed, then Mata output is being suppressed, including any output created by your program. Say that you reach a point in your program where you wish to output an error message. You coded

printf("{err:you made a mistake}
")

Even though you coded the SMCL directive {err:}, the error message will still be suppressed. SMCL determines how something is rendered, not whether it is rendered. What you need to code is

displayas("err")
printf("{err:you made a mistake}
")

Actually, you could code

displayas("err")
printf("you made a mistake
")
because, in addition to setting the output level (telling Stata that all subsequent output is of the specified level), it also sets the current SMCL rendering to what is appropriate for that kind of output. Hence, if you coded

```
    displayas("err")
    printf("{res:you made a mistake}\n")
```

the text you made a mistake would appear in the style of results despite any quietlys attempting to suppress output. Coding the above is considered bad style.

### Conformability

```
    displayas(level):
    level: 1 × 1
    result: void
```

### Diagnostics

`displayas(level)` aborts with error if `level` contains an inappropriate string.

### Also see

[M-5] `display()` — Display text interpreting SMCL

[M-5] `printf()` — Format output

[M-4] `IO` — I/O functions