

---

Subject: Re: Losing \$where when fixed selection is not present

Posted by [AJM](#) on Mon, 17 Aug 2009 17:15:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

You like being awkward, don't you!

Attached is a new version of std.list2.inc in which I have cut out the overwriting of \$where with \$selection. Both of these values will be passed to initialise() and therefore \_cm\_initialise(). I have changed the logic inside initialise() as follows:

```
$where2 = $this->_cm_initialise($where, $selection, $search);
if ($where2 != $where) {
    // this was changed in _cm_initialise(), so use the new version
    $where = $where2;
} else {
    if (!empty($selection)) {
        // $selection takes precedence over $where
        $where = $selection;
        $selection = null;
    } // if
} // if
```

If \_cm\_initialise() changes the contents of \$where then \$selection will be ignored. If \$where is not changed and \$selection is not empty then \$where will be overwritten by \$selection.

Note that the argument list for \_cm\_initialise() has been changed to the following:

```
function _cm_initialise ($where, &$selection, $search)
// perform any initialisation for the current task.
// NOTE: $selection is passed by reference as it may be amended.
// NOTE: $search is only available for OUTPUT tasks.
{
    // customisable code goes here

    return $where;
} // _cm_initialise
```

\$selection is passed by reference so that you can change its contents. If you erase it then it cannot be used to overwrite the contents of \$where.

Try the attached file to see if it works OK.

## File Attachments

---

1) [std.list2.zip](#), downloaded 2541 times

---