About the SCSS Treasures Catalog

Dr.Brian Coghlan

The catalog entries can be sorted as you wish by selecting the sort keys with the drop-down boxes. The four sort keys are employed in order, so the left key is the major key and the right the minor key. For example if the left key selected *Literature Only* and other keys remained as per default then only literature would be displayed and in order of vintage, etc. Selecting *Name/Title[1-4]* sorts by the first four letters of the name, so e.g. all DEC hardware can be sorted together. The catalog can be searched using the standard browser Edit → Find menu.

The accession index is self-identifying and date-stamped, with trailing item number and sub-item number. The name/title is overloaded as the name of non-literature items or the title of literature items, as are the synopsis of non-literature items and author of literature items. If the accession index is highlighted (in blue) as a link then clicking on it will open a new tab showing the contents of the folder associated with the catalog item. If the name is highlighted as a link then clicking on it will open a new tab with a PDF description of the catalog item.

The catalog has been designed to survive for decades as a valuable historical resource, to continue to function without attention and present a very low burden to catalog curators, system administrators and server machinery. It is designed to:

- ***be maintenance-free*** by avoiding software that will evolve through new versions that might not be fully backwards-compatible with previous versions. It achieves this by using only basic Javascript and HTML that should be around for a very long time. The importance of this to a persistent historical resource cannot be overstated.
- ***minimize the burden on school server machines*** by making the remote browsers do most of the work, not the school servers. This is achieved by using Javascript, and allows access to scale up substantially without unduly impacting server performance.
- ***function as a generic repository of information in folders*** associated with catalog items. This is achieved by holding information in files (descriptions, manuals, datasheets, brochures, whatever) within a filesystem structure that directly maps onto the simple hierarchy of basic URLs and uses only basic server actions widely supported by UNIX/Linux, NTFS and other filesystems.
- ***support public and private access*** via server-side mechanisms.
- ***accomodate access via asymmetric broadband links*** that have high download and low upload bandwidths. This is achieved by mainly using downloads, first downloading Javascript, which then only downloads URL targets.
- ***maximize robustness and responsiveness*** by minimizing the catalog webpage complexity using only a simple layout with no images and a minimal palette. The use of Javascript helps as most of the work is localised to the remote browser.
- ***present printer-friendly information*** by only using HTML or PDF, with descriptions of catalog items in self-identifying PDF files that for maximal compatibility derive from simple Word-2003 documents.
- ***index catalog items by simple text files*** that can easily be understood and edited by catalog curators. This is achieved by representing items as BibTex-style entries that describe each item by a number of key-value pairs that create a Javascript object.