

ZSpy Databus Debugger Tool

The framework provides a debug tool called ZSpy. ZSpy is instantiated by the ZLoader prior to loading any other applications. ZSpy runs in the same thread as ZLoader; this thread is not shared by any ZApplication.

ZSpy shows the current state of the databus, and provides information about the nodes on the databus. The developer can navigate to a node and check the publisher, list of subscribers, current node content and attributes. The databus tree is dynamic and changes are automatically reflected.

ZSpy maintains the most recently changed node and the node changed prior to it. The most recently changed node appears in bold font. Both nodes - most recently changed and changed before last - appear in bright red color. The parents of these nodes at all levels appear in darker red color.

At any time the tree can be manually refreshed with the refresh button.

The screenshot shows the ZSpy Databus Debugger Tool interface. The window title is "ZSpy". The interface is divided into several sections:

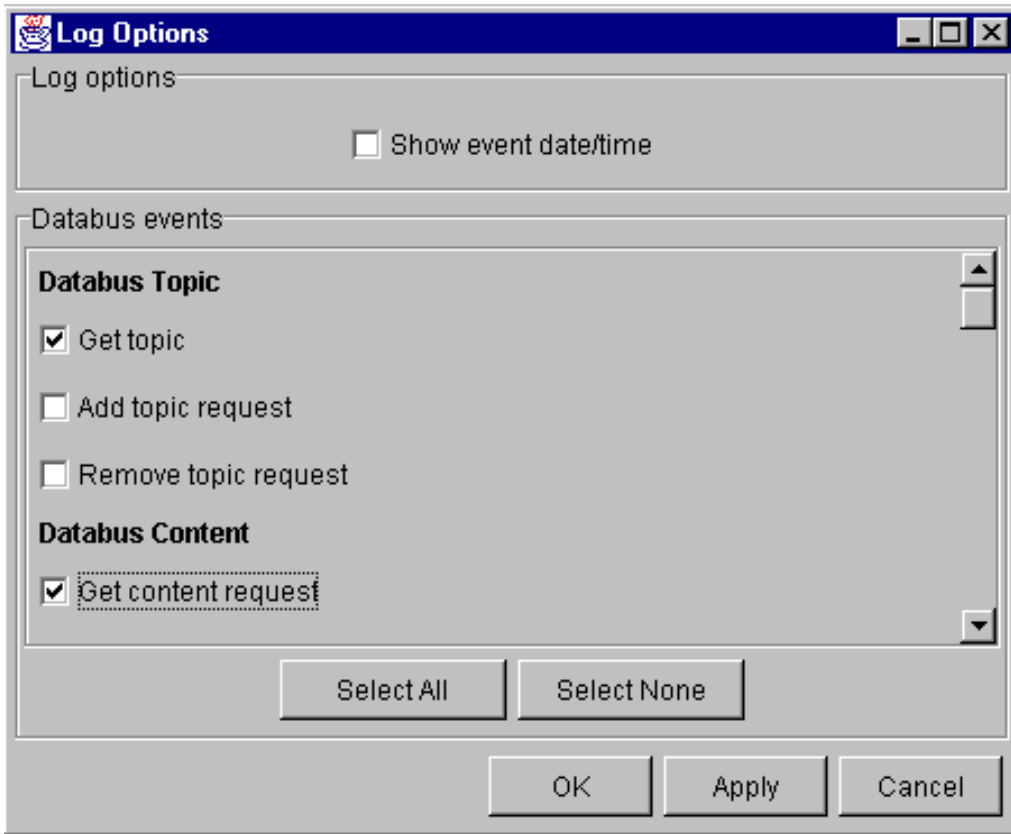
- Tree View (Left):** Displays a hierarchical tree of databus nodes. The root node is "%universal_root%". It has children: "PREFERENCES", "%RefData_root%", "%Search_root%", and "__RETAIL_BUS". "%Search_root%" is expanded, showing "PREFERENCES", "SEARCH", and "CRITERIA". "SEARCH" is expanded, showing "RESULTS", "TOTALCOUNT", "CSEGMENTCOUNT", "___CFETCHEDCOUNT___", "C_LAST_KEY_CUST_ID", "C_LAST_KEY_SEARCH_NM", and "C_MORE_INDICATOR". The "___CFETCHEDCOUNT___" node is highlighted in blue.
- Refresh Button (Bottom Left):** A button labeled "Refresh".
- Details Panel (Right):** Displays information for the selected node. It includes:
 - Path:** "//%Search_root%/SEARCH/RESULTS/___CFETCHEDCOUNT___"
 - Value:** "0"
 - Attributes:** "*** none ***"
 - Publisher:** "com.schwab.retail.zapp.document.ZElementPublisher"
 - Subscribers:** "com.schwab.retail.zapp.view.ZLabel"
- Log Window (Bottom Left):** Shows the text "Signing on to SEntry" and "Authenticated by SEntry".
- Options Button (Bottom Right):** A button labeled "Options".

Logging

Along with the databus tree, ZSpy also displays a log window. Applications can write messages to the log at any time. The messages can be pre-defined and assigned a category, or can just be a plain text. A Date/time stamp is automatically added

to every message (this can be turned off from the Options window).

The Options window allows the enabling and disabling of various messages pre-defined by the framework. For convenience, messages are divided into several categories:



ZSpy provides several static functions to be used by applications: log, notify, etc. Refer to the ZSpy JavaDoc for full explanation.

A constant is defined in ZConst class called `_debug`. If it is set to false, the ZSpy window will not come up, and all calls to ZSpy functions will be ignored.