



Shared Element Management Interface Control Document

Shared elements are generated by executables that require their usage, and by programs that are the sources for their content. A general purpose shared element management tool is needed to provide ad-hoc control over their implementation. Shared elements consist of two components, variables and pipes.

Shared variables are containers of single discrete pieces of information that are used in a setting where the most recent information is required. A shared variable has no history. When its value is read the latest state is delivered and when it is written to its current state is overwritten. Shared variables (SVs) are intended to have one source and many uses. There are no restrictions to the number of users or sources for each shared variable as they are accessed at will. The design of a shared variable system helps assure that a shared variable is written to by appropriate sources and used by appropriate applications. Shared variables are limited longs (32 bits), doubles (32 bits) or strings (any length).

Shared Variable Pool

Shared pipes provide a connection from one application to another. The pipe can contain any type of information or structure. A pipe has a single receive end but may receive information from any number of sources. Each executable has a pipe that is named for the instance of that application. It is up to the receiving application to determine the type of information.

Shared Element Manager

Shared Variables

The commands and controls listed in Table 1 are executed on the Shared Variables tab of the Shared Element Management program (see Figure 1).

Table 1: djSVManagement's Shared Variables Commands & Controls

Command/Control	Functions and Usage
Selected Entry	An entry in the shared variable list is selected when it is clicked on and the blue highlight bar moves to that selection. The parameters of the shared variable are then placed in entry fields.
New	Clear the entry fields.
Create	Create a new shared variable from data entered into the entry



Kairos Autonomi
498 W. 8360 S.
Sandy, Utah 84070
801-255-2950 (office)
801-907-7870 (fax)
www.kairosautonomi.com

BULLETIN
BUL-046

	fields.
Del	Delete the selected shared variable from list and shared variable pool.
Read	Perform a single read of the selected shared variable and update its value in the list from the shared variable pool.
Write	Perform a single write of the selected shared variable from the value entry field to the shared variable pool.
+ & -	Increment and decrement the value of the selected shared variable both in the list and in the shared variable pool. This is a read/modify/write operation.
0	Place a 0 in the value and shared variable pool for the selected shared variable.
Save	Write to a file all of the shared variables and their parameters for later usage.
Load	Read from a file the parameters of shared variables. Add the shared variables to the list and create them in the shared variable pool if they do not exist.
Delete All	Delete all shared variables from the list and from the shared variable pool. The programs exit after this deletion.
Refresh List	Read the values of all shared variables in the list and update the values in the list.
Update SVs	Repopulate the list with all shared variables from the shared variable pool.
Auto Load SVs	When the SVMangement program is first loaded, the last accessed file is loaded into the shared variable list.
Continuous Read of SVs	Reads of all shared variables are performed at the INTERVAL defined by the user.

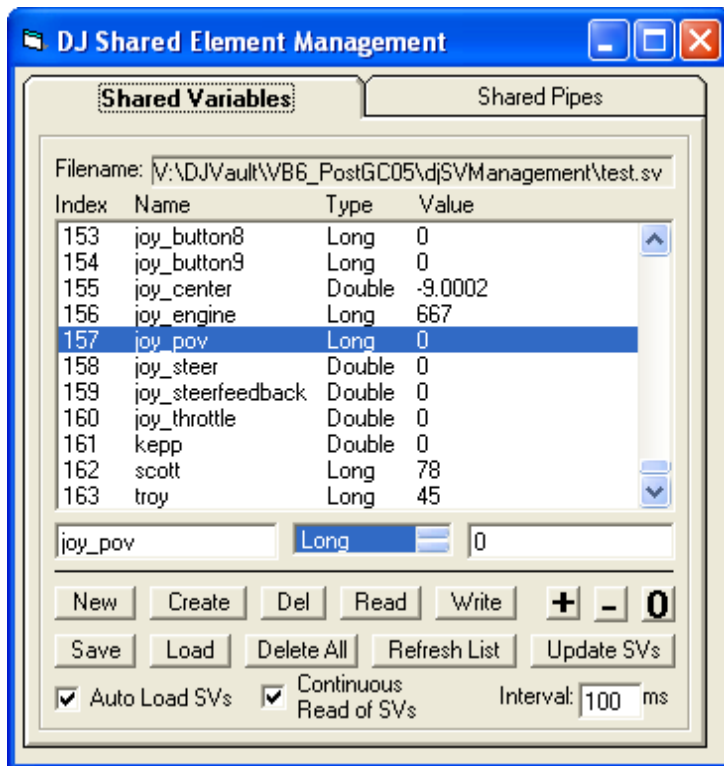


Figure 1: djSVMManagement – Shared Variables tab

Once a shared variable has been created its type cannot be changed. The variable must be deleted and then recreated. A shared variable is only created after it has its first value written to it. It is possible for a program to assign a shared variable for read only usage and the variable not be created until it is written to. When a shared variable has been assigned by a program, but not created and then read, the read values are 0 for numbers and the empty string for strings.

Shared variables may be created outside of the SVMManagement program by other programs while it is running. The user must manually update the displayed list of shared variables with the appropriate command button or that list is updated upon program start.

Shared Pipes

The commands and controls in Table 2 are executed on the Shared Pipes tab (see Figure 2) of the Shared Element Management program.



Kairos Autonomi
498 W. 8360 S.
Sandy, Utah 84070
801-255-2950 (office)
801-907-7870 (fax)
www.kairosautonomi.com

BULLETIN
BUL-046

Table 2: djSVManagement's Shared Variables Commands & Controls

Command/ Control	Functions and Usage
Selected Entry	An entry in the pipe accesses list is selected when it is clicked on and the blue highlight bar moves to that selection. The parameters of the pipe access are then placed in entry fields.
New	Create a new entry using the supplied data.
Save	Save the created pipe accesses to a file.
Load	Load pipe access from a selected file.
Update	Update the selected pipe access from the information in the entry fields.
Add	Create and add a new pipe access to the pipe access list.
Delete	Remove a pipe access from the pipe access list.
Pipes	Load all available pipes from the system into the pipe selection combo box.
Up	Move the selected pipe access up in the list.
Down	Move the selected pipe access down in the list.

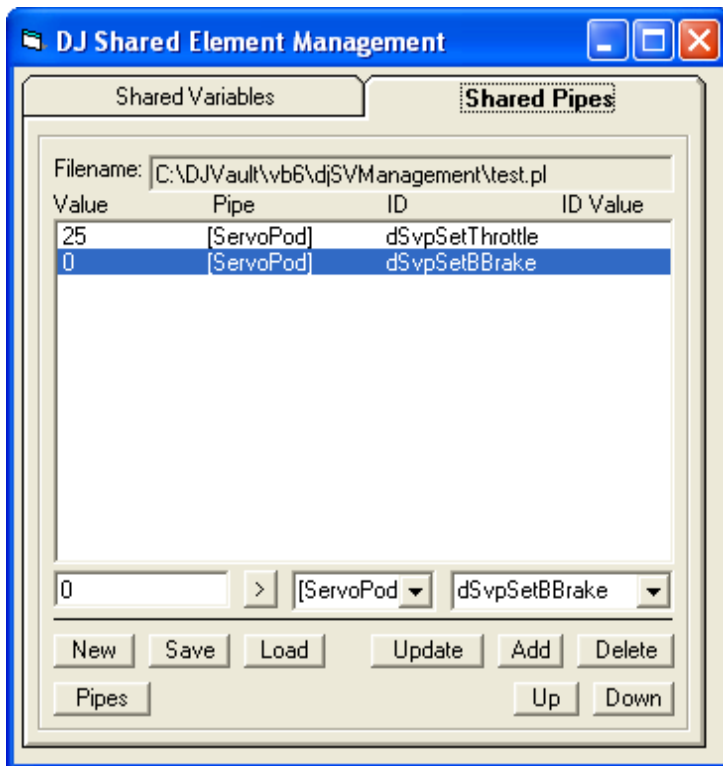


Figure A-2: djSVManagement - Shared Pipes tab

Shared pipes can only be created by an application. The SVMManagement program provides the ability to identify available pipes and place information in the pipe.

Shared pipes are primarily used by automated driving systems and are not usually used in drive-by-wire systems.

Parameters entered into the SVMManagement program are automatically saved to an INI file upon program exit and restored upon program execution.