Documentation of the tie examples

=================================

metakit.tcl

This is the implementation of a data source storing the array

in a metakit database.

server.tcl

sending\_client.tcl

receiving\_client.tcl

These three scripts belong together. They demonstrate how to

sharing an array across processes. It uses the package "comm"

and the data source "remotearray".

server.tcl

is invoked without arguments. It will print the id of

the TCP server port it is listening on. It has a

single array 'server'. Changes to the array are

reported on stdout.

sending\_client.tcl

is invoked with the id of the server as its only

argument. It has a local array 'sender'. Changes to

'sender' are exported to the server vie tie,

remotearray, and comm. The changes made are hardwired

into the script and executed with a delay of 1/10th of

a second between them, after a 2 second startup delay.

receiving\_client.tcl

is invoked with the id of the server as its only

argument. It has a local array 'receiver'. Changes to

receiver are reported to stdout. The script imports

the server array, and any changes on the server are

mirrored in the receiver.

Open three xterm and start the three scripts in them, in the

order

server.tcl

receiving\_client.tcl

sending\_client.tcl

Two seconds after the sending client has started both server

and receiver start to report the changes made by the sender to

its array and broadcast to server and then the receiver.

transceiver.tcl

A combination of both sending\_client.tcl and

receiving\_client.tcl. Exports the local array to the server,

and imports the server array to the local one. Performs

changes both local and on the server, showing that both

changes get distributed to both partners, independent where

the change was made.