```
-- Variable setup
set samp_file_path to "/Users/jdsant/.samp"
set samp_secret to ""
set samp_hub_xmlrpc_url to ""
set samp_client_key to ""

-- Read the contents of the .samp file
set theFile to POSIX file samp_file_path
open for access theFile
set sampFileContent to (read theFile)
close access theFile

-- Use an AppleScript trick to get the different sampLines
set prevTIDs to AppleScript's text item delimiters
set myDelimiter to "\n"
set AppleScript's text item delimiters to myDelimiter
set sampLines to text items of sampFileContent
set AppleScript's text item delimiters to prevTIDs

-- For each line, look if the key name corresponds with either
-- samp.secret or samp.hub.xmlrpc.url
repeat with sampLine in sampLines
    if sampLine contains "=" then
        set AppleScript's text item delimiters to "="
        set itemsToParse to text items of sampLine
        set AppleScript's text item delimiters to prevTIDs
        if the first item of itemsToParse is equal to "samp.secret" then
            set samp_secret to the second item of itemsToParse
        end if
        if the first item of itemsToParse is equal to "samp.hub.xmlrpc.url" then
            set samp_hub_xmlrpc_url to the second item of itemsToParse
        end if
    end if
end repeat

set table_votable_load_subscribers to {}

-- Prepare client metadata to be sent to the hub
set client_metadata to {|samp.name|:"AS_SAMP", |samp.description.text|:"An
      AppleScript XML-RPC based SAMP client.", |samp.documentation.url|:"http://
      developer.apple.com/documentation/AppleScript/Conceptual/soapXMLRPC/",|
      samp.icon.url|:"file:///Users/jdsant/Documents/IAA/MaterialesTesis/SAMPy
      %20tests/SEScriptEditorX.png"} ¬


if samp_hub_xmlrpc_url is not equal to "" then
    tell application "http://192.168.2.1:56483/xmlrpc"
        -- AppleScript cannot use a variable to set the
        -- XML-RPC destination, we need to set it by hand

        -- Perform the registration call, passing the .samp file's
        -- samp.secret; register_map is the result, and contains
        -- the shared hub-application secret key
        set register_map to call xmlrpc {method name:"samp.hub.register", parameters:
                {samp_secret}}

        -- retrieve the private app-hub key
        set samp_client_key to |samp.private-key| of register_map

        -- declare metadata to the hub (nothing returned)
        call xmlrpc {method name:"samp.hub.declareMetadata", parameters:
                {samp_client_key, client_metadata}}

        -- obtain a list of registred clients
        set registered_clients_list to call xmlrpc {method
                name:"samp.hub.getRegisteredClients", parameters:
                {samp_client_key}}

        -- We do not need to loop through client to get those supporting an mtype
        -- However, this call would have returned the list of
        -- subscriber to the table.load.votable MType
        set table_votable_load_subscribers to call xmlrpc {method
                name:"samp.hub.getSubscribedClients", parameters:
                {samp_client_key, "table.load.votable"}}

        -- Prepare a message map to be broadcasted
        set messageMap to {|samp.mtype|:"table.load.votable", |samp.params|:{|
                url|:"file:///Users/jdsant/CVS/devel/astrogrid/desktop/api/examples/
                workflows/sample-conesearch-input_veron.vot", name:"ASS Veron
                Ceti", |table-id|:"AS_SAMP#1"}}

        -- Broadcast the prepared message map: the result is that
        -- all applications able to load the VOTable would read
        -- that file
        set notified_ids to call xmlrpc {method name:"samp.hub.notifyAll", parameters:
                {samp_client_key, messageMap}}

        repeat 10000000 times
            get notified_ids
        end repeat

        -- We do a VERY LONG loop in order to be able to see
        -- this script as a registered application
        -- Unregister with the hub, so that we not leave a zombie
        -- application
        call xmlrpc {method name:"samp.hub.unregister", parameters:
                {samp_client_key}}


        set all_results to {registered_clients_list, table_votable_load_subscribers,
                notified_ids}
        get all_results
    end tell
end if
```