

Bulk Change Owners of Structures and Generators

```
import com.atlassian.jira.ComponentManager
import com.atlassian.jira.user.ApplicationUsers
import org.apache.log4j.Category

def oldOwnerKey = 'admin'
def newOwnerKey = 'ragnar'
def resync = true // do full resync after update

def plugin = ComponentManager.getInstance().getPluginAccessor().getPlugin('com.almworks.jira.structure')
def loader = plugin.getClassLoader()
def StructureAuth = loader.loadClass('com.almworks.jira.structure.api.auth.StructureAuth')
def JiraUser = loader.loadClass('com.almworks.jira.structure.api.permissions.PermissionSubject$JiraUser')
def structureManager = plugin.getModuleDescriptor('structure-manager').getModule()
def syncManager = plugin.getModuleDescriptor('sync-manager').getModule()

def oldOwner = JiraUser.newInstance(oldOwnerKey)
def newOwner = JiraUser.newInstance(newOwnerKey)
def newOwnerUser = ApplicationUsers.byKey(newOwnerKey)
if (newOwnerUser == null) {
    def message = "Cannot find user by user key: $newOwnerKey"
    log.error(message)
    return message
}

// The actual work is done here

def changedStructures = []
def changedSynchronizers = []
def success = false
def exception = null

try {
    StructureAuth.sudo {
        structureManager.getAllStructures(null, true).each { st ->
            // Change owner
            if (st.owner == oldOwner) {
                st.owner = newOwner
                st.saveChanges()
                changedStructures << st
            }
            // Change owner of synchronizers installed for this structure
            syncManager.getInstalledSynchronizersForStructure(st.id).each { sync ->
                if (sync.userKey == oldOwnerKey) {
                    def enabled = syncManager.isAutosyncEnabled(sync.instanceId)
                    if (enabled) {
                        syncManager.setAutosyncEnabled(sync.instanceId, false)
                        syncManager.updateSynchronizer(sync.instanceId, sync.getParameters(), newOwnerUser)
                        if (resync) {
                            syncManager.resync(sync.instanceId, true, null)
                        } else {
                            syncManager.setAutosyncEnabled(sync.instanceId, true)
                        }
                    } else {
                        syncManager.updateSynchronizer(sync.instanceId, sync.getParameters(), newOwnerUser)
                    }
                    changedSynchronizers << sync
                }
            }
        }
    }
    success = true
} catch (Exception e) {
    log.warn("Failed to change owner from '$oldOwnerKey' to '$newOwnerKey'", e)
    exception = e
}

// Output message about changed structures and synchronizers to the log and to the console output
def msg = "Script to change owner from '$oldOwnerKey' for '$newOwnerKey' " +
    (success ? "finished successfully" : "failed (${exception} && exception.message)")) + "\n" +
    "Changed structures:\n" + changedStructures.collect({ "${it.id} ${it.name}" }).join("\n") + "\n" +
    "Changed synchronizers:\n" + changedSynchronizers.collect({ "${it.instanceId} (for structure ${it.structuredId})" }).join("\n")

log.warn(msg)
msg.replaceAll("\n", "<br>")
```