

```
/*
```

This script include when called from a write ACL script will only grant write access if the record is not part of an open update set, other than your own or if the last update was made in the Default update set (which is always open).

In a write ACL of for instance sys_script_include disable the "Admin Overrides" checkbox (mandatory for this whole check to work) and add the following code to the script field of the ACL:

```
answer = new UpdateSetChecker().checkAllowedWrite(current);
```

See attached screenshot of what happens if you try to edit an object which is part of an open update set.

This whole setup works best if you only start close update sets until you are finished with all the update sets for a release. Or you put the finished update sets in the status Ignore until you are ready to transport them to QA/PROD.

If you want you could even create a new status for the update set, like pending or ready. Anything different from the state Complete.

Below you can find the Script Include UpdateSetChecker with the function checkAllowedWrite.

```
*/
```

```
//Script Include: UpdateSetChecker
```

```
var UpdateSetChecker = Class.create();
UpdateSetChecker.prototype =
Object.extend(Object.prototype, {
  initialize: function() {

  },

  checkAllowedWrite: function(current) {
    //gs.log("*****
UpdateSetChecker.UpdateSetChecker: START");
    var my_update_set =
gs.getPreference('sys_update_set','empty')
    var find_name = current.sys_class_name+ '_' +
current.sys_id;
    //gs.log('*****
UpdateSetChecker.UpdateSetChecker: ' + find_name);
    var write_allowed = false;

    var version = new GlideRecord('sys_update_version');
    version.setLimit(1);
    version.addQuery('name',find_name);
    version.addQuery('state', 'current');
    version.query();
```

```

        if (version.next()) {
            var updateSetGR = new
GlideRecord('sys_update_set');
            if (!updateSetGR.get(version.source)) {
                write_allowed = true;
            }
            if (version.source.state != 'complete') {
                if (version.source.sys_id ==
my_update_set) {
                    write_allowed = true;
                }
            }
            if (version.source.state == 'complete') {
                write_allowed = true;
            }
            if(version.source.name == 'Default'){
                write_allowed = true;
            }
        } else {
            write_allowed = true;
        }

        if (!write_allowed && version.source.name!
='Default') {
            gs.addInfoMessage("This record is part of
the open update set " + version.source.name + " which is not your
current update set and therefor can not be updated.");
        }

        return write_allowed;
    },
    type: 'UpdateSetChecker'
});

```