

Simulated Railroad Framework, <http://simulrr.sourceforge.net>
Synopsis: [000_Synopsis](#)

This file valid for step 0033.10
Issue Date: 2017-03-17

The Contained Key Lock
=====

1 Synopsis

The "Contained Key Lock" is a MIDAS Object, that is provided together with the SMUOS Framework.

The Contained Key Lock needs the "key manager" extension of the Simple Scene Controller, see chapter "Use Case CarriedKeysSupport" in [121_SimpleSceneController](#).

The "Contained Key Lock" is implemented within two X3D prototypes, MoosLockB and MoosLockBNs within the files MoosLockB.x3d and MoosLockBNs.x3d, respectively.

2 Purpose of the Contained Key Lock

The "Contained Key Lock" can be unlocked by a contained key.

The model/module author specifies the "fitting keys" and if one of these keys is contained in the lock, then the lock will be unlocked globally in all scene instances.

Similar to the "Key Container" MIDAS Object, the key can be stored, created and deleted as well as displayed.

The "contained key" is modelled by an SFString value, where following rules apply:

- only one key can be put into the Contained Key Lock
- if there is already a key in the lock, no key can be put into the lock
- if you try to remove a list of keys from the lock, then the first key that is equal the contained key, will be removed

3 External View

The MIDAS Object "Contained Key Lock" can be used in

- bound/intrinsic models in static modules
- bound/intrinsic models in dynamic modules
- unbound models (not yet tested)

Following fields are provided at the external interface uiObj:

Standard Fields

Please refer to chapter 5 of the paper [013_ModelsAndObjects](#) for a description of fields that must be supported by any MIDAS Object.

"addKeys" (MFString)

Add a list of keys to the contained key lock. In one scene instance, send an MFString value to the contained key lock and the first key of the list will be added to the contained key lock globally, if the lock is empty ("containedKey" will be updated in all scene instances).

"removeKeys" (MFString)

Remove a list of keys from the contained key lock. In one scene instance, send an MFString value to the contained key lock and the list of keys will be removed from the contained key lock globally ("containedKey" will be updated in all scene instances). Only the first key, that is equal the contained key, will be removed.

"takeKeys" (MFString)

Remove a list of keys from the contained key lock. In one scene instance, send an MFString value to the contained key lock and the list of keys will be removed from the contained key lock globally ("containedKey" will be updated in all scene instances). Only the first key, that is equal the contained key, will be removed.

Additionally, the taken key will be added to the "carriedKeys" (output of the Simple Scene Controller at the uiControl interface), locally in this scene instance.

"set_bind" (SFBool)

Make the contained key lock the actual bound key container in this scene instance.

After set_bind=true, all "putKeys" events to the Simple Scene Controller (via the uiControl interface) will lead to moving the keys from the "carried keys" (locally in this scene instance) to the "containedKey" of this contained key lock (globally for all scene instances).

"initialKey" (SFString)

Use this field to initialize the state of the contained key lock.

"containedKey" (SFString)

After each change of the global state, this field will output the actual key contained in the contained key lock.

"fittingKeys" (MFString)

With this field, the model/module author can specify a set of keys, each of which will unlock the lock.

"locked" (SFBool)

At this field, the lock will report its state.

4 Internal View

MoosLockB uses MibStandard as a base. A network sensor MoosLockBNs has been implemented to send the diverse requests to the object controller (OBCO) and to reply to a specific scene instance, when "carriedKeys" are to be changed. A special field of the network sensor is used to distribute the "reset key counter". The OBCO will maintain the "global state" and distribute it to all scene instances.

5 Additional Info

none