

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

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

The Key Container  
=====

## 1 Synopsis

-----

The "Key Container" is a MIDAS Object that is provided together with the SMUOS Framework.

The Key Container needs the "key manager" extension of the Simple Scene Controller, see the chapter "Use Case CarriedKeysSupport" in [121\\_SimpleSceneController](#). The "Key Container" is implemented within two X3D prototypes, MoosKeyContainer and MoosKeyContainerNs within the files MoosKeyContainer.x3d and MoosKeyContainerNs.x3d, respectively.

## 2 Purpose of the Key Container

-----

The "Key Container" MIDAS Object helps to provide interactivity in all situations, where "SrrTrains/SMUOS keys" have to be stored, created and deleted as well as displayed.

The "contained keys" are modelled by an MFString value, where following rules apply:

- each key is identified by an unstructured SFString value
- key identifiers may exist more than once (several identical keys)
- when a key is deleted from a key container, then the first occurrence of the string is deleted

## 3 External View

-----

The MIDAS Object "Key Container" 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 key container. In one scene instance, send an MFString value to the key container and the list of keys will be added to the key container globally ("containedKeys" will be updated in all scene instances).

### "removeKeys" (MFString)

-----

Remove a list of keys from the key container. In one scene instance, send an MFString value to the key container and the list of keys will be removed from the key container globally ("containedKeys" will be updated in all scene instances).

"takeKeys" (MFString)

-----  
Remove a list of keys from the key container. In one scene instance, send an MFString value to the key container and the list of keys will be removed from the key container globally ("containedKeys" will be updated in all scene instances).

Additionally, the list of keys 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 key container 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 "containedKeys" of this key container (globally for all scene instances).

"initialKeys" (MFString)

-----  
Use this field to initialize the state of the key container.

"containedKeys" (MFString)

-----  
After each change of the global state, this field will output the actual list of keys contained in the key container.

#### 4 Internal View

-----  
MoosKeyContainer uses MibStandard as a base. A network sensor MoosKeyContainerNs 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