

Simulated Railroad Framework, <http://simulrr.sourceforge.net>

Synopsis: [000_Synopsis](#)

This file valid for step 0033.11

Issue Date: tbd.

Console Interface

=====

1 Synopsis

It's one of the ideas of the SrrTrains concept to provide access to the railway simulation via a simple command line interface.

Parameters and/or variables of the simulation should be read out and influenced via a simple command line.

The whole simulation is embedded into the scene (the SRR/SMUOS Framework doesn't use the external EAI/SAI) and it is distributed to independent modules and models.

The only central access point is the Simple Scene Controller and its user interface uiControl.

Hence, the user interface uiControl provides an input field "consoleCommand" (SFString), which can be used to induce a command into the simulation.

A second field, the output field "consoleResponse" (MFString) will deliver the response from the simulation.

The actual user interface, which is used to provide the console to the user, is up to the frame author (could be a GUI realized in an external application / web page or could be a HUD realized in VRML/X3D or something else).

It's up to the frame author, to which scene instances (users) he grants the ability to use the console interface. In multi-user-mode the SRR/SMUOS Framework will distribute the console commands and their resulting changes to all scene instances of the simulation.

2 Modules, UOCs, MIDAS Objects and Parameters

MIDAS Objects are always assigned to modules or to UOCs, so that each MIDAS Object of the scene can be identified by an "extended object ID",

```
<extObjId> = Bdo.<moduleName>--<objId>
or
<extObjId> = Uoc.<uocName>--<objId>
.
```

A <moduleName> is used in the <extObjId>, if the object is a bound object (which is bound to a module), otherwise a <uocName> is used, if the object is an unbound object or an astral object (for an explanation of the terms "bound", "unbound" and "astral" please refer to the paper [013_ModelsAndObjects](#)).

The programmer of a MIDAS Object can decide to support the console interface, i.e. to provide parameters that can be influenced or read out by the console interface.

Additionally, the SSC Base provides and each SSC Extension, that supports one or more UOCs, can decide to provide so-called SSC Parameters.

SSC Parameters are similar to parameters of MIDAS Objects in they can be accessed via the console interface.

SSC Parameters are identified by the special objId "#parm", hence using the
<specialExtObjId> = Uoc.<uocName>--#parm
at the console interface.

The SMUOS Framework will deliver the console command from the uiControl interface to some instance of the MIDAS object or to some instance of the SSC extension and it will route the answer back to the uiControl interface. It's up to the programmer of the MIDAS Object or of the SSC Extension to process the console command and to change the simulation accordingly.

Parameter values are transported from the uiControl interface to the MIDAS object or to the SSC Extension and back again as string values, it's up to the programmer of the MIDAS Object or of the SSC Extension to interpret and create the strings correctly.

The console interface basically allows read and write access to all parameters, it's up to the programmer of the MIDAS Object or of the SSC Extension to restrict access accordingly (e.g. for read-only variables or parameters).

The "SMS Dispatcher Stub" (prototype SmsDispatcherStub in the file SmsDispatcherStub.x3d) contains the common services, that are necessary to support the console interface for MIDAS Objects and SSC Parameters.

In the case of MIDAS Objects, the SMS Dispatcher Stub is instantiated automatically as a part of the MIDAS Base (MIB) and in the case of SSC Parameters it has to be instantiated by the programmer of the SSC Extension.

3 The Command Line Syntax

The Simple Scene Controller accepts one console command at a time. It will ignore further console commands (within the current scene instance), until the console response will have been output via the uiControl interface.

A console command is an SFString value of one of the following formats:

```
?
help
options[:<optAddress>]
read:<address>
set:<address>[=[<value>]]
```

where [] denotes an optional part and the address parameters can attain one of the following alternatives:

```
<optAddress> = <dispatcherNames>[-<objId>[-<parameterNames>]]
<address> = <dispatcherNames>-<objId>[-<parameterNames>]
```

The names are built according to the following rules:

```
<dispatcherNames> = Mod.<moduleName> || Uoc.<uocName> || '*'
<parameterNames> = <parameterName> || '*'
```

When the <objId> is #parm, then only one parameter name is allowed:

```
<objId>=#parm ==> <parameterNames> = <parameterName>
```

The console response will be created as an MFString value, according to the given console command. Errors will be indicated by the string prefix '***'.

The console interface sends one and only one console response to each and every console command, unless the console is already active (waiting for a response). In the latter case, the console command will be silently discarded.

3.1 Some Examples

'options' will output an MFString with one dispatcher name in each SFString. All registered modules/UOCs will be reported (UOCs first).

'options:Bdo.Hill' will output all <extObjIds> of all bound MIDAS Objects, which are announced in the module 'Hill' (MIDAS Objects that don't provide a console interface, will not announce themselves at the SSC Dispatcher).

'options:Bdo.Hill-Carousel.Switch' will output all <extObjId>-<parameterName>s of all parameters of the carousel switch in module 'Hill' (the 'Switch' object is contained within the 'Carousel' object).

'set:Bdo.Hill-Carousel.Switch-toggle' will toggle the carousel switch (no matter if it is locked or not).

4 How to Implement a Console Interface for a New MIDAS Object / SSC Extension

The "base class" of MIDAS Objects (see [301_MidasObjects](#)) provides following fields to the author of a MIDAS object:

- availableParameterNames
- parameterNames
- valueToSet
- defaultValueFlag
- sessionId
- set
- read
- options
- response

The SMS Dispatcher Stub, which is used by SSC Extensions to provide one or more UOCs, provides the same fields to the author of an SSC Extension. Additionally, the SMS Dispatcher Stub provides the field

- availableParameterLocalDelivery

With the field "availableParameterNames", you have to tell the SMUOS Framework, which parameters are supported by this MIDAS object / UOC.

The field "availableParameterLocalDelivery" indicates for each available parameter name, whether the SSC Dispatcher should consider this SSC Parameter a local SSC Parameter or a global one.

Receiving an SFBool event at the field "set" means, that all parameters defined by "parameterNames" shall be set to the value "valueToSet". If "defaultValueFlag" is true, the parameters shall be set to their default values. A response shall be sent to the field "response".

Receiving an SFBool event at the field "read" means, that the values of all parameters defined by "parameterNames" shall be read out and reported at the MFString field "response".

Receiving an SFBool event at the field "options" means, that the options of all parameters defined by "parameterNames" shall be reported at the MFString field "response".

The response shall contain exactly as many strings as the field "parameterNames" contains (report the empty string '', if you do not know one of the parameters).

The options of a parameter will be given to the user as a hint, which kind of value is expected in a "set" command.

"sessionId" indicates the scene instance that has sent the console command.

5 Additional Info

5.1 Possible Console Responses

5.1.1 Overall Response

Either

- one module name / UOC name per line
- or
- answers of all addressed modules / UOCs (one after the other - see "per module/UOC")
- or
- *** SSC not activated
- or
- *** Syntax Error
- or
- no answer at all (console is already active)

5.1.2 Response per Module/UOC

Either

- <dispatcherName>-<objId> of all objects of the module/UOC
- or
- answers of the addressed object (see "per object")
- or
- <dispatcherName>-***Inactive
- or
- <dispatcherName>-***Timeout

5.1.3 Answer per Object

Either

- <dispatcherName>-<objId>-<parameterName> of all parameters of the object
- or
- answers of all addressed parameters (one after the other - see "per parameter")
- or
- <dispatcherName>-<objId>-***NotFound

5.1.4 Answer per Parameter

Either

- <dispatcherName>-<objId>-<parameterName>=<userDefinedResponse>*)
- or
- <dispatcherName>-<objId>-<parameterName>=***NotFound

*) The "user defined response" is defined by the programmer of the MIDAS Object or of the SSC Extension, respectively.