

SICE 2011

RT-Middleware Tutorial

Date: 2011/9/13 10:00~16:30

Place: SICE 2011



RT-Middleware tutorial



10:00 - 10:45	Part 1: Introducing RT-Middleware
	Tetsuo Kotoku (AIST)
	An introduction to RT-Middleware, RT-Systems and RT-Components.
11:00 - 12:30	Part 2: Building RT-Systems using RT-Middleware
	Geoffrey Biggs (AIST)
	Hands-on practice using small samples to construct complete RT-Systems.
13:30 - 15:00	Part 3: Creating RT-Components
	Geoffrey Biggs (AIST)
	Hands-on practice creating RT-Components.
15:15 - 16:00	Part 4: Human interaction with OpenHRI
	Yosuke Matsusaka (AIST)
	A demonstration of RT-Components for human-robot interaction.
16:00 - 16:30	Part 5: Discussion

Part 2: Building RT-Systems using RT-Middleware

Geoffrey Biggs (AIST)



OpenRT Platform

■ Software platform for next-generation robots

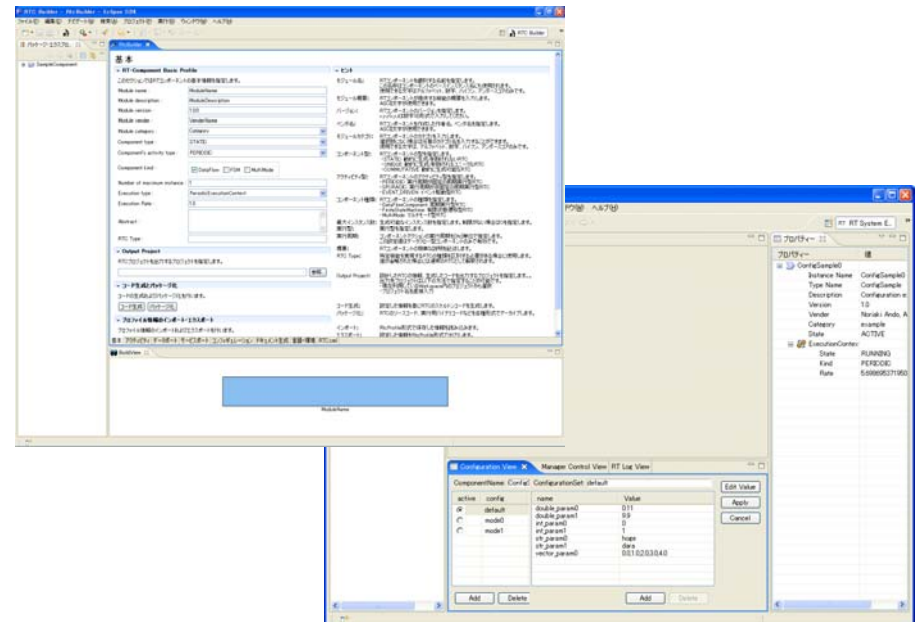
- <http://www.openrtp.jp/wiki/>
- System construction, simulation, motion generation, scenario creation, etc.

■ OpenRT Platform tools

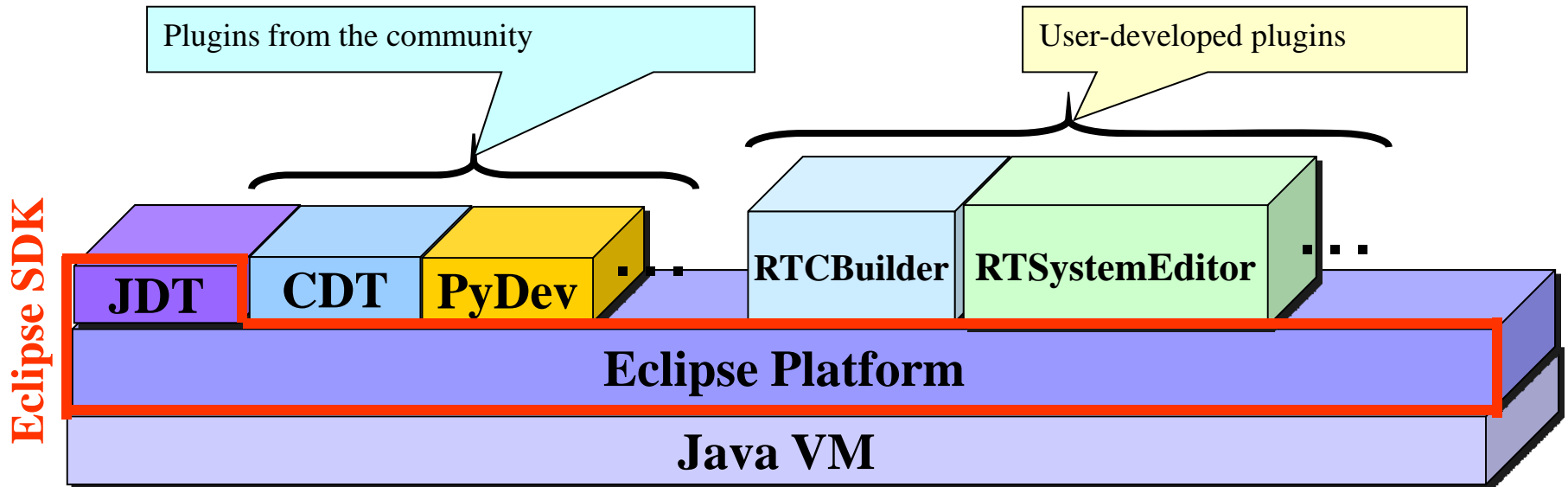
- Support for all phases of development.
- Uses Eclipse for the IDE platform.

■ Included tools

- **RTCBuilder**
- **RTCDebugger**
- **RTSystemEditor**
- Robot modelling tool
- Simulator (OpenHRP)
- Motion generation tool
- Scenario editor



- Open source, community-developed Integrated Development Environment
 - Multiplatform: Windows, Linux, ...
 - New tools are developed as plug-ins, allowing customisation
 - Tools can become standalone by utilising Rich Client Platform (RCP)



Installing RTCBuilder and RTSystemEditor

■ Just download and extract

- The Java run-time must be installed separately

The screenshot shows the OpenRTM-aist website. The main content area is titled "OpenRTM Eclipse tools 1.0-RELEASE". It includes a "Table of contents" section with links to "全部入りパッケージ" (Full package), "バイナリ" (Binaries), "RTSystemEditor/RTCBuilderデベロップメント" (Development), "Eclipse/JDK/JRE等" (Eclipse/JDK/JRE etc.), and "過去のバージョン" (Previous versions). Below this is a section for "全部入りパッケージ" (Full package) which contains a table of download links for various operating systems and architectures.

Eclipse-3.4.2 [Ganymede SR2]		
Eclipse3.4.2+RTSE+RTCB Windows用全部入り	eclipse342_rtmttools100release_win32_ja.zip MD5:A52450B24F0A1C59402D5340D9FA8D56	2010.06.01
Eclipse3.4.2+RTSE+RTCB (英語版) Windows用全部入り	eclipse342_rtmttools100release_win32_en.zip MD5:2A1895F0E01D874E35CDD29EFDCE1DE7	2010.06.01
Eclipse3.4.2+RTSE+RTCB Linux用全部入り	eclipse342_rtmttools100release_linux_ja.tar.gz MD5:FD54B638BB72A351D92ACD22CD4099C7	2010.06.01
Eclipse3.4.2+RTSE+RTCB (英語版) Linux用全部入り	eclipse342_rtmttools100release_linux_en.tar.gz MD5:4B1F4ACEE7F8E99B9C36D08068DBE5E1	2010.06.01
Eclipse3.4.2+RTSE+RTCB MacOSX用全部入り	eclipse342_rtmttools100release_macosx_ja.tar.gz MD5:19277C8E1E672688347C6767B57D9D1F	2010.06.10

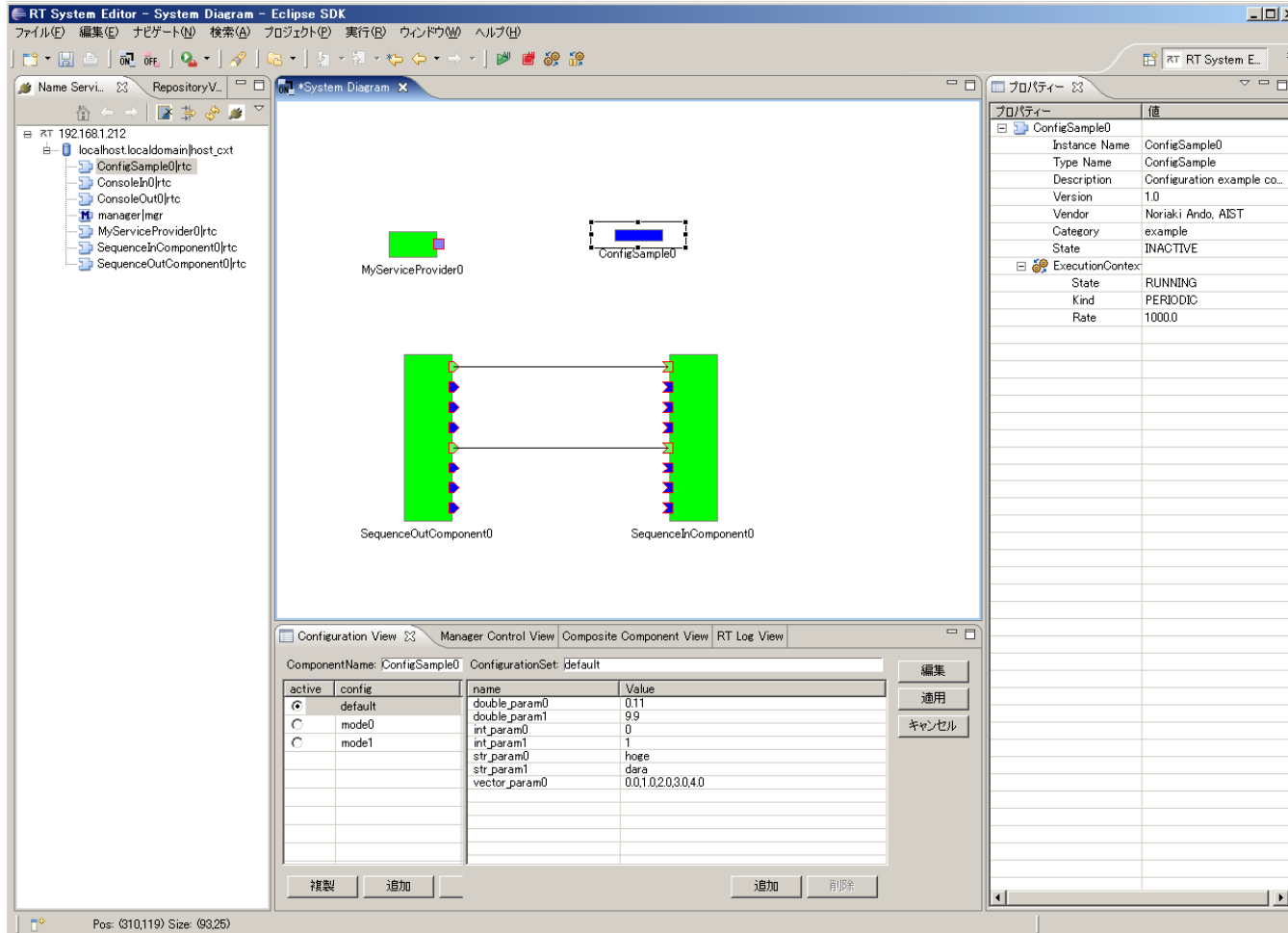
RTSystemEditor



RTSystemEditor outline

■ What is RTSystemEditor?

- A tool for combining RT-Components into complete systems



The screenshot displays the RTSystemEditor interface within the Eclipse SDK. The main window shows a system diagram with components: MyServiceProvider0, ConfigSample0, SequenceOutComponent0, and SequenceInComponent0. The left sidebar shows a project tree for 'localhost.localdomain/host_cxt'. The bottom panel shows the configuration for 'ConfigSample0'.

active	config	name	Value
<input checked="" type="radio"/>	default	double_param0	0.11
<input type="radio"/>	mode0	double_param1	9.9
<input type="radio"/>	mode1	int_param0	0
		int_param1	1
		str_param0	hoge
		str_param1	dara
		vector_param0	0.0,1.0,2.0,3.0,4.0

Properties panel (プロパティ) for ConfigSample0:

プロパティ	値
Instance Name	ConfigSample0
Type Name	ConfigSample
Description	Configuration example co...
Version	1.0
Vendor	Noriaki Ando, AIST
Category	example
State	INACTIVE
ExecutionContext	
State	RUNNING
Kind	PERIODIC
Rate	1000.0

Screen layout

The screenshot shows the RT System Editor interface. The main window displays a system diagram with components like MyServiceProvider0, ConfigSample0, and SequenceOutComponent0. A Name server view is overlaid on the left, showing a tree structure of components. A Property view is overlaid on the right, showing details for ConfigSample0. A Configuration view is overlaid at the bottom, showing a table of configuration parameters.

ComponentName	ConfigSet	Value
default	double_param0	011
mode0	double_param1	99
mode1	int_param0	0
mode1	int_param1	1
mode1	str_param0	hoge

Name server view

Property view

Configuration view

The Manager Control View shows a list of modules and components. Buttons for 'Create', 'Fork', and 'Shutdown' are visible.

Manager view

component	port
ConsoleIn0	ConsoleIn0.out
ConsoleOut0	ConsoleOut0.in

Composite component view

Execution Context	owned	participate0	rate
owned0	participate0	0	1000.2

Execution context view

component	time	level	component	logger	message
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!
Notify1	2011-04-28	ERROR	Notify1	RTC	test log!
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!
Notify1	2011-04-28	ERROR	Notify1	RTC	test log!
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!
Notify1	2011-04-28	ERROR	Notify1	RTC	test log!
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!
Notify1	2011-04-28	ERROR	Notify1	RTC	test log!

Log view

Preparation

- Start the Naming Service

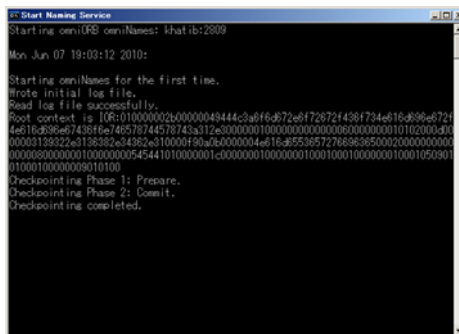
- [Start menu]→[All programs]→[OpenRTM-aist 1.1]→[C++]→[Tools]→[Start Naming Service]

- Start the ConsoleInComp

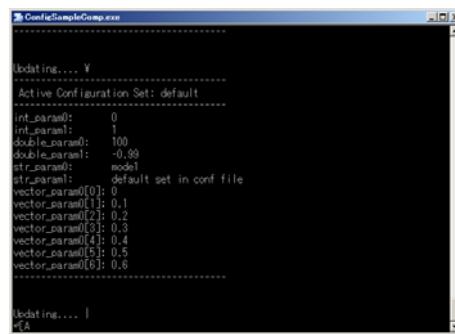
- [Start menu]→[All programs]→[OpenRTM-aist 1.1]→[C++]→[components]→[examples]→ [ConsoleInComp.exe]

- Start the ConsoleOutComp

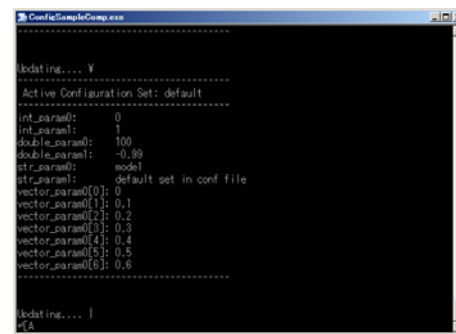
- [Start menu]→[All programs]→[OpenRTM-aist 1.1]→[C++]→[components]→[examples]→ [ConsoleOutComp.exe]



```
Start Naming Service
Starting omniORB omniNames: khatib:2309
Mon Jun 07 19:03:12 2010:
Starting omniNames for the first time.
Wrote initial log file.
Read log file successfully.
Root context is [0:010000002b00000049444c3a6f6e672e6f726724436f734e616d696e672f
4e616d696e67436f6e746578744578743a312e3000001000000000000000000010102000d00
000018322e3138232c34352d310000f90a7b5000004e816a955857276698385000200000000
00000000001000000054544101000001c000000100000010001000100000010001050901
010001000000000010100].
Checkpointing Phase 1: Prepare.
Checkpointing Phase 2: Commit.
Checkpointing completed.
```



```
ConfigSampleComp.exe
Updating... V
-----
Active Configuration Set: default
-----
int_param0: 0
int_param1: 1
double_param0: 100
double_param1: -0.89
str_param0: model
str_param1: default set in conf file
vector_param[0]: 0
vector_param[1]: 0.1
vector_param[2]: 0.2
vector_param[3]: 0.3
vector_param[4]: 0.4
vector_param[5]: 0.5
vector_param[6]: 0.6
-----
Updating... |
%A
```

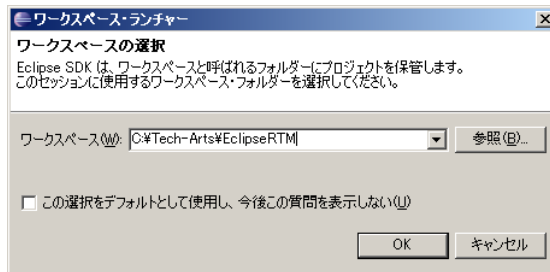


```
ConfigSampleComp.exe
Updating... V
-----
Active Configuration Set: default
-----
int_param0: 0
int_param1: 1
double_param0: 100
double_param1: -0.89
str_param0: model
str_param1: default set in conf file
vector_param[0]: 0
vector_param[1]: 0.1
vector_param[2]: 0.2
vector_param[3]: 0.3
vector_param[4]: 0.4
vector_param[5]: 0.5
vector_param[6]: 0.6
-----
Updating... |
%A
```

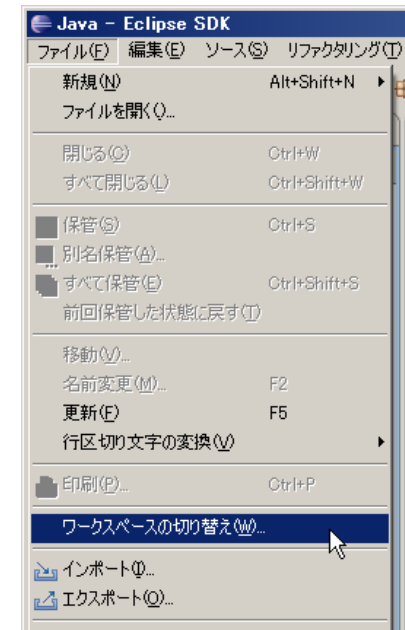
Launch the tool

- In Windows
 - Double-click on Eclipse.exe
- In Unix
 - Start it from the command line
 - e.g. `$ /usr/local/Eclipse/eclipse`

■ Select your workspace location



■ Changing the workspace:



❌ Workspaces

Projects and files created in Eclipse are saved in the directory specified as the Workspace.

Preparation

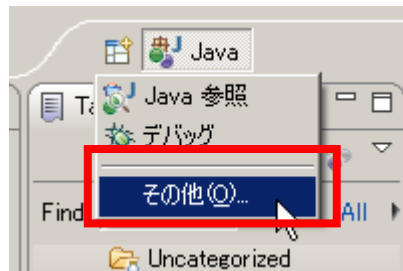
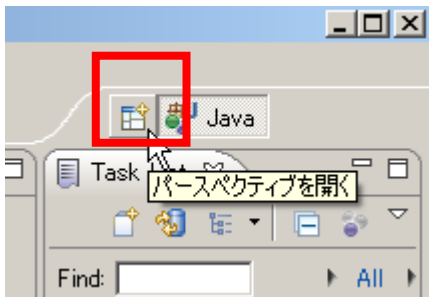
- Close the Welcome Screen
 - Only on the first time starting Eclipse



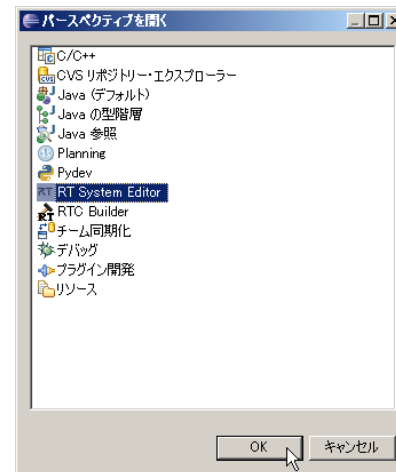
❌ Perspective
A tool in Eclipse.
Changes the menus, toolbars, editors, views, etc. to match the perspective's goals.

- Change the perspective

① Click the “Change perspective” button in the top right, and select “Other”

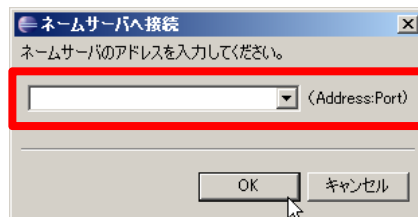
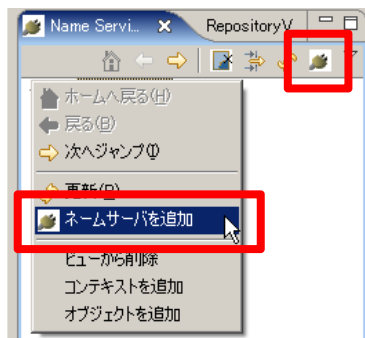


② Select “RTSystemEditor”

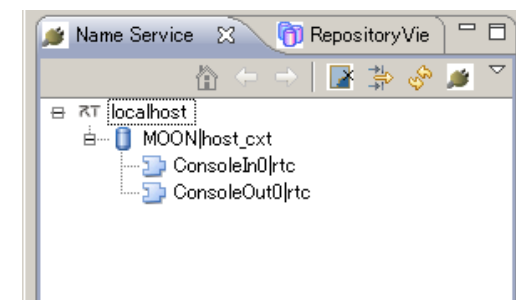


Basic operation of RTSystemEditor

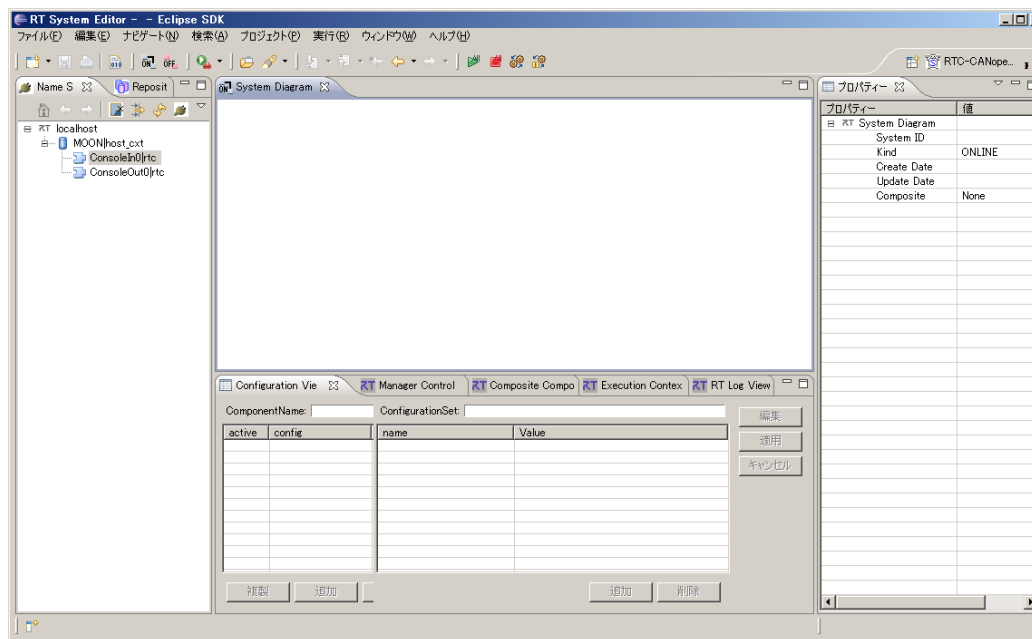
■ Connect to name servers



※Specify the name server's address and port. The default port can be changed in the settings screen.

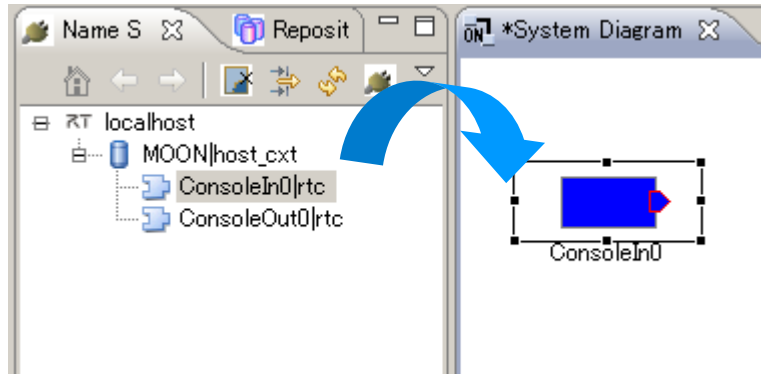


■ Open a new system editor

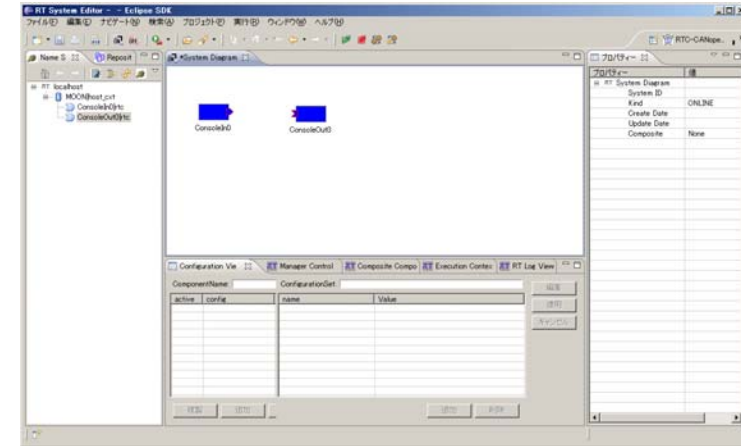


Creating a new RT-System

■ Placing RT-Components



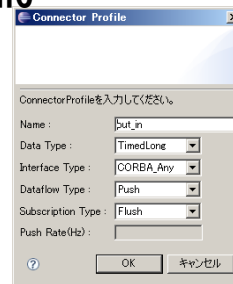
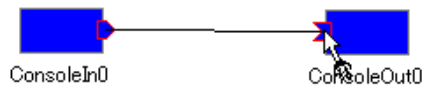
❌ Drag the target component from the name server view



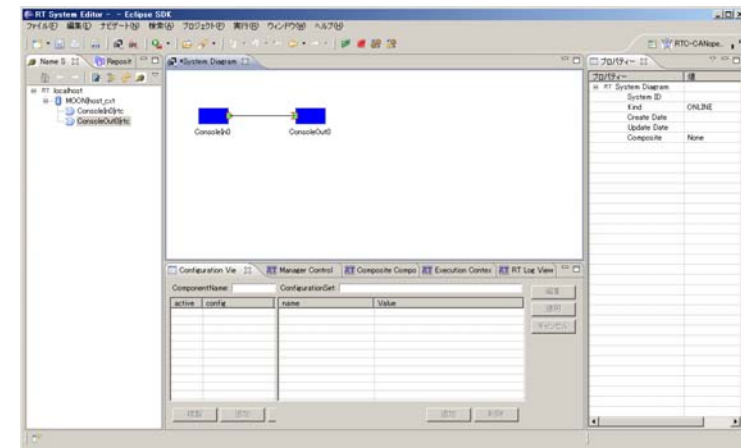
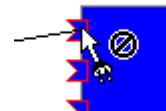
■ Connecting ports

① Drag from the source port to the target port

② Enter the connection profile



❌ The cursor changes to indicate incompatible ports, such as mis-matched port profiles.

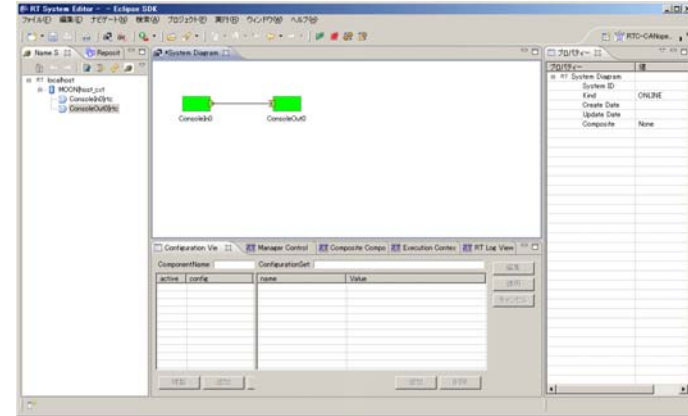
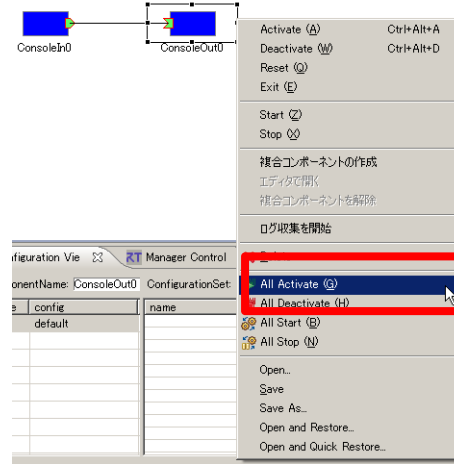
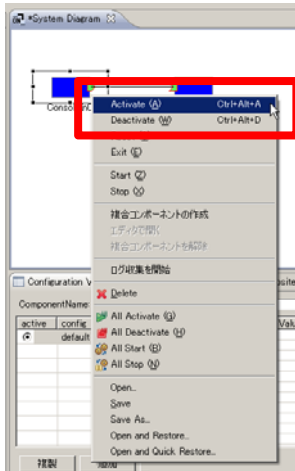


Creating a new RT-System

■ Activate the components

✖ Activating individually

✖ Activating all at once



■ Sample execution

① Enter numbers into ConsoleIn

② They will be displayed by ConsoleOut

```

ConsoleInComp.exe
dataport.interface_type: corba_cdr
-----
Connector Listener: ON_CONNECT
Profile:name: ConsoleIn0_out_ConsoleOut0.in
Profile:id: 2e25e2b7-cb49-43e1-8b0a-634f0a8dfc75
Profile:properties:
- data_type: IDL:RTC/TimeDLong:1.0
- interface_type: corba_cdr
- dataflow_type: push
- subscription_type: flush
- publisher
- push_policy: all
- serializer
- cdr
- endian: little.big
- corba_cdr
- inport_iior: IOR:010000001a0000004944c3a4f70656e52544d2f1496e506f72744364723a
812e300000001000000000000000064000000110102000d0000003139322e3136382e31312e390000
eb090e000000fec9eb194e0000152c0000000040000020000000000000000800000010000000094
5441010000001c000000010000000100001000000001000105090101000100000009010100
- consumer:
-----
Please input number: 128
    
```

```

ConsoleOutComp.exe
- dataflow_type: push
- subscription_type: flush
- publisher
- push_policy: all
- serializer
- cdr
- endian: little.big
- provider:
-----
Data Listener: ON_RECEIVED
Profile:name: ConsoleIn0_out_ConsoleOut0.in
Profile:id: 2e25e2b7-cb49-43e1-8b0a-634f0a8dfc75
Data: 128
-----
Data Listener: ON_BUFFER_WRITE
Profile:name: ConsoleIn0_out_ConsoleOut0.in
Profile:id: 2e25e2b7-cb49-43e1-8b0a-634f0a8dfc75
Data: 128
-----
Received: 128
TimeStamp: 0[s] 0[ns]
    
```

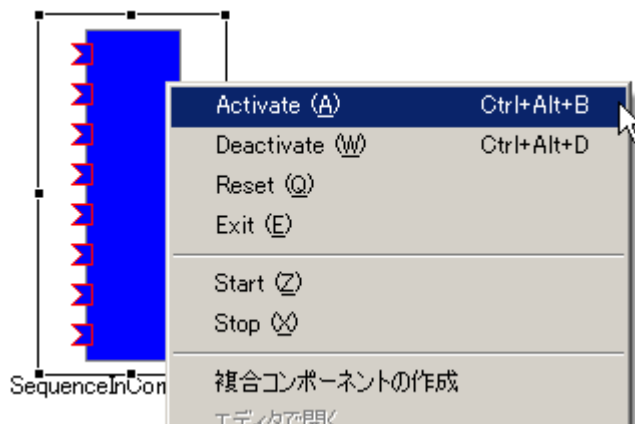
✖ To stop, use Deactivate

✖ To remove a connection, delete the connection line.

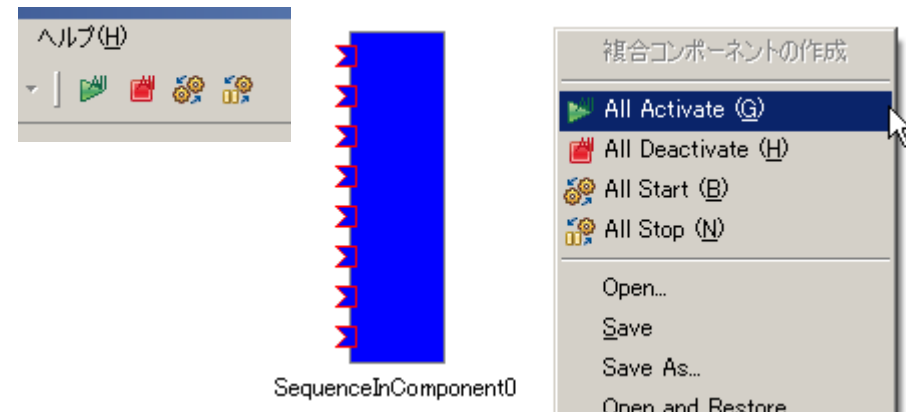
RT-Component control

Action	Explanation
Activate	Start RTC execution
Deactivate	Stop RTC execution
Reset	Reset the RTC from the Error state
Exit	Stop the RTC and cause it to exit
Start	Start the Execution Context
Stop	Stop the Execution Context

Individual component control



All components (entire RT-System) control



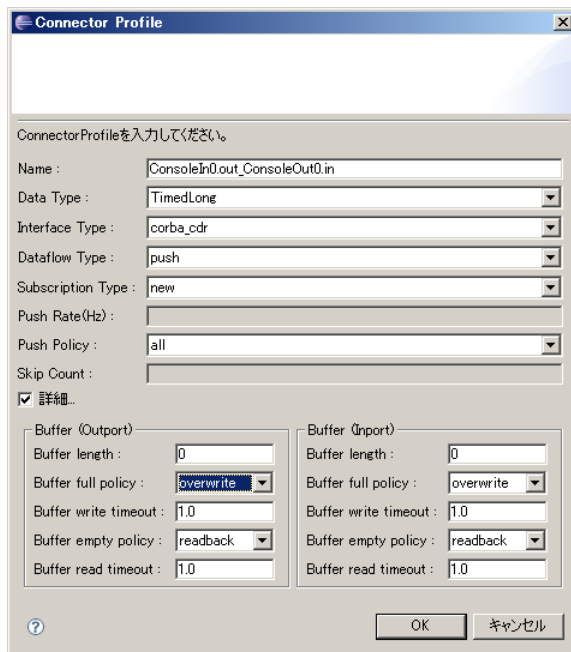
DataPort connector profiles

Item	Meaning
Name	Connector name
DataType	Transported data type (TimedOctet, LaserScan, etc.)
InterfaceType	Type of transport in use
DataFlowType	Type of data transmission (push, pull, etc.)
SubscriptionType	Data transmission timing when DataFlowType is Push .
Push Rate	Data rate in Hertz. Only used when SubscriptionType is Periodic .
Push Policy	Data transmission policy. Only used when SubscriptionType is New or Periodic .
Skip Count	Data skip count. Only used when Push Policy is Skip .

- SubscriptionType
 - New: Send when new data is written to the port's buffer.
 - Periodic: Send at a fixed rate.
 - Flush: Send immediately without buffering.
- Push Policy
 - all: Send all data.
 - fifo: Send data one at a time in FIFO order.
 - skip: Send intermittent data at the configured rate.
 - new: Send only new data, deleting older data in the buffer.

DataPort connector profiles

Item	Meaning
Buffer length	Length of the output port's buffer.
Buffer full policy	Action to take when the output port's buffer becomes full. Select from overwrite, block and do_nothing.
Buffer write timeout	Time in seconds before timeout when writing data (0 for none).
Buffer empty policy	Action to take when the input port's buffer is empty when reading. Choose from readback, do_nothing and block.
Buffer read timeout	Time in seconds before timeout when writing data (0 for none).



✘ The settings can be configured separately on the input and output ports.

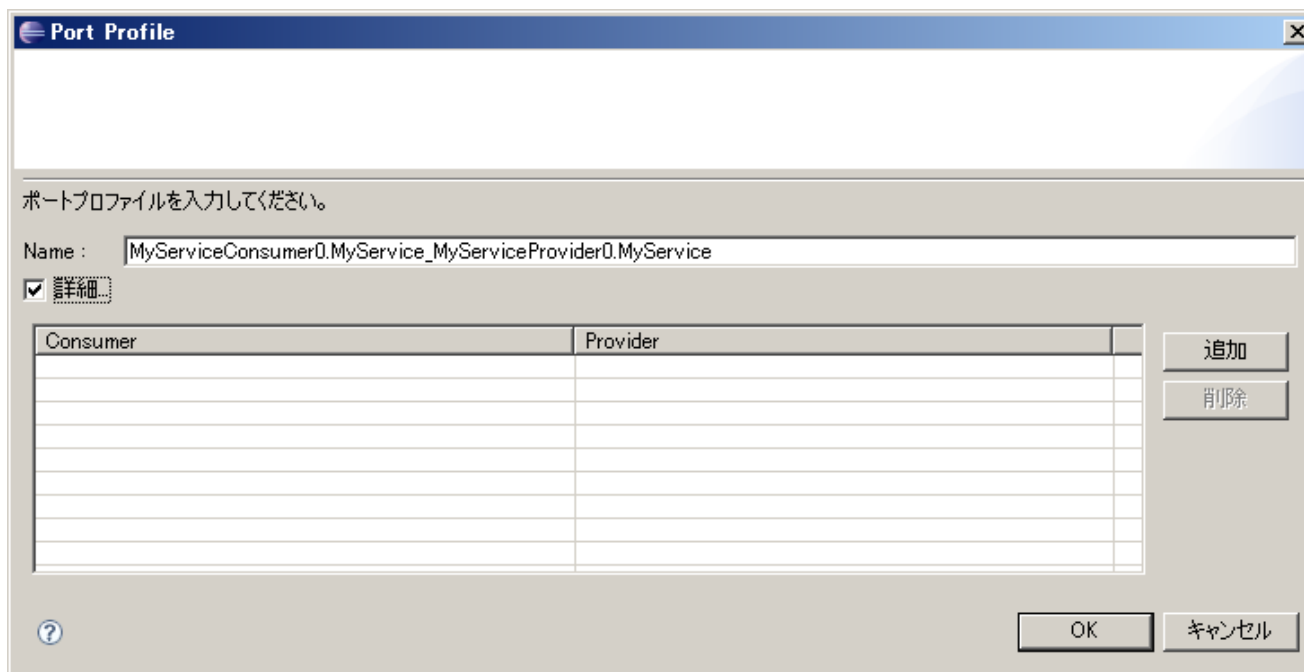
■ Buffer Policy

- overwrite
- readback: Re-read the most recent value
- block
- do_nothing

✘ When the buffer policy is block and a timeout time is set, the timeout event will occur after the timeout.

ServicePort connector profiles

Item	Meaning
Name	Connector name
Interface information	Specifies the interfaces to connect. A single ServicePort may offer or use many ServiceInterfaces. This specifies which to actually connect.



Port Profile

ポートプロファイルを入力してください。

Name : MyServiceConsumer0.MyService_MyServiceProvider0.MyService

詳細

Consumer	Provider

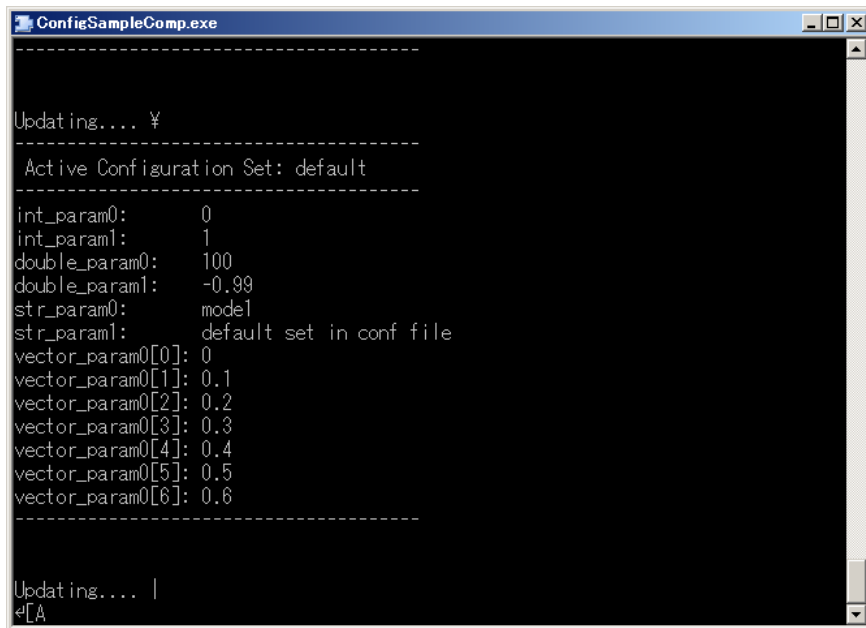
追加
削除

OK キャンセル

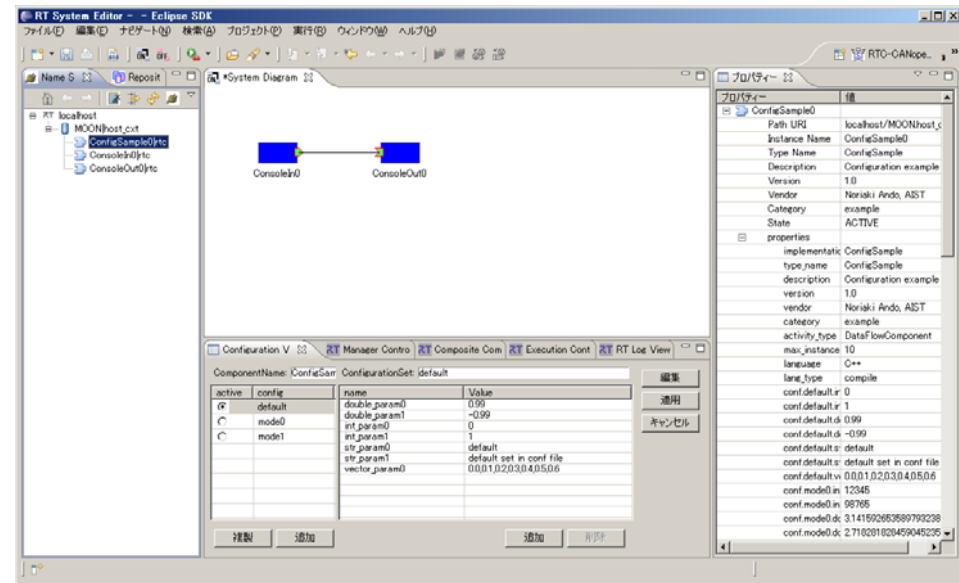
Preparation

Start the ConfigSampleComp

- [Start menu] → [All programs] → [OpenRTM-aist 1.1] → [C++] → [components] → [examples] → [ConfigSampleComp.exe]



```
ConfigSampleComp.exe
-----
Updating... ¥
-----
Active Configuration Set: default
-----
int_param0:      0
int_param1:      1
double_param0:   100
double_param1:   -0.99
str_param0:      model
str_param1:      default set in conf file
vector_param0[0]: 0
vector_param0[1]: 0.1
vector_param0[2]: 0.2
vector_param0[3]: 0.3
vector_param0[4]: 0.4
vector_param0[5]: 0.5
vector_param0[6]: 0.6
-----
Updating... |
^C
```

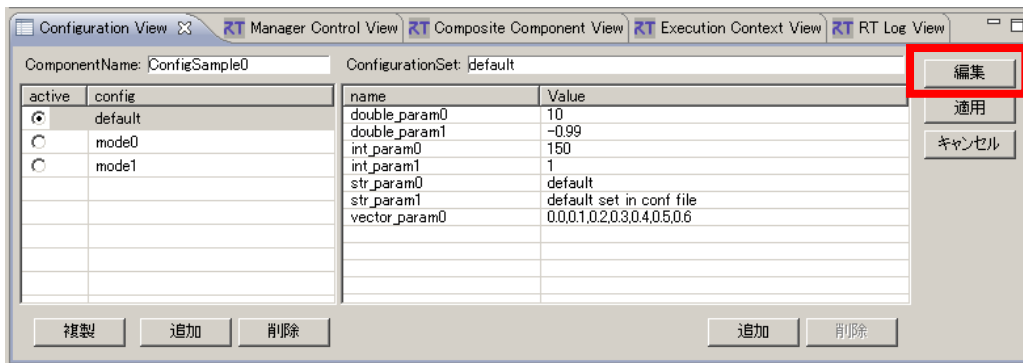


The screenshot shows the RT System Editor Eclipse SDK interface. The main window displays a system diagram with two components, ConsoleIn0 and ConsoleOut0, connected by a line. The right-hand side shows the 'Properties' view for the selected component, ConfigSample0, with various attributes like Path URI, Instance Name, Type Name, Description, Version, Vendor, Category, State, and properties. Below the properties view is a 'Configuration View' table.

ComponentName	ConfigSet	name	Value
active	default	double_param0	0.99
	default	double_param1	-0.99
	model0	int_param0	0
	model1	int_param1	1
		default	default set in conf file
		str_param1	default set in conf file
		vector_param0	0.001,0.2,0.3,0.4,0.5,0.6

Configuration view

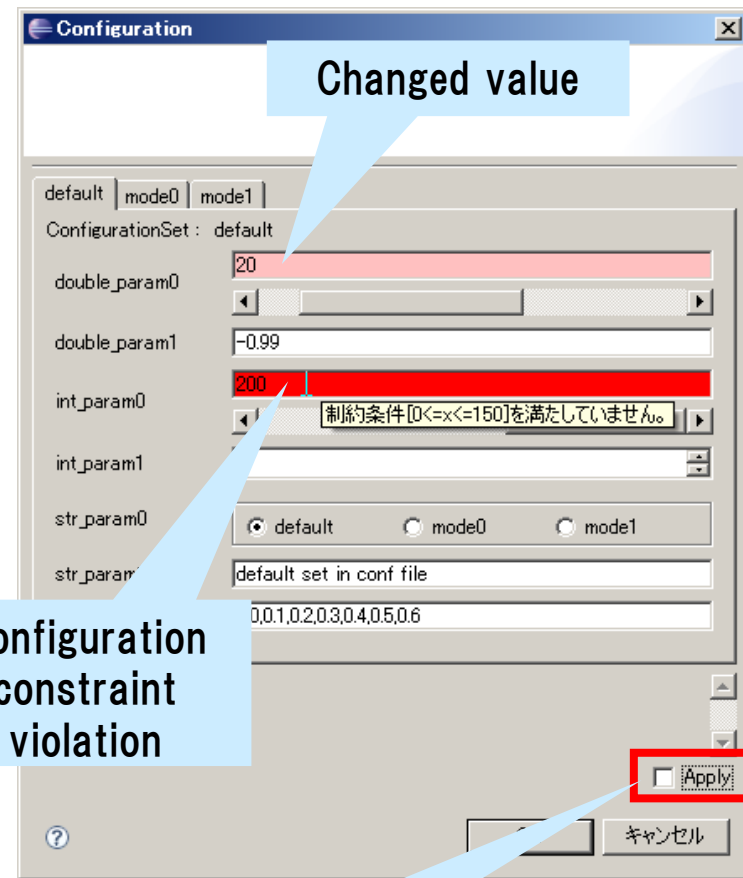
■ View and edit the configuration parameters of an RTC



✘ Use the “Edit” button to edit all values at once using a GUI

✘ If the “Apply” checkbox is on, edited values will be applied to the RTC immediately when changed.

✘ Configuration sets can be applied from the above tab.



Apply checkbox

Setting configuration parameters

- In `rtc.conf`

```
[Category].[Component name].config_file: [Configuration file name]
```

✘e.g. `example.ConfigSample.config_file: configsample.conf`

- In the configuration file

- Configuration parameter

```
conf.[Component name].[Parameter]: [Default value]
```

✘e.g. `conf.mode0.int_param0: 123`

- Widget information

```
conf.__widget__[Parameter] : [Widget type]
```

✘e.g. `conf.__widget__.str_param0: radio`

- Constraints information

```
conf.__constraints__[Parameter]: [Constraints]
```

✘e.g. `conf.__constraints__.str_param0: (bar,foo,foo,dara)`

```
conf.__[Component name].[Parameter]: [Constraints]
```

✘e.g. `conf.__mode1.str_param0: (bar2,foo2,dara2)`

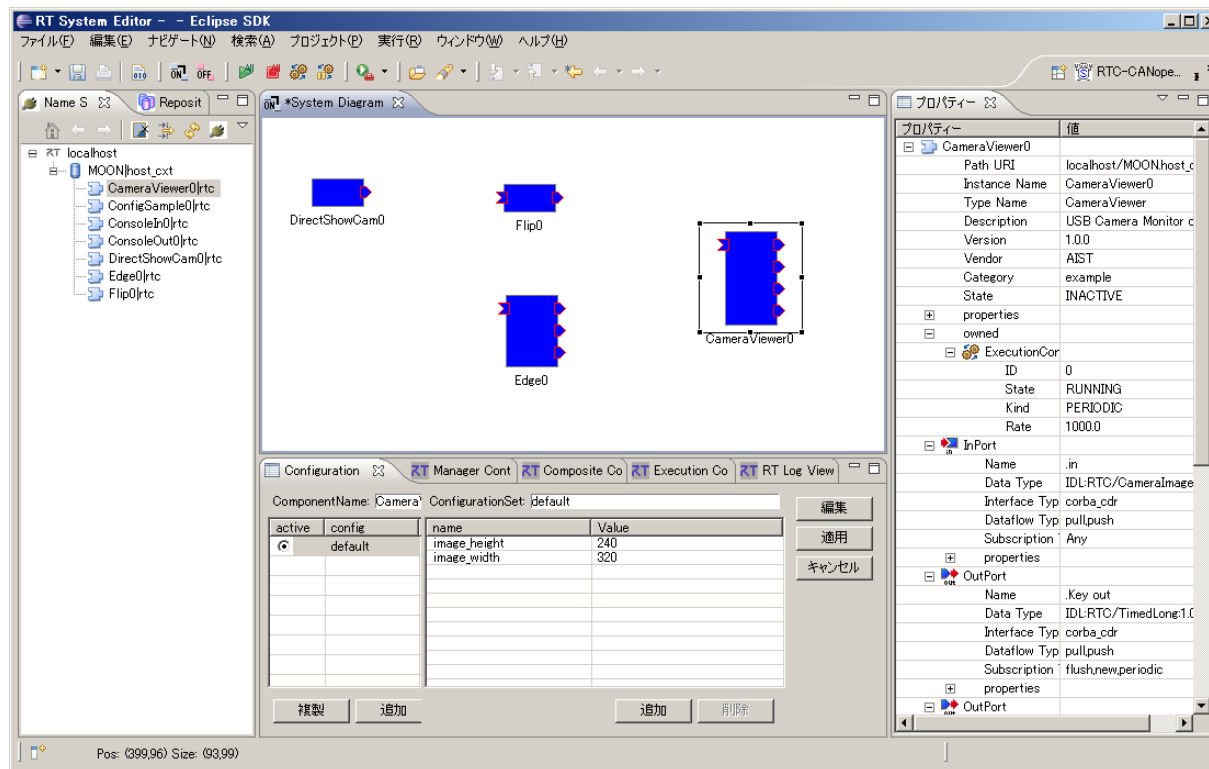
These should be set by the component developer, not the component user.

RTCBuilder can be used to set these values.

- Start the CameraViewerComp
 - [Start menu]→[All programs]→[OpenRTM-aist 1.1]→[C++]→[components]→[opencv_rtcs]→[**CameraViewerComp.exe**]
- Start the DirectShowCamComp
 - [Start menu]→[All programs]→[OpenRTM-aist 1.1]→[C++]→[components]→[opencv_rtcs]→[**DirectShowCamComp.exe**]
- Start some image processing components
 - [Start menu]→[All programs]→[OpenRTM-aist 1.1]→[C++]→[components]→[opencv_rtcs]→[**FlipComp.exe**]
 - [Start menu]→[All programs]→[OpenRTM-aist 1.1]→[C++]→[components]→[opencv_rtcs]→[**EdgeComp.exe**]

System construction

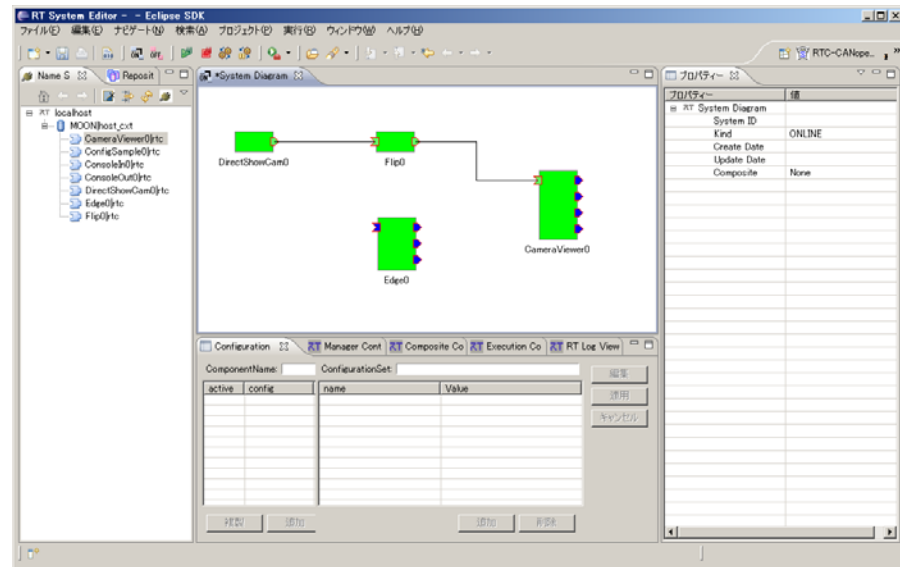
- Place the following components in the system editor
 - DirectShowCam
 - Flip
 - Edge
 - CameraViewer



Making the system

■ Connecting Flip

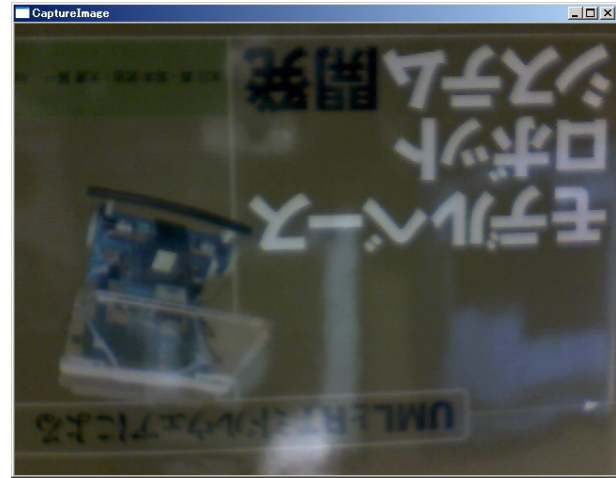
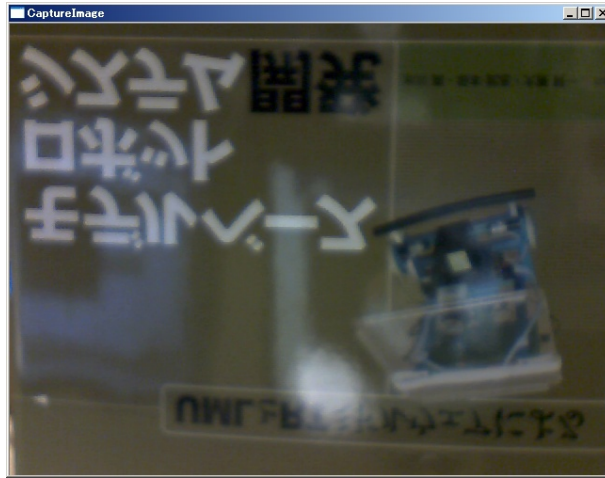
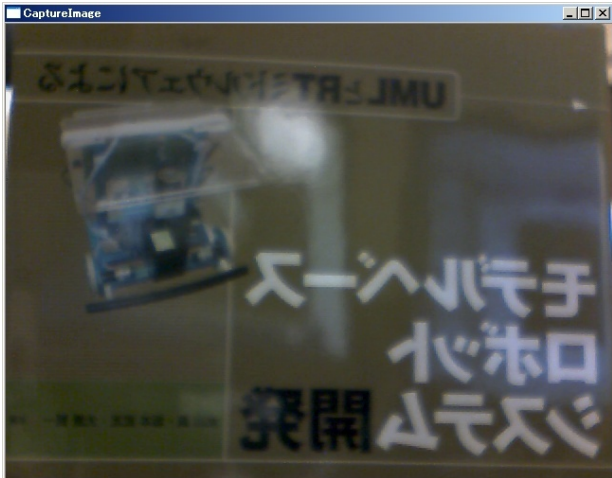
- Connect DirectShowCam → Flip → CameraViewer
(Use the default connection profile).
- Select AllActivate



flip_mode=1

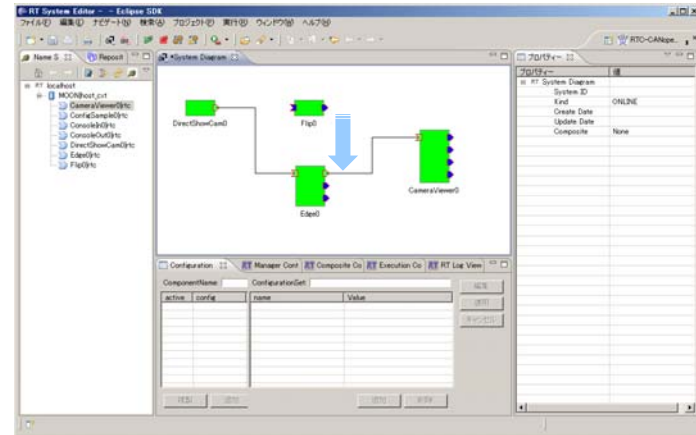
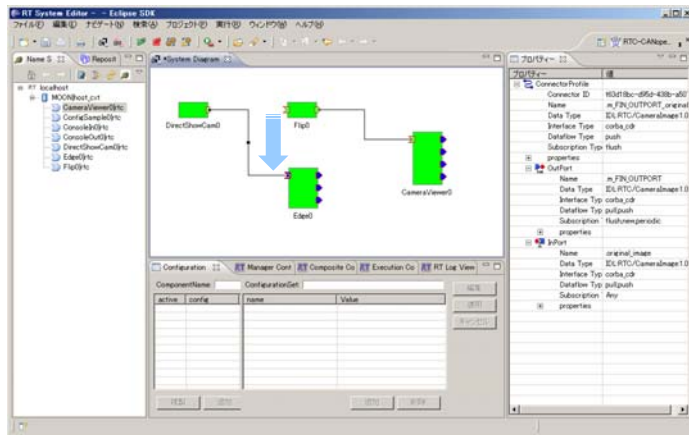
flip_mode=0

flip_mode=-1



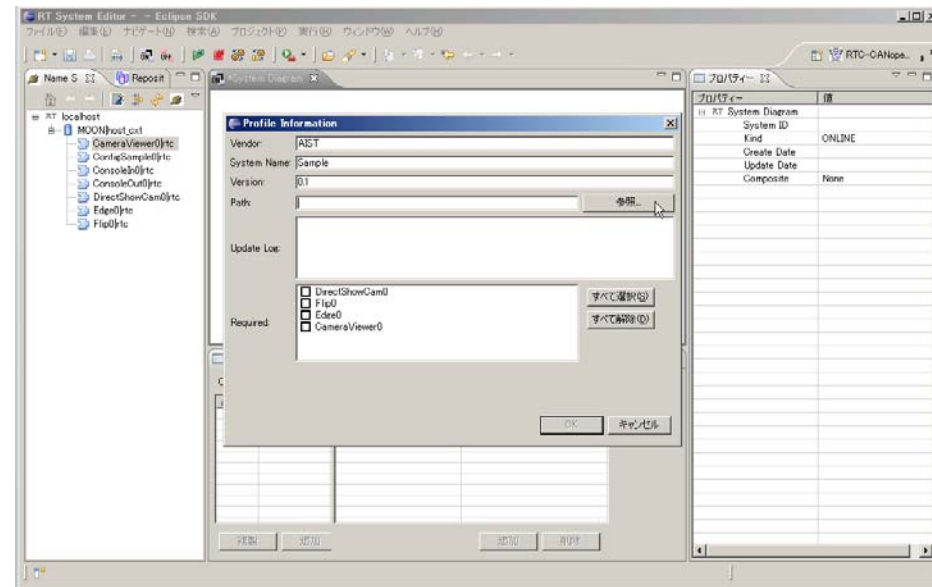
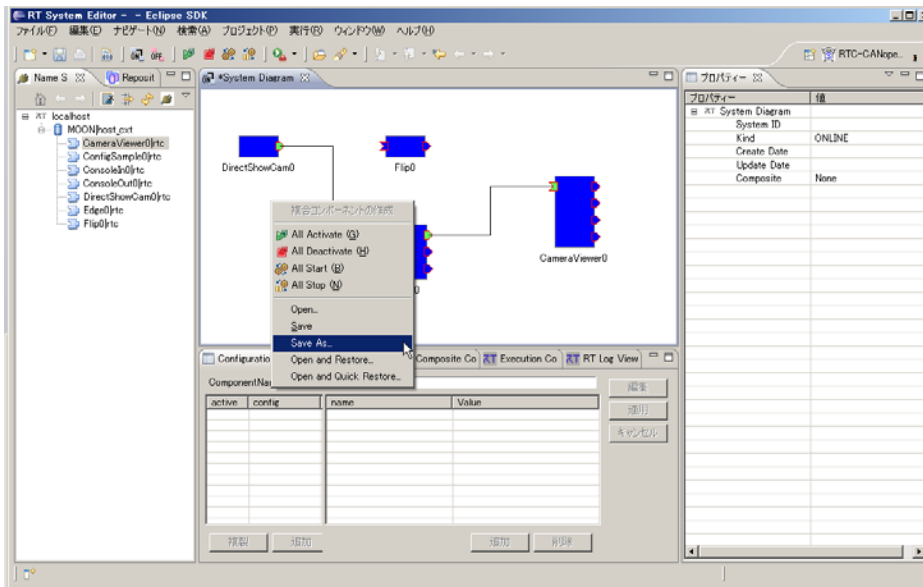
Changing the system

- Replace Flip with Edge
 - Select the connections to Flip
 - Drag the end of the connections at Flip to Edge's ports.
 - Use the default connector profile.



Saving and restoring systems

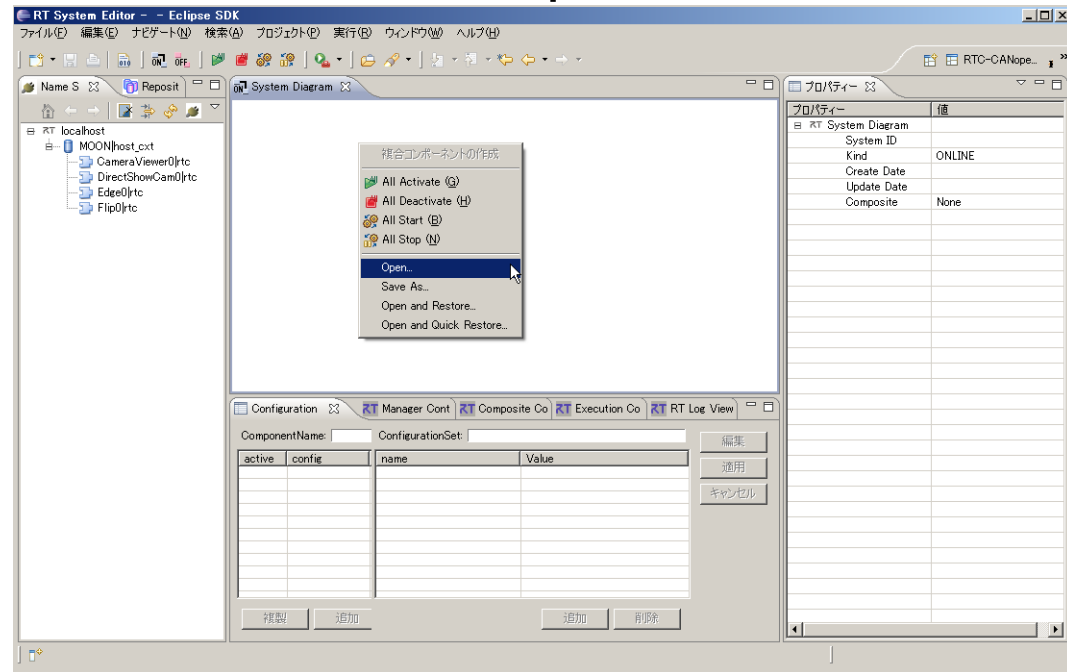
- Right-click on the editor.
- Select “Save as...” from the context menu.
- In the “Profile Information” screen, provide a vendor name, system name, version and file name.



❌ The system will be saved as an RTSPProfile in XML format.

Saving and restoring systems

- Close the editor.
- Restart the components.
- Start an editor, right-click in it, and select one of these options:
 - Open
 - Open and Restore
 - Open and Quick Restore
- Select the file you saved previously.



✘ Open: Load the used components only.

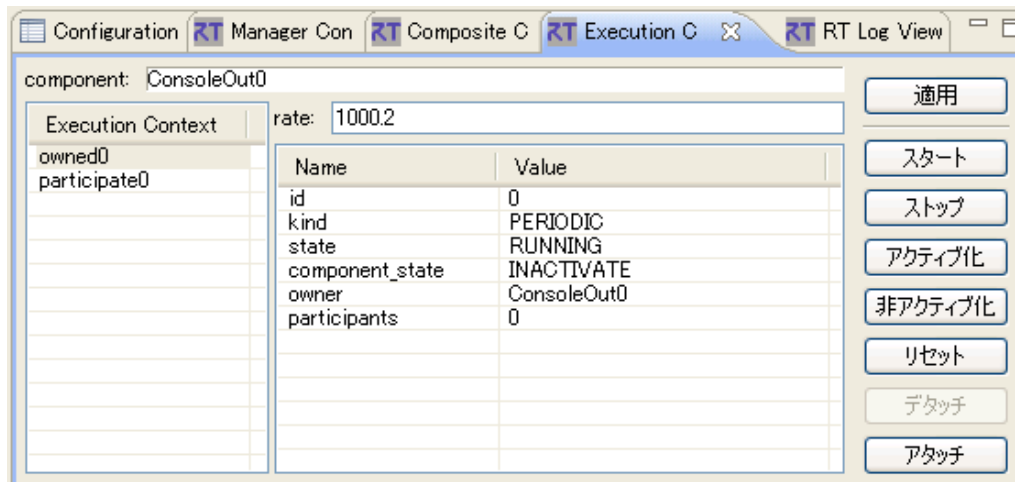
Open and restore: Load the system, restoring connections and configurations parameters.

Open and Quick Restore: Same as above, but search for components using IOR only.

Supplementary explanations

Execution Context view

- Displays the Execution Contexts the RTC is participating in.

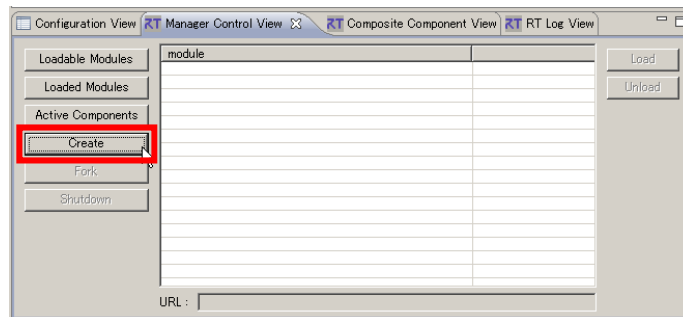


Attribute	Explanation
id	The EC's ID. For online editors, the context_handle.
kind	The EC's type (PERIODIC/EVENT_DRIVEN/OTHER).
state	The EC's state (RUNNING/STOPPING).
component state	The RTC's state in the EC (ACTIVE/INACTIVE/ERROR).
owner	The RTC that owns the EC.
participants	The number of RTCs participating in the EC.

✘ It is also possible to change EC execution rates, start/stop ECs, and manage attached RTCs.

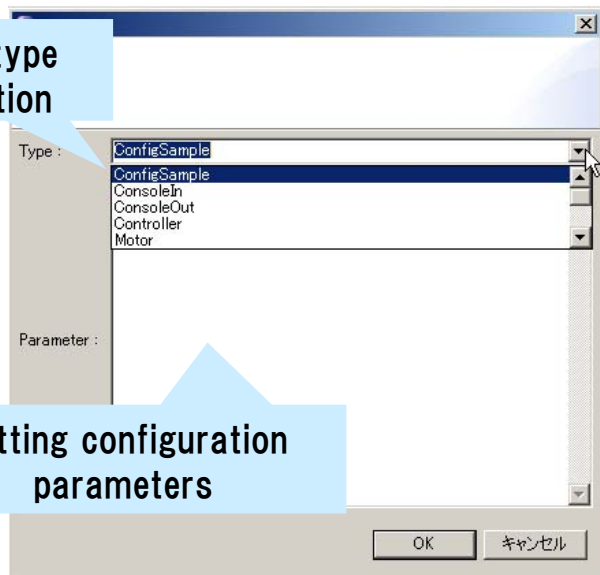
Manager view

■ Manage RTC instances.



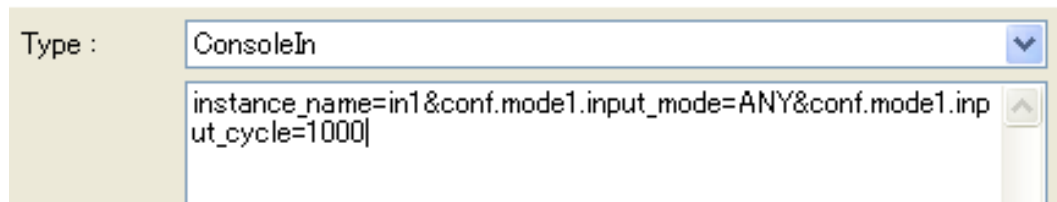
RTC type selection

Setting configuration parameters

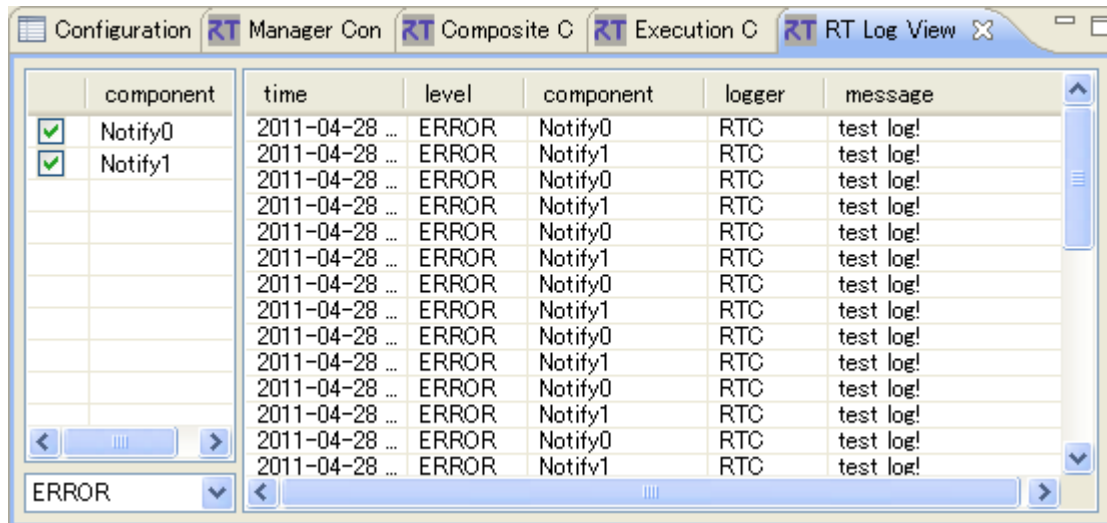


● Setting configuration parameters

- Specify parameters in the following format:
`conf.[configuration set].[Parameter]=[Value]`

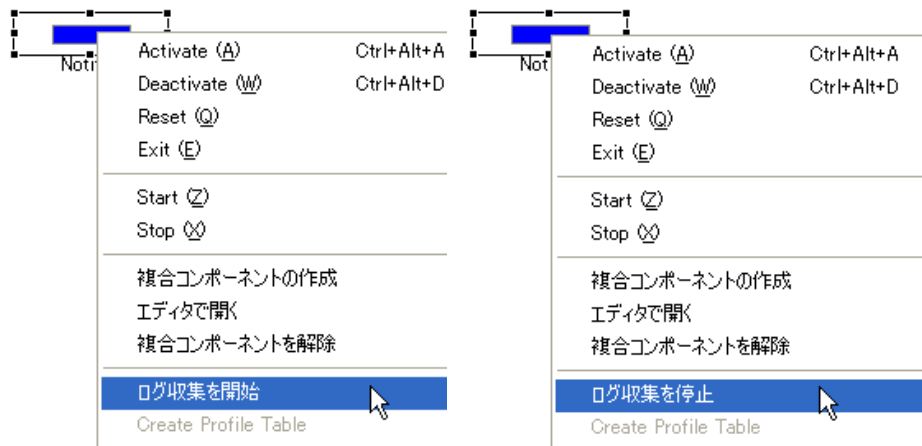


■ Display logs received from the selected RTC.

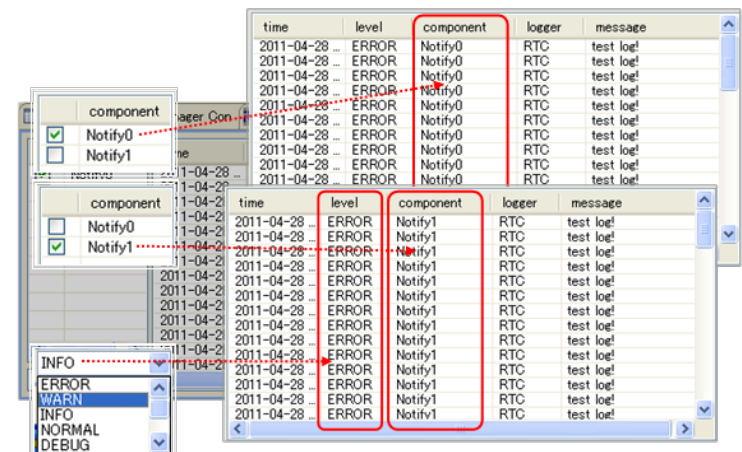


❌ In development

● Starting and stopping logging



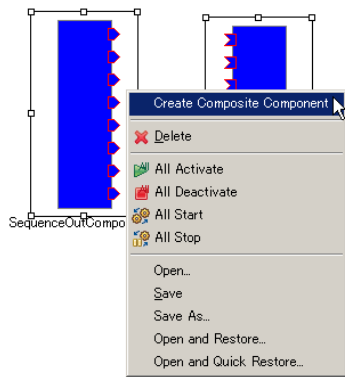
● Log information filtering



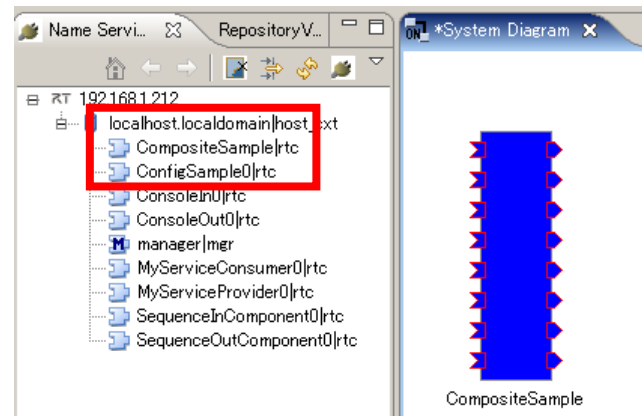
Composite components

- Treat several RTCs as a single RTC.
- Creating a composite component

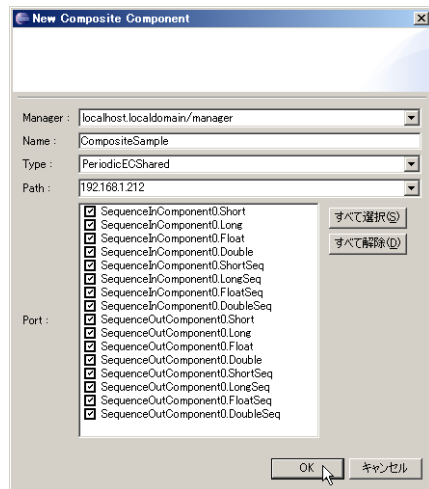
① Select several components and right-click



③ The composite component



② Specify the new component's properties



Item	Meaning
Manager	Select the manager to manage the RTC
Name	The composite RTC's instance name
Type	The type of composite component
Path	The path to the composite component
Port	The ports to export from the component

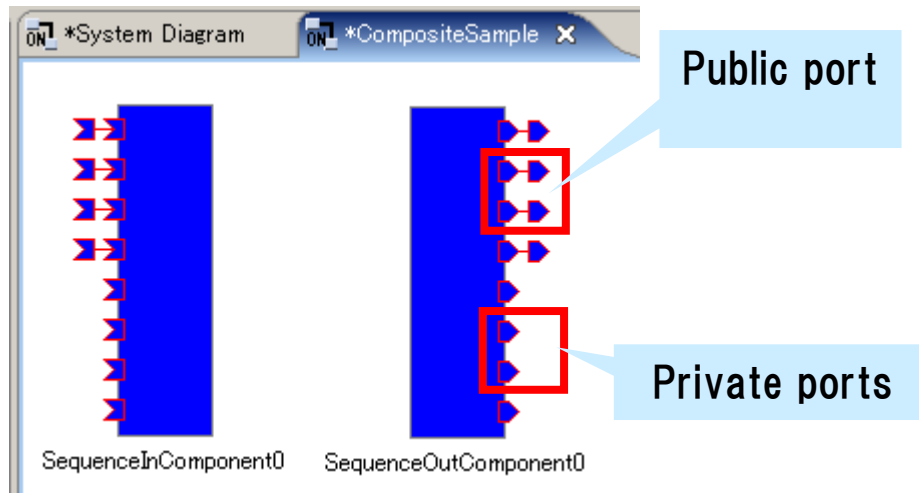
✘ Ports connected outside the composite component are always public.

■ Composite component types

Type name	Explanation
PeriodicECShared	Share only the EC. State is not shared.
PeriodicStateShared	Share state and the EC.
Grouping	Just group the components in the tool.

■ Composite component editor

- Double-click the component to display.



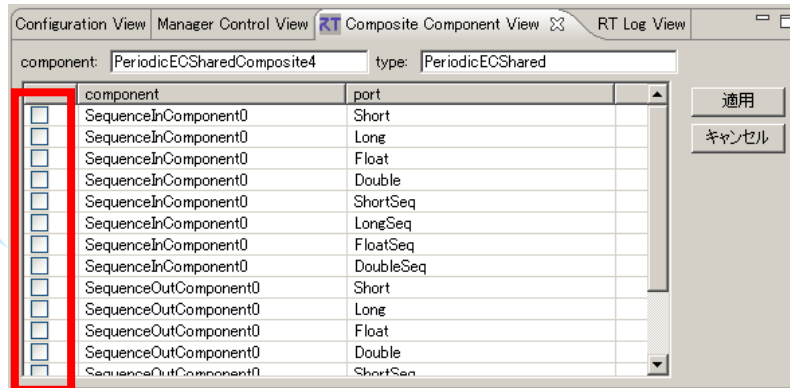
- ✘ Additional components can be added to the composition (all ports will be private).
- ✘ Components can be removed from the composition (it will be displayed in the parent editor again).

Composite components

■ Publicport settings

● Composite component view

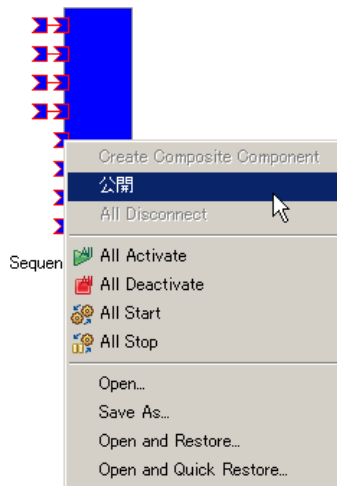
Public port information



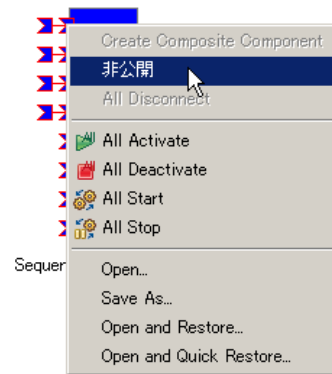
✘ Edit the port information and click “Apply.”

● Composite component editor

✘ Making private ports public



✘ Making public ports private

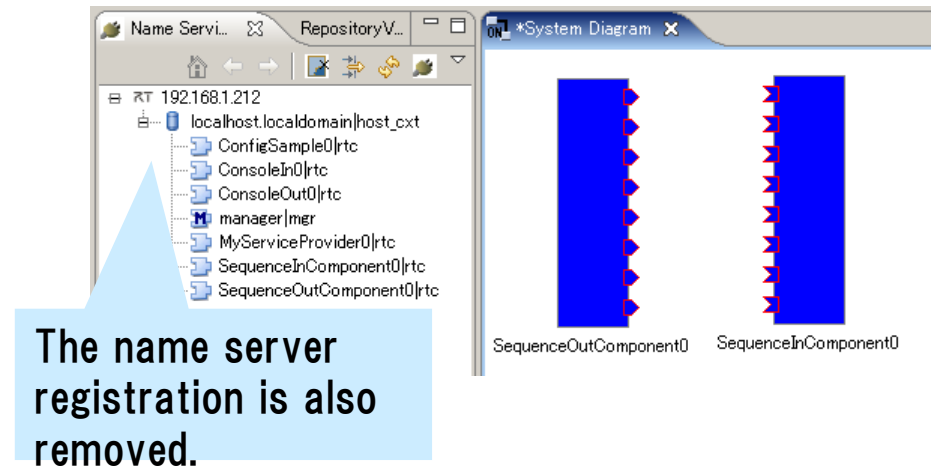
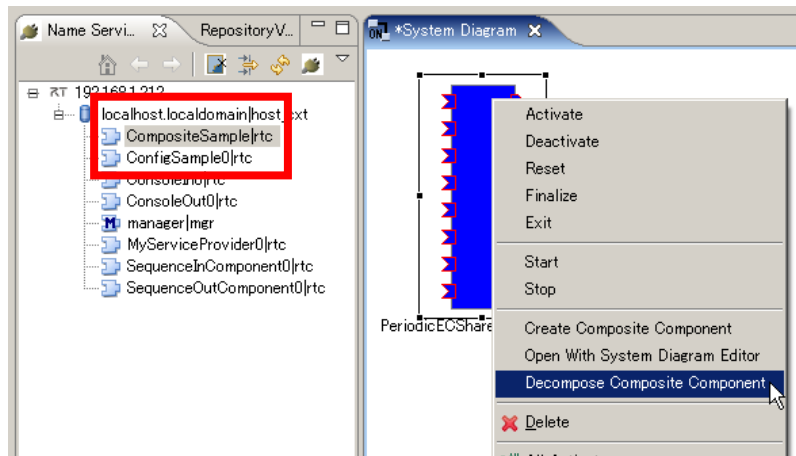


Ports connected externally to the component cannot be made private.

Composite components

■ Decomposing composite components

- ① Right-click on the composite component and select “Decompose...”
- ② The internal components will be displayed again.



❌ Deleting a composite component from the editor does not decompose it.

Offline editor

- Build RT-Systems using RT-Component profiles.
 - Components do not need to be executing.

Repository view

Offline system editor

Property view

Configuration view

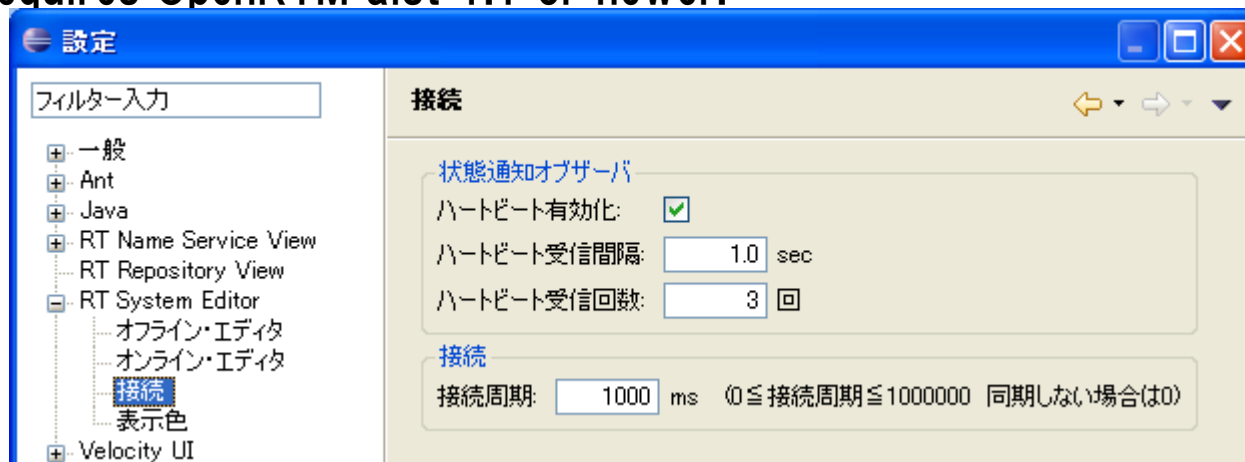
active	config	name	Value

Settings screen

■ State and connection observer

● Settings relating to the RTC observer

- Instead of RTSE polling RTCs for information, the RTCs send heartbeats to RTSE.
- Requires OpenRTM-aist-1.1 or newer.



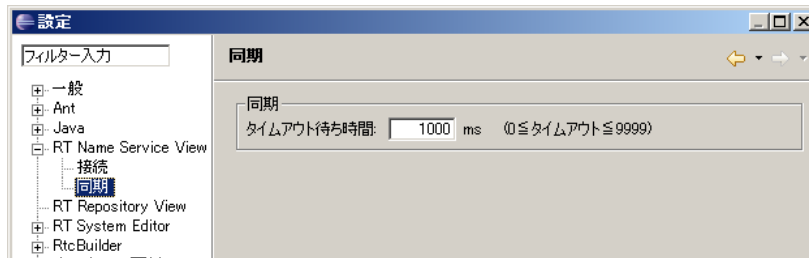
- Use heartbeats: Use heartbeat information to detect liveness.
- Heartbeat period: The maximum length of time to allow between heartbeats.
- Heartbeat count: If the heartbeat of an RTC fails this many times in a row, the RTC is determined to have failed.

Settings screen

- RT Name Service View - Connections – Connection period
 - The period at which to update the name service view.



- RT Name Service View – Frequency - Timeout
 - Time to wait for a response from remote objects in the name service view



If the connection period is set to 0, the name server view will not update.

SICE 2011

RT-Middleware Tutorial

