

RTミドルウェア講習会

日時: 2012年5月22日(水) 10:00～16:45
場所: つくば国際会議場 小会議室303



RTミドルウェア講習会



10:00-10:50	第1部(その1):インターネットを利用したロボットサービスとRSiの取り組み(最新動向)
	担当: 成田 雅彦(産業技術大学院大学) 概要: インターネットやクラウドとロボットとの連携は急速に注目を集めている領域です。本講演では、インターネットやクラウドとロボットとの連携の動向を外観し、RSi(ロボットサービスイニシアティブ)の仕様であるRSNPと最新の取り組みを紹介します。
11:00-11:50	第1部(その2): OpenRTM-aistおよびRTコンポーネントプログラミングの概要
	担当: 安藤 慶昭(産業技術総合研究所) 概要: RTミドルウェアはロボットシステムをコンポーネント指向で構築するソフトウェアプラットフォームです。RTミドルウェアを利用することで、既存のコンポーネントを再利用し、モジュール指向の柔軟なロボットシステムを構築することができます。RTミドルウェアの産総研による実装であるOpenRTM-aistについてその概要およびRTコンポーネントの機能やプログラミングの流れについて説明します。
11:50-12:00	質疑応答・意見交換
12:00-13:00	昼食
13:00-14:00	第2部: RTコンポーネントの作成入門
	担当: 坂本 武志(株式会社グローバルアシスト) 概要: RTCBuilderを使用したRTコンポーネントの作成方法を実習形式で体験していただきます。
14:00-16:45	第3部: プログラミング実習
	担当: Geoffrey Biggs, 原功, 安藤慶昭(産業技術総合研究所), 坂本武志(株式会社グローバルアシスト) 概要: OpenRTM-aistを利用してコンポーネントを作成し実際にロボットを動かしていただきます。 今回は2つのコース(Kobuki & Raspberry Piコース, G-ROBOT & Choreonoidコース)を用意しました。

第2部 RTコンポーネントの作成入門

株式会社 グローバルアシスト
坂本 武志



2013.5.22 ROBOMECH2013 RTM講習会

3

OpenRT Platform



- ロボット知能ソフトウェアプラットフォーム
 - <http://www.openrtp.jp/wiki/>
 - システム設計, シミュレーション, 動作生成, シナリオ生成などをサポート
 - OpenRT Platformツール群
 - コンポーネント開発, システム開発における各開発フェーズの作業支援
 - 開発プラットフォームにEclipseを採用
 - 構成
 - RTCビルダ
 - RTCデバッグ
 - RTシステムエディタ
 - ロボット設計支援ツール
 - シミュレータ
 - 動作設計ツール
 - シナリオ作成ツール
- など

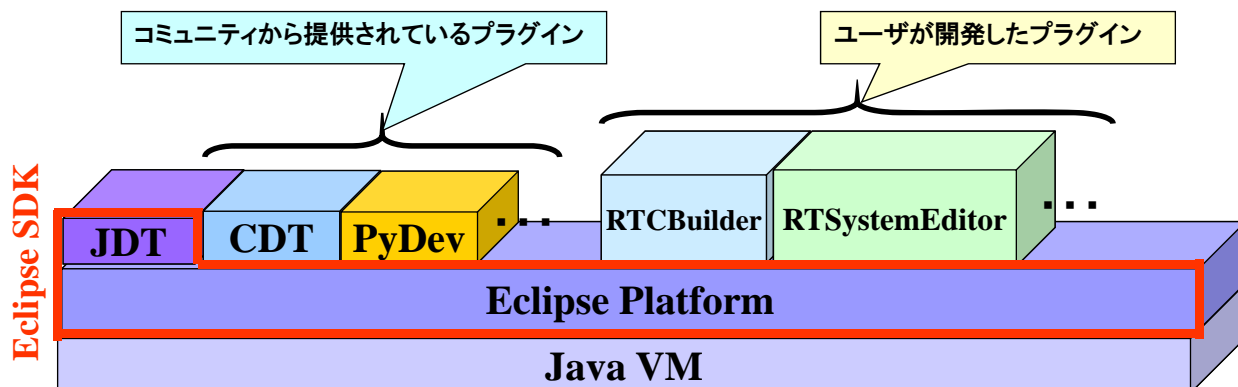


2013.5.22 ROBOMECH2013 RTM講習会

4

■ オープンソース・コミュニティで開発されている統合開発環境

- マルチプラットフォーム対応. WindowsやLinuxなど複数OS上で利用可能
- 「Plug-in」形式を採用しており, 新たなツールの追加, 機能のカスタマイズが可能
- RCP(Rich Client Platform)を利用することで, 簡単に単独アプリ化が可能



RTCBuilder, RTSystemEditorのインストール

■ ダウンロードし, 解凍するだけ

※Javaの実行環境については, 別途インストールが必要

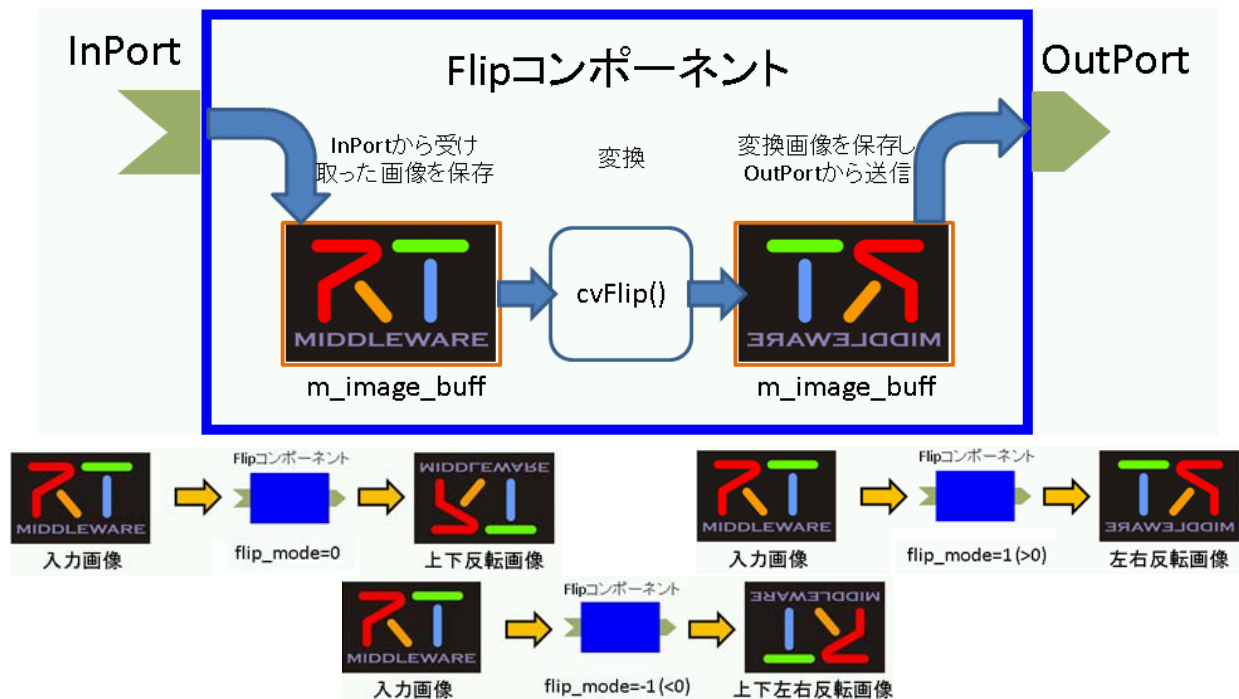
The screenshot shows the OpenRTM-aist website. The main content area displays 'OpenRTM Eclipse tools 1.1.0-RC2'. Below this, there is a 'Table of contents' section with links to '全部入りパッケージ' (Full package), 'バイナリ' (Binaries), 'Eclipse/JDK/JRE等' (Eclipse/JDK/JRE etc), and '過去のバージョン' (Past versions). A section titled '全部入りパッケージ' (Full package) contains a table with download links for different operating systems and versions.

Eclipse-3.4.2 [Gangmode SR2]		
Eclipse] 4.2-RTSE-RTCB	eclipse342_rtmtools110-rc2_win32_ja.zip	2011.07.22
Windows/日本語入り	MD5:2e6f9fa3e370b6e7ac1f934b36c7abf	

Below the table, there are instructions for installing the tools on Ubuntu 8.04, Ubuntu 9.10, Ubuntu 10.04, and Linux (Eclipse) 4.2. The instructions include commands for setting up the environment and installing the tools.

```
$ su
# vi /etc/apt/sources.list
1行追加 - deb http://jp.archive.ubuntu.com/ubuntu/ jaunty main restricted
# apt-get update
# apt-get install xulrunner-1.9
# dpkg -i | grep xulrunner-1.9
$ xulrunner-1.9
```

- 入力画像を反転して出力するコンポーネント
 - OpenCVのcvFlip関数を利用



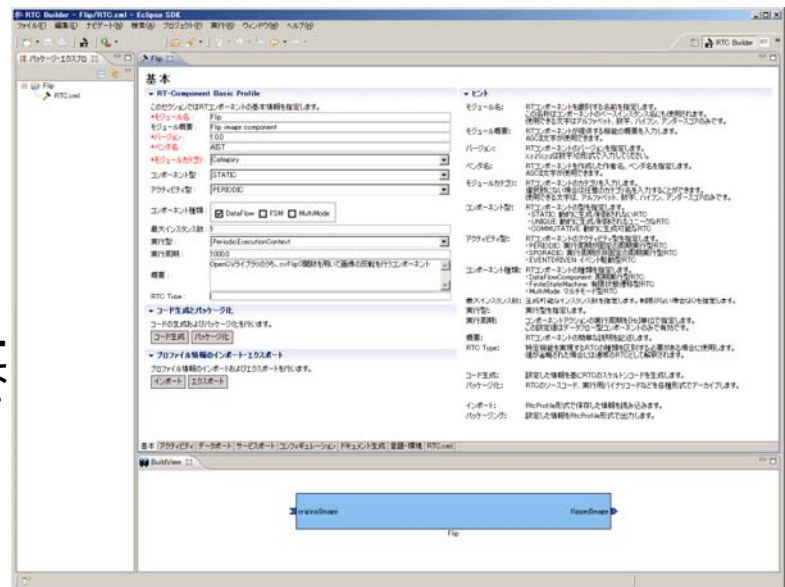
コンポーネント開発ツール RTCBuilderについて

RTCBuilder概要

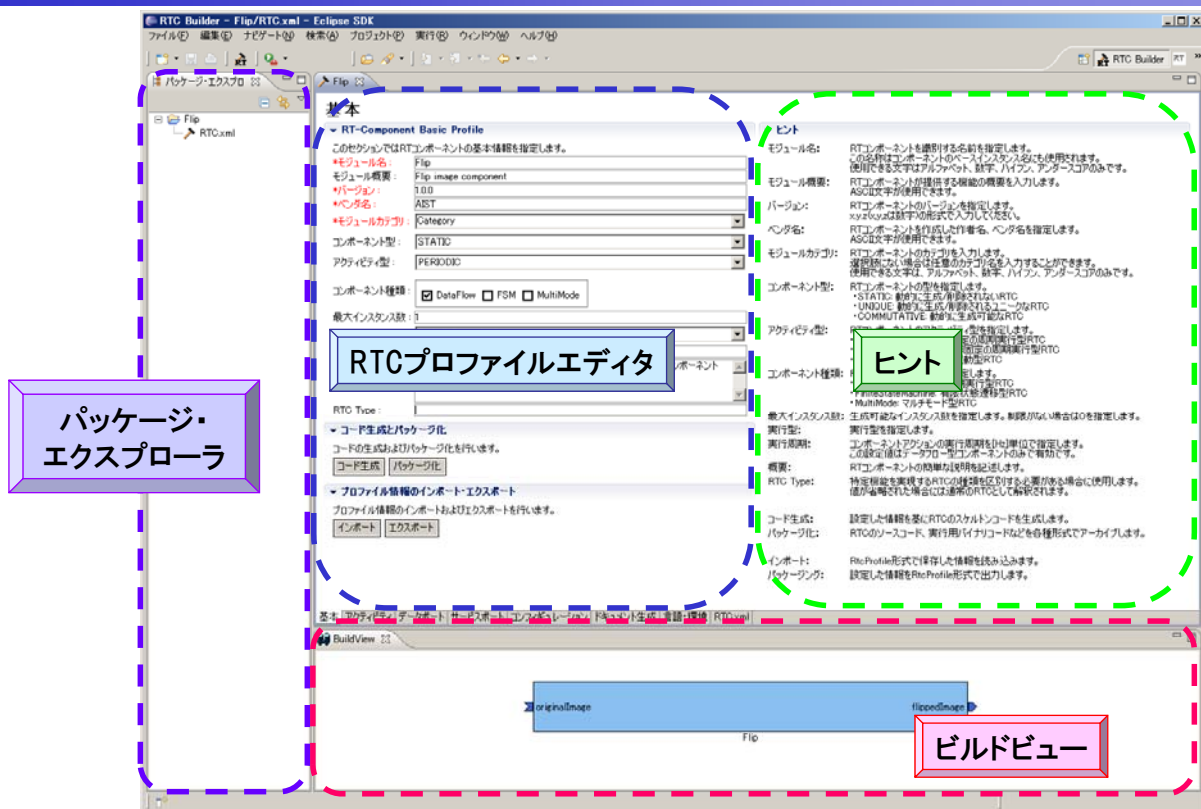
■ RTCBuilderとは？

- コンポーネントのプロファイル情報を入力し、ソースコード等の雛形を生成するツール
- 開発言語用プラグインを追加することにより、各言語向けRTCの雛形を生成することが可能
 - C++
 - Java
 - Python

- ※C++用コード生成機能はRtcBuilder本体に含まれています。
- ※その他の言語用コード生成機能は追加プラグインとして提供されています

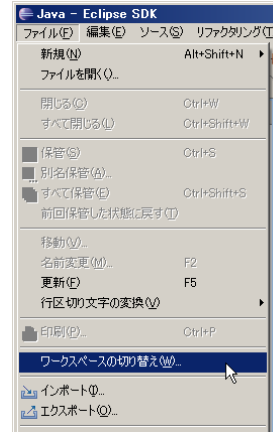
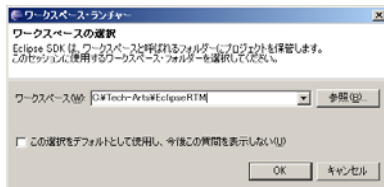


画面構成



ツールの起動

- Windowsの場合
 - Eclipse.exeをダブルクリック
- Unix系の場合
 - ターミナルを利用してコマンドラインから起動
 - Ex) \$ /usr/local/Eclipse/eclipse
- ワークスペースの選択(初回起動時)
- ワークスペースの切替(通常時)



※ワークスペース

Eclipseで開発を行う際の作業領域

Eclipse上でプロジェクトやファイルを作成するとワークスペースとして指定したディレクトリ以下に実際のディレクトリ、ファイルを作成する

準備

- 初期画面のクローズ
 - 初回起動時のみ

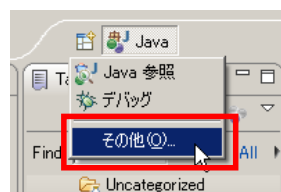
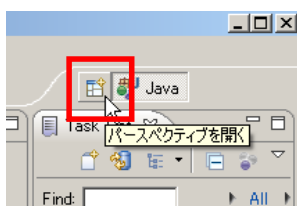


※パースペクティブ

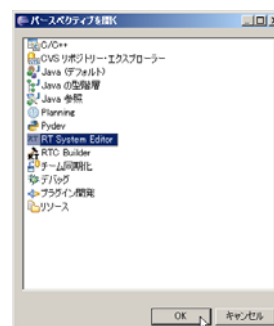
Eclipse上でツールの構成を管理する単位
メニュー、ツールバー、エディタ、ビューなど
使用目的に応じて組み合わせる
独自の構成を登録することも可能

- パースペクティブの切り替え

①画面右上の「パースペクティブを開く」を選択し、一覧から「その他」を選択

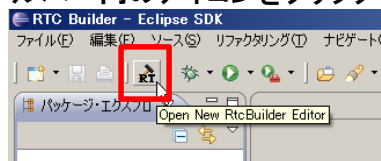


②一覧画面から対象ツールを選択



プロジェクト作成/エディタ起動

① ツールバー内のアイコンをクリック

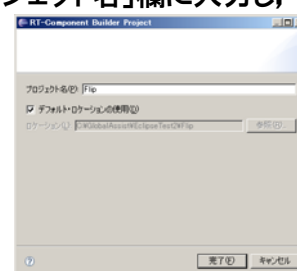


- ※メニューから「ファイル」-「新規」-「プロジェクト」を選択
【新規プロジェクト】画面にて「その他」-「RtcBuilder」を選択し、「次へ」
- ※メニューから「ファイル」-「Open New Builder Editor」を選択

- ※任意の場所にプロジェクトを作成したい場合
②にて「デフォルト・ロケーションの使用」チェックボックスを外す
「参照」ボタンにて対象ディレクトリを選択
→物理的にはワークスペース以外の場所に作成される
論理的にはワークスペース配下に紐付けされる

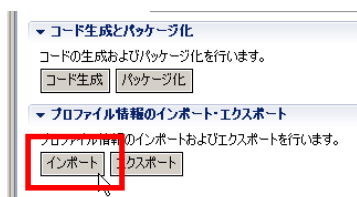
プロジェクト名: Flip

② 「プロジェクト名」欄に入力し, 「終了」

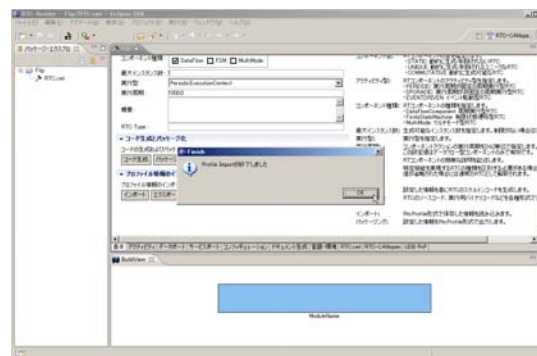


プロファイル インポート

① 「基本」タブ下部の「インポート」ボタンをクリック



② 【インポート】画面にて対象ファイルを選択



■ 作成済みのRTコンポーネント情報を再利用

- 「エクスポート」機能を利用して出力したファイルの読み込みが可能
- コード生成時に作成されるRtcProfileの情報を読み込み可能
- XML形式, YAML形式での入出力が可能

■ コード生成

RTC Type :

▼ コード生成とパッケージ化

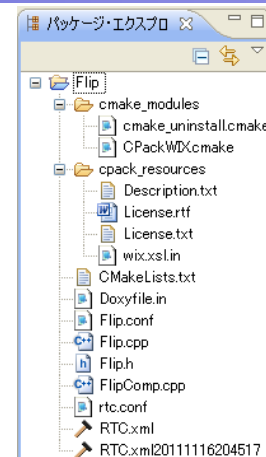
コードの生成およびパッケージ化を行います。

コード生成 パッケージ化

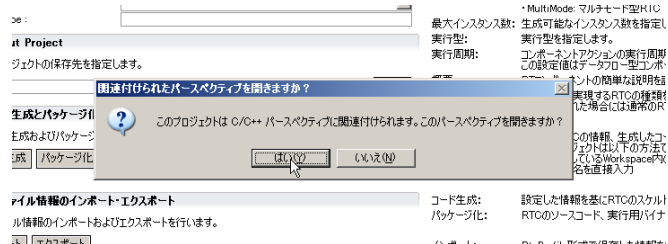
▼ プロファイル情報のインポート・エクスポート

プロファイル情報のインポートおよびエクスポートを行います。

インポート エクスポート



■ コード生成実行後、パースペクティブを自動切替



※生成コードが表示されない場合には、「リフレッシュ」を実行

C++版RTC → CDT

Java版RTC → JDT

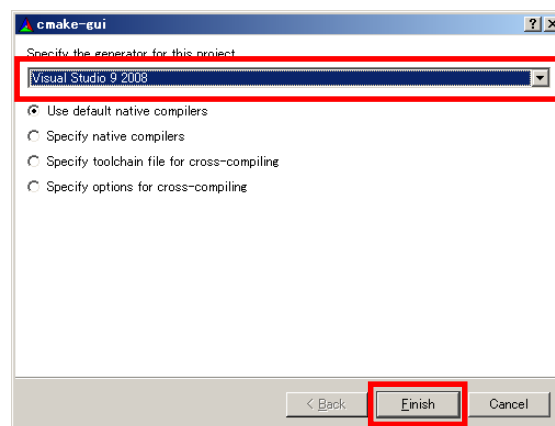
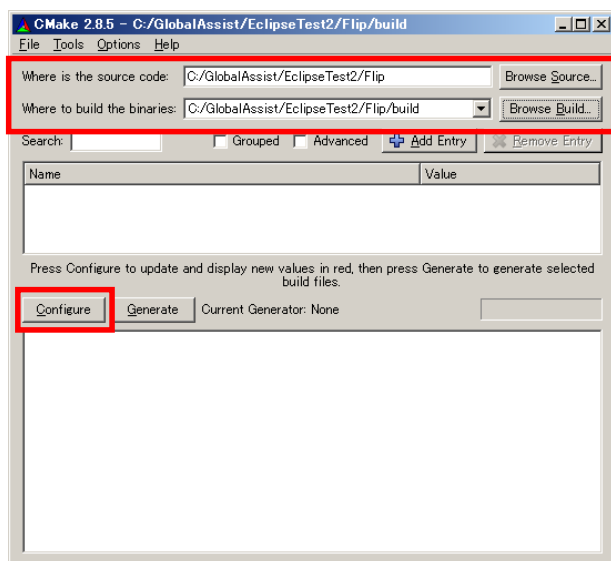
(デフォルトインストール済み)

Python版 → PyDev

コンパイル(Windows,CMake利用)

①GUI版Cmakeを起動し, source, binaryのディレクトリを指定

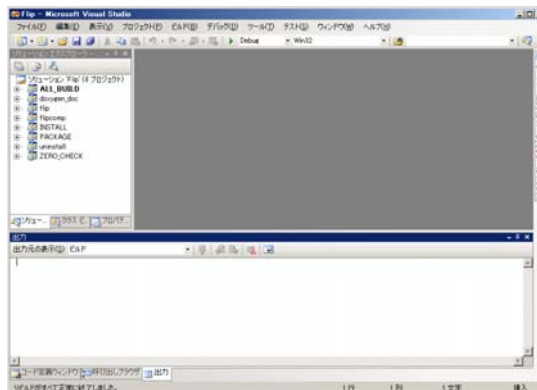
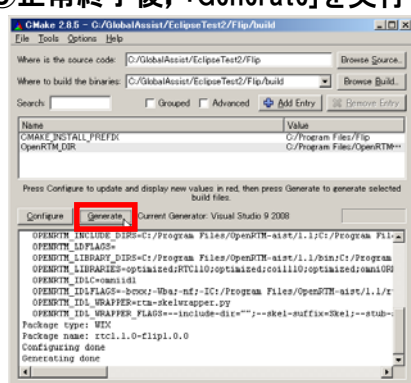
②「Configure」を実行し, 使用するプラットフォームを選択



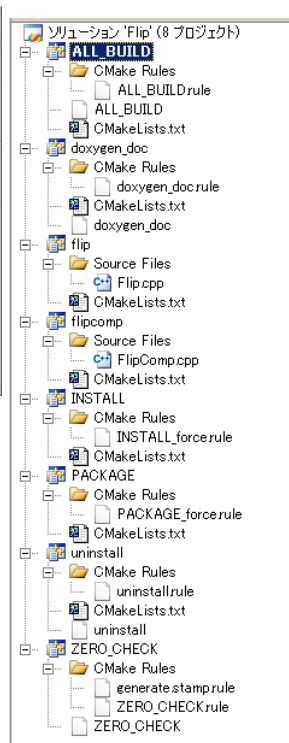
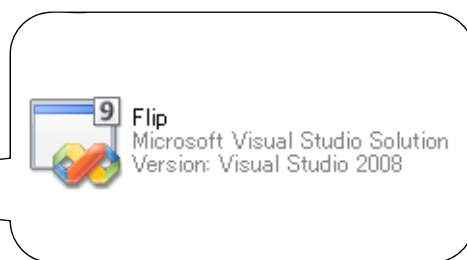
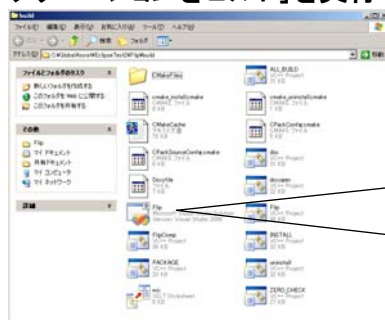
※binaryには, sourceとは別のディレクトリを指定する事を推奨

※日本語は文字化けしてしまうため英数字のみのディレクトリを推奨

③正常終了後、「Generate」を実行



④binaryとして指定したディレクトリ内にあるソリューションファイルを開き、「ソリューションをビルド」を実行



RTCプロファイルエディタ

画面要素名	説明
基本プロファイル	RTコンポーネントのプロファイル情報など、コンポーネントの基本情報を設定。 コード生成、インポート/エクスポート、パッケージング処理を実行
アクティビティ・プロファイル	RTコンポーネントがサポートしているアクティビティ情報を設定
データポート・プロファイル	RTコンポーネントに付属するデータポートに関する情報を設定
サービスポート・プロファイル	RTコンポーネントに付属するサービスポートおよび各サービスポートに付属するサービスインターフェースに関する情報を設定
コンフィギュレーション	RTコンポーネントに設定するユーザ定義のコンフィギュレーション・パラメータセット情報およびシステムのコンフィギュレーション情報を設定
ドキュメント生成	生成したコードに追加する各種ドキュメント情報を設定
言語・環境	生成対象コードの選択やOSなどの実行環境に関する情報を設定
RTC.xml	設定した情報を基に生成したRTC仕様(RtcProfile)を表示

- RTコンポーネントの名称など、基本的な情報を設定

基本

▼ RT-Component Basic Profile

このセクションではRTコンポーネントの基本情報を指定します。

*モジュール名: Flip

モジュール概要: Flip image component

*バージョン: 1.0.0

*ベンダ名: AIST

*モジュールカテゴリ: Category

コンポーネント型: STATIC

アクティビティ型: PERIODIC

コンポーネント種類: ☒ DataFlow ☐ FSM ☐ MultiMode

最大インスタンス数: 1

実行型: PeriodicExecutionContext

実行周期: 0.0

OpenCVライブラリのうち、cvFlip関数を用いて画像の反転を行うコンポーネント

概要:

RTC Type:

▼ ヒント

モジュール名: RTコンポーネントを識別する名前を指定します。この名称はコンポーネントのベースインスタンス名にも使用されます。使用できる文字はアルファベット、数字、ハイフン、アンダースコアのみです。

モジュール概要: RTコンポーネントが提供する機能の概要を入力します。ASCII文字が使用できます。

バージョン: RTコンポーネントのバージョンを指定します。

モジュール名: Flip

モジュール概要: 任意(Flip image component)

バージョン: 1.0.0

ベンダ名: 任意(AIST)

モジュールカテゴリ: 任意(Category)

コンポーネント型: STATIC

アクティビティ型: PERIODIC

コンポーネントの種類: DataFlow

最大インスタンス数: 1

実行型: PeriodicExecutionContext

実行周期: 1000.0

※エディタ内の項目名が赤字の要素は必須入力項目
※画面右側は各入力項目に関する説明

アクティビティ・プロフィール

■ 生成対象RTCで実装予定のアクティビティを設定

アクティビティ

▼ アクティビティ

このセクションでは使用するアクティビティのワークフローを指定します。

onInititalize	onFinalize
onStartUp	onShutdown
onStartup	onShutdown
onActivated	onDeactivated
onError	onReset
onExecute	onStateUpdate
onModeChangend	onRateChangend
onAction	Mode変更イベントのアクション
onModeChangend	Mode変更イベントのアクション

▼ Documentation

このセクションでは各アクションの概要と使用可能なドキュメントを記述します。
上段のアクションを選択すると、それらのドキュメントが記録されます。

アクティビティ名: ☒ ON ☐ OFF

動作概要	ドキュメント
コンポーネント自身の各種初期化処理	ドキュメント
なし	ドキュメント
事前条件	ドキュメント
事後条件	ドキュメント

コンポーネントの初期化と処理が正常に完了している

①設定対象のアクティビティを選択

②使用/未使用を設定

以下をチェック:

- onActivated
- onDeactivated
- onExecute

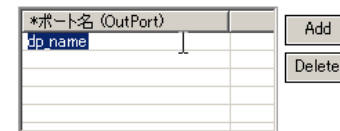
- ※現在選択中のアクティビティは、一覧画面にて赤字で表示
- ※使用(ON)が選択されているアクティビティは、一覧画面にて背景を水色で表示
- ※各アクティビティには、「動作概要」「事前条件」「事後条件」を記述可能
→記述した各種コメントは、生成コード内にDoxygen形式で追加される

■ 生成対象RTCに付加するDataPortの情報を設定

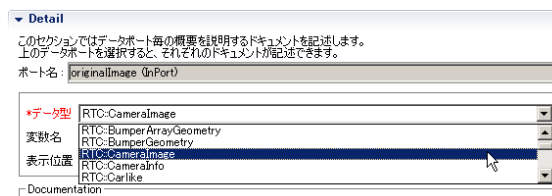


① 該当種類の欄の「Add」ボタンをクリックし、ポートを追加後、直接入力で名称設定

※ポート)の情報を設定します。

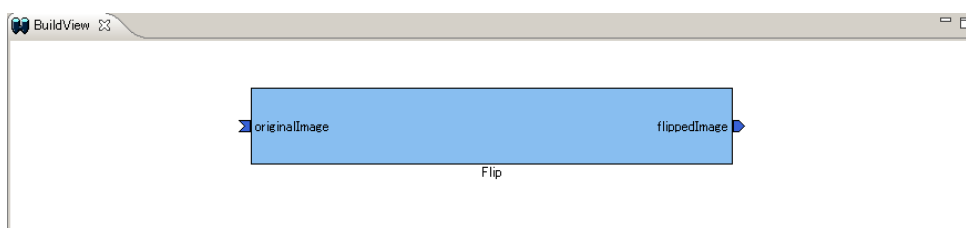


② 設定する型情報を一覧から選択



- ※データ型は、型定義が記載されたIDLファイルを設定画面にて追加することで追加可能
- ※OpenRTM-aistにて事前定義されている型については、デフォルトで使用可能
→ [RTM_Root]rtm/idl 以下に存在するIDLファイルで定義された型
- ※各ポートに対する説明記述を設定可能
→ 記述した各種コメントは、生成コード内にDoxygen形式で追加される

※Portの設定内容に応じて、下部のBuildViewの表示が変化



- InPort
ポート名: **originalImage**
データ型: **RTC::CameraImage**
変数名: **originalImage**
表示位置: **left**
- OutPort
ポート名: **flippedImage**
データ型: **RTC::CameraImage**
変数名: **flippedImage**
表示位置: **right**

サービスポート・プロファイル

■ 生成対象RTCに付加するServicePortの情報を設定

■ サービスインターフェースの指定

- IDLファイルを指定すると、定義されたインターフェース情報を表示

今回のサンプルでは未使用

コンフィギュレーション・プロファイル

■ 生成対象RTCで使用する設定情報を設定

コンフィギュレーション・パラメータ

- ①「Add」ボタンをクリックし、追加後、直接入力で名称設定

- ②詳細画面にて、型情報、変数名などを設定

名称: flipMode
データ型: int
デフォルト値: 0
変数名: flipMode
制約条件: (-1, 0, 1)
Widget: radio

- ※データ型は、short,int,long,float,double,stringから選択可能(直接入力も可能)
- ※制約情報とWidget情報を入力することで、RTSystemEditorのコンフィギュレーションビューの表示を設定することが可能

制約条件, Widgetの設定方法

■ 制約条件について

- データポートとコンフィギュレーションに設定可能
- チェックはあくまでも**コンポーネント開発者側の責務**
 - ミドルウェア側で検証を行っているわけではない

■ 制約の記述書式

- 指定なし: 空白
- 即値: 値そのもの
 - 例) 100
- 範囲: <, >, <=, >=
 - 例) 0<=x<=100
- 列挙型: (値1, 値2, ...)
 - 例) (val0, val1, val2)
- 配列型: 値1, 値2, ...
 - 例) val0, val1, val2
- ハッシュ型: { key0: 値0, key1: 値1, ... }
 - 例) { key0: val0, key1: val1 }

■ Widget

- text(テキストボックス)
 - デフォルト
- slider(スライダー)
 - **数値型**に対して**範囲指定**の場合
 - 刻み幅をstepにて指定可能
- spin(スピナ)
 - **数値型**に対して**範囲指定**の場合
 - 刻み幅をstepにて指定可能
- radio(ラジオボタン)
 - 制約が**列挙型**の場合に指定可能

※ 指定したWidgetと制約条件がマッチしない場合は、テキストボックスを使用

言語・環境・プロファイル

■ 生成対象RTCを実装する言語, 動作環境に関する情報を設定

言語・環境

▼ 言語
このセクションでは使用する言語を指定します

☒ C++
☐ Python
☐ Java
☐ Ruby

☐ Use old build environment.

▼ 環境
このセクションでは依存するライブラリや使用するOSなどを指定します

Version	OS

Add
Delete

詳細情報

OS Version	CPU

Add
Delete

Add
Delete

言語: RTコンポーネントを作成する言語を選択します。リスト中の言語から選択可能です。
環境: 言語ごとのライブラリの依存関係や、使用するOSなどの環境を選択します。
詳細情報で設定した内容(OS情報、ライブラリ情報など)は、プロファイル内にのみ保存されます。

このチェックボックスをONにすると、旧バージョンと同様なコード(Cmakeを利用しない形式)を生成

「C++」を選択

システム構築支援ツール RTSystemEditorについて

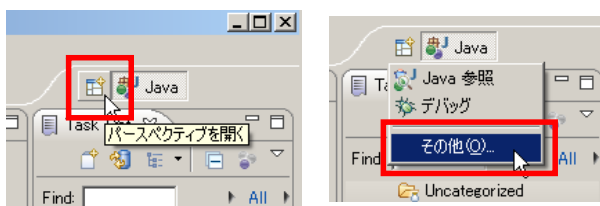


準備

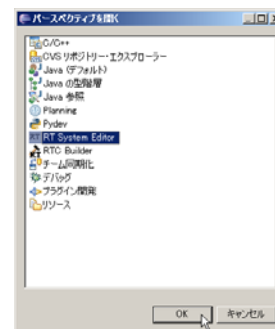


■ パースペクティブの切り替え

①画面右上の「パースペクティブを開く」を選択し、一覧から「その他」を選択



②一覧画面から対象ツールを選択

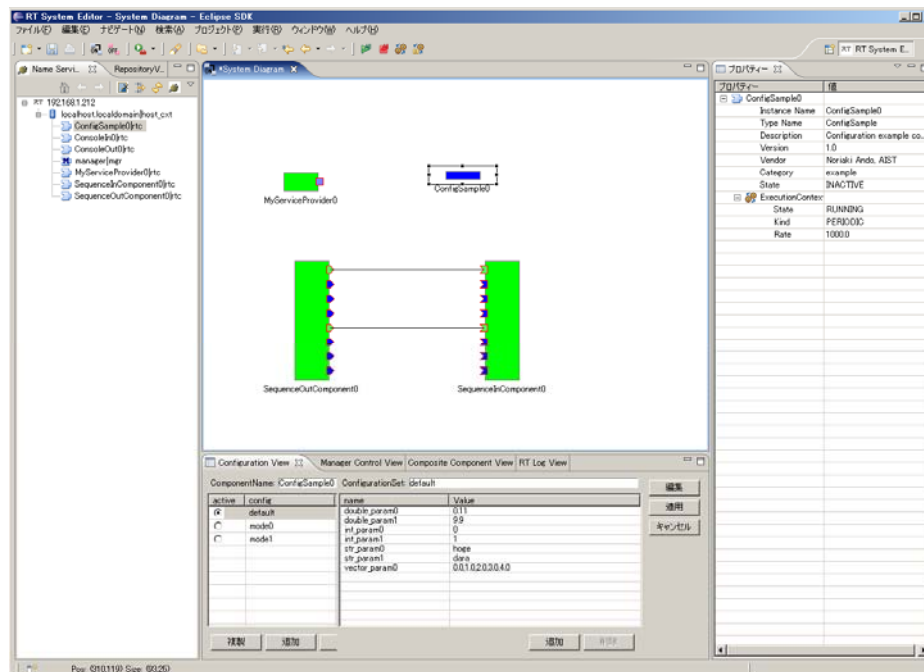


※パースペクティブ

Eclipse上でツールの構成を管理する単位
メニュー、ツールバー、エディタ、ビューなど
使用目的に応じて組み合わせる
独自の構成を登録することも可能

■ RTSystemEditorとは？

- RTコンポーネントを組み合わせて、RTシステムを構築するためのツール



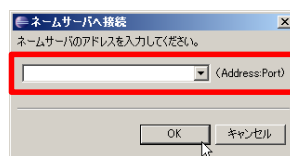
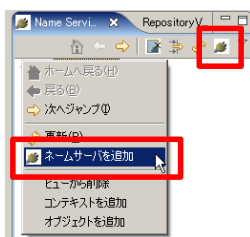
画面構成



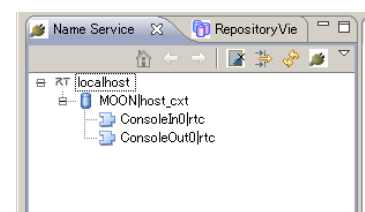
- Naming Serviceの起動
 - [スタート]メニューから
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[tools]→[Start Naming Service]
- CameraViewerCompの起動
 - [スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[opencv-rtcs]→ [CameraViewerComp.exe]
- DirectShowCamCompの起動
 - [スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[opencv-rtcs]→ [DirectShowCamComp.exe]

RTシステム構築の基本操作

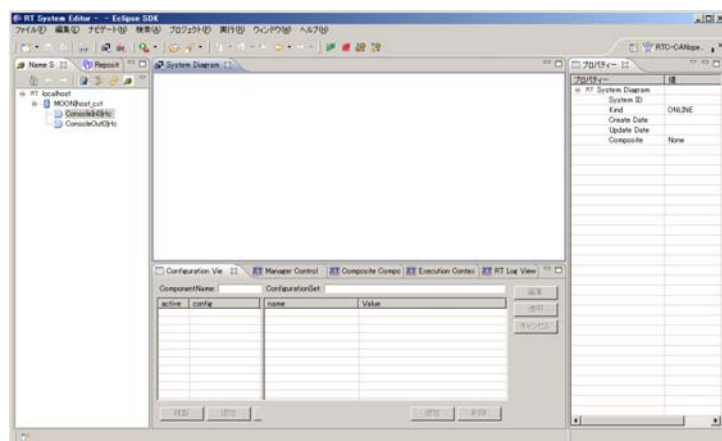
■ ネームサービスへ接続



※対象ネームサーバのアドレス、ポートを指定
→ポート省略時のポート番号は
設定画面にて設定可能

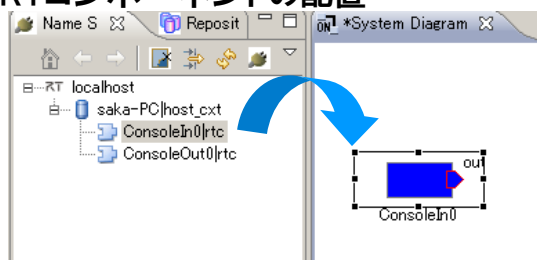


■ システムエディタの起動

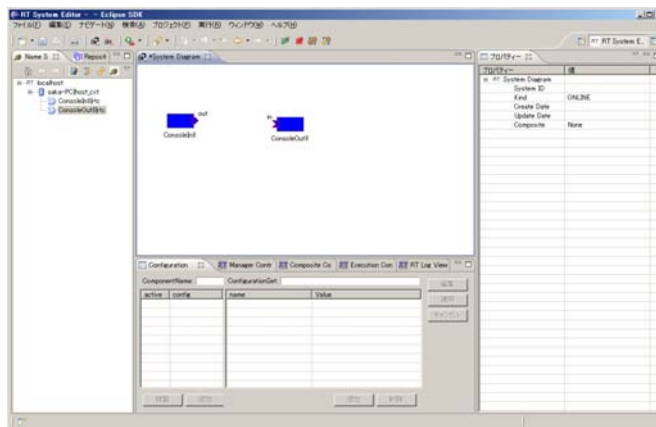


RTシステム構築の基本操作

RTコンポーネントの配置

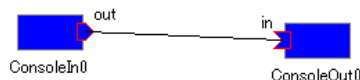


※ネームサービスビューから対象コンポーネントをドラッグアンドドロップ

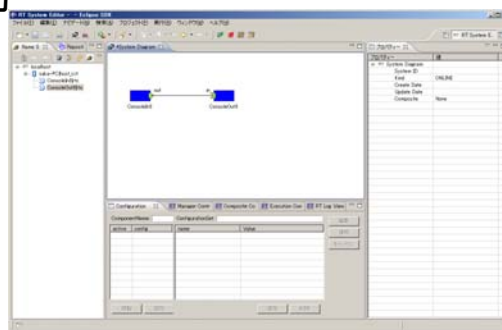


ポートの接続

①接続元のポートから接続先の②接続プロファイルを入力
ポートまでドラッグ



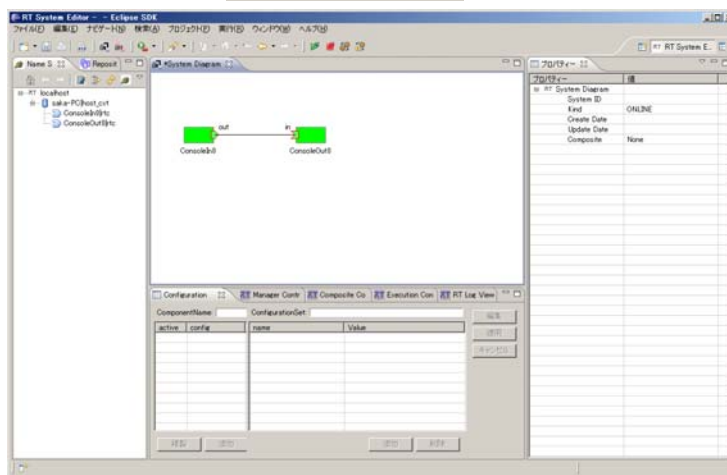
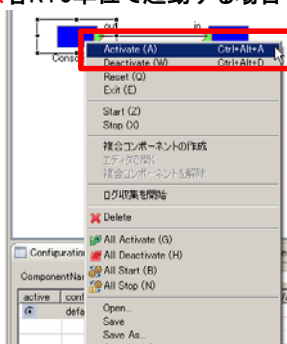
※ポートのプロパティが異なる場合など、接続不可能なポートの場合にはアイコンが変化



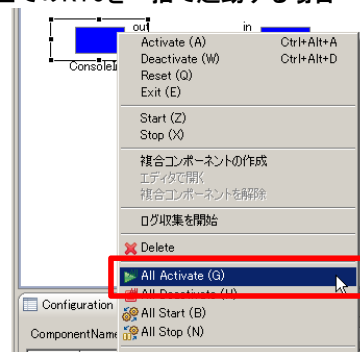
RTシステム構築の基本操作

コンポーネントの起動

※各RTC単位で起動する場合

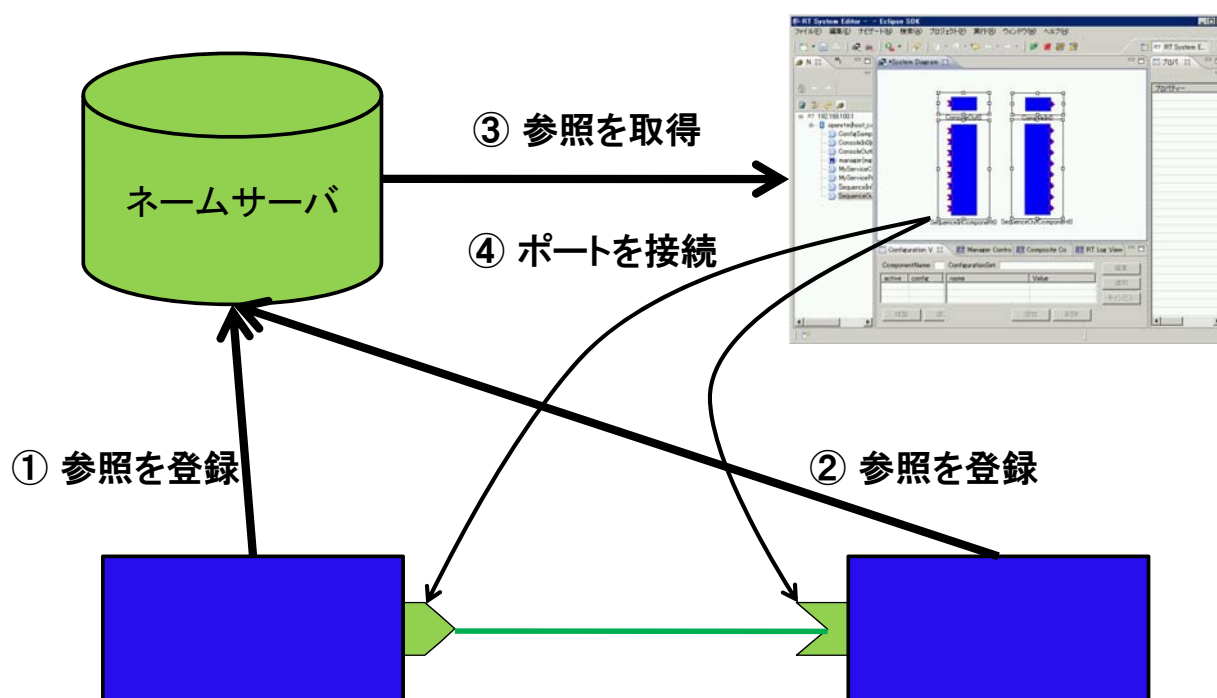


※全てのRTCを一括で起動する場合



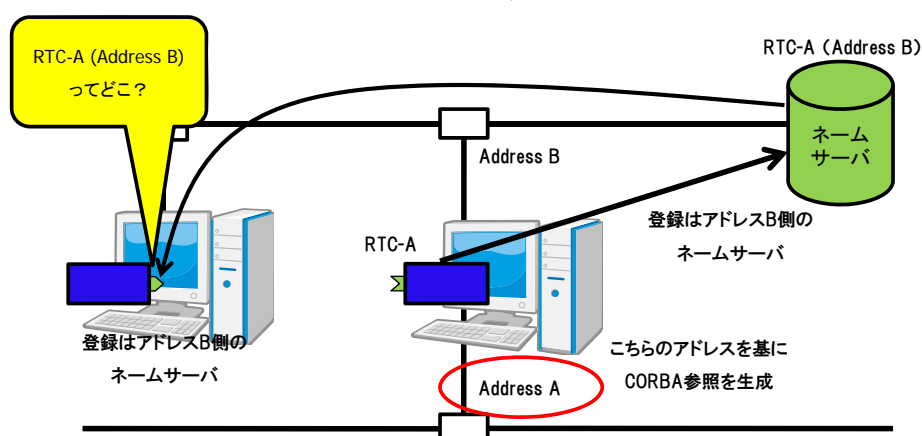
※停止はDeactivateを実行

※RTC間の接続を切る場合には接続線をDelete
もしくは、右クリックメニューから「Delete」を選択



ネームサービスに接続できない場合

■ ネットワークインターフェースが2つある場合



■ RTC.confについて

- RTC起動時の登録先NamingServiceや、登録情報などについて記述

■ 記述例:

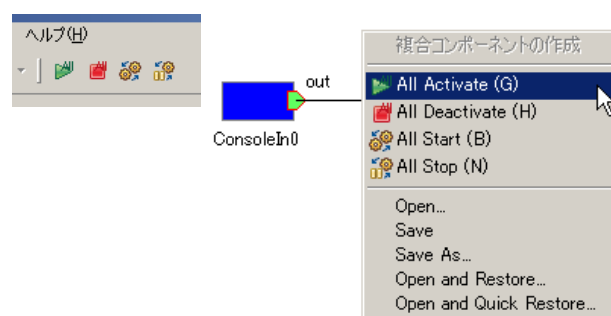
- **corba.nameservers**: localhost:9876
- **naming.formats**: SimpleComponent/%n.rtc
- **corba.endpoints**: 192.168.0.12:

アクション名	説明
Activate	対象RTCを活性化する
Deactivate	対象RTCを非活性化する
Reset	対象RTCをエラー状態からリセットする
Exit	対象RTCの実行主体(ExecutionContext)を停止し、終了する
Start	実行主体(ExecutionContext)の動作を開始する
Stop	実行主体(ExecutionContext)の動作を停止する

■各コンポーネント単位での動作変更



■全コンポーネントの動作を一括変更



※ポップアップメニュー中でのキーバインドを追加

※単独RTCのActivate/Deactivateについては、グローバルはショートカットキー定義を追加

接続プロファイル(DataPort)について

項目	設定内容
Name	接続の名称
DataType	ポート間で送受信するデータの型. ex)TimedOctet, TimedShortなど
InterfaceType	データを送受信するポートの型. ex)corba_cdrなど
DataFlowType	データの送信方法. ex)push, pullなど
SubscriptionType	データ送信タイミング. 送信方法がPushの場合有効 . New, Periodic, Flushから選択
Push Rate	データ送信周期(単位:Hz). SubscriptionTypeがPeriodicの場合のみ有効
Push Policy	データ送信ポリシー. SubscriptionTypeがNew, Periodicの場合のみ有効 . all, fifo, skip, newから選択
Skip Count	送信データスキップ数. Push PolicyがSkipの場合のみ有効

■ SubscriptionType

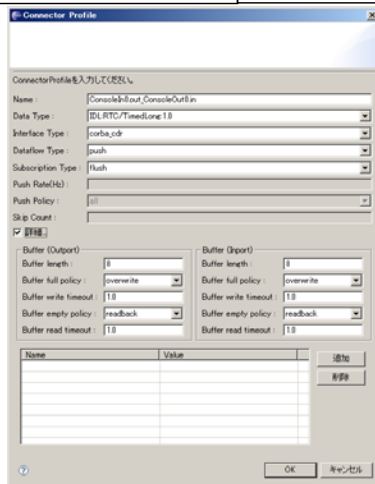
- New : バッファ内に新規データが格納されたタイミングで送信
- Periodic : 一定周期で定期的にデータを送信
- Flush : バッファを介さず即座に同期的に送信

■ Push Policy

- all : バッファ内のデータを一括送信
- fifo : バッファ内のデータをFIFOで1個ずつ送信
- skip : バッファ内のデータを間引いて送信
- new : バッファ内のデータの最新値を送信(古い値は捨てられる)

接続プロファイル(DataPort)について

項目	設定内容
Buffer length	バッファの大きさ
Buffer full policy	データ書き込み時に、バッファフルだった場合の処理。 overwrite, do_nothing, blockから選択
Buffer write timeout	データ書き込み時に、タイムアウトイベントを発生させるまでの時間(単位:秒)
Buffer empty policy	データ読み出し時に、バッファが空だった場合の処理。 readback, do_nothing, blockから選択
Buffer read timeout	データ読み出し時に、タイムアウトイベントを発生させるまでの時間(単位:秒)



- ※OutPort側のバッファ, InPort側のバッファそれぞれに設定可能
- ※timeoutとして「0.0」を設定した場合は、タイムアウトしない

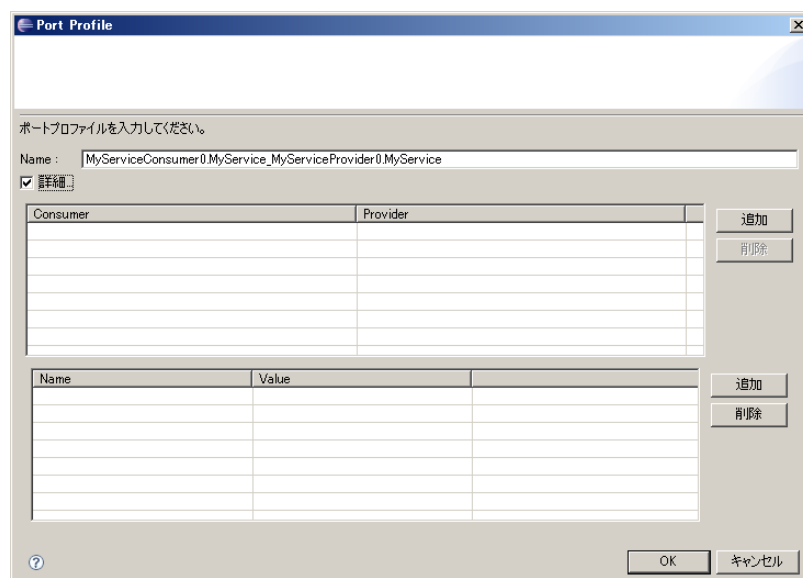
■ Buffer Policy

- overwrite : 上書き
- readback : 最後の要素を再読み出し
- block : ブロック
- do_nothing : なにもしない

- ※Buffer Policy = Block+timeout時間の指定で、一定時間後読み出し/書き込み不可能な場合にタイムアウトを発生させる処理となる

接続プロファイル(ServicePort)について

項目	設定内容
Name	接続の名称
インターフェース情報	接続するインターフェースを設定。 接続対象のServicePortに複数のServiceInterfaceが定義されていた場合、どのインターフェースを実際に接続するかを指定



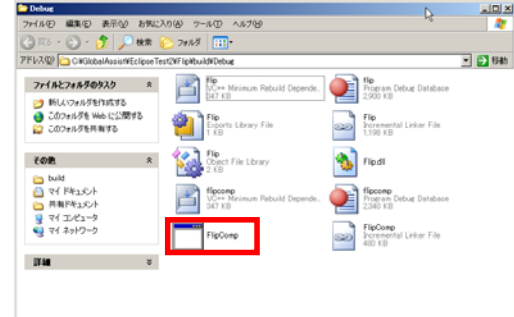
画像処理関連コンポーネントの起動

■ 画像処理用コンポーネントの起動

■ Flipコンポーネントの起動

先ほどコンパイルしたコンポーネントの起動

binaryにて指定したディレクトリ以下のSrc/Debug内のFlipComp.exeを起動



([プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[opencv-rtcs]→ [FlipComp.exe])

■ [スタート]メニューから起動

[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[opencv-rtcs]→ [EdgeComp.exe]

システムの構成

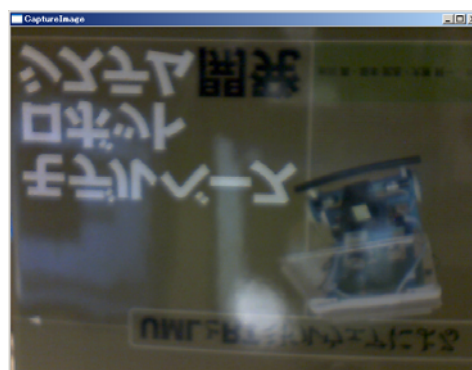
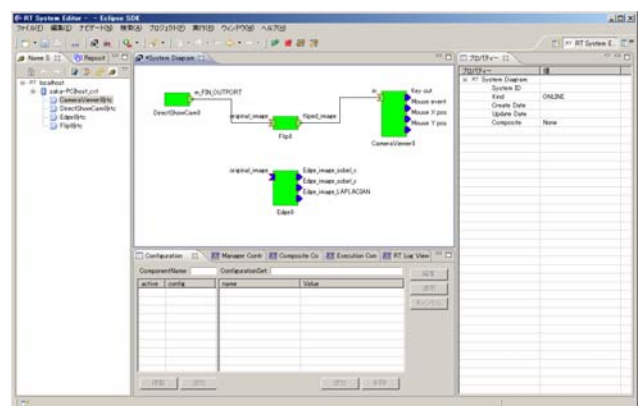
■ Flip側との接続

■ DirectShowCam → Flip

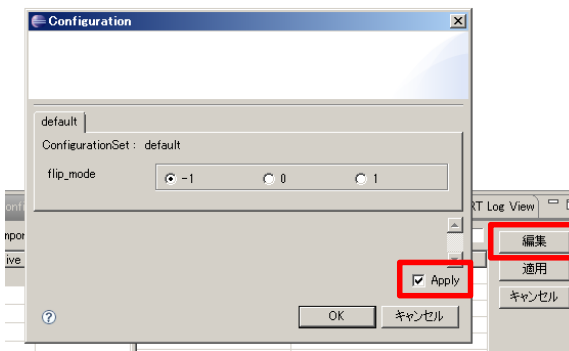
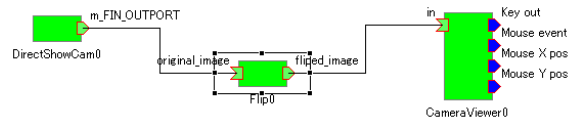
→ CameraViewerと接続

(接続プロファイルはデフォルト設定)

■ AllActivateを実行



コンフィギュレーションの変更

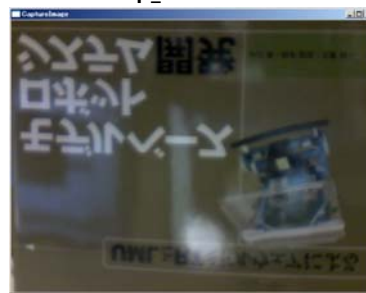


- ConfigurationViewの「編集」
- 表示されたダイアログ内で「flip_mode」の値を変更
- 「Apply」のチェックボックス

flip_mode=1



flip_mode=0

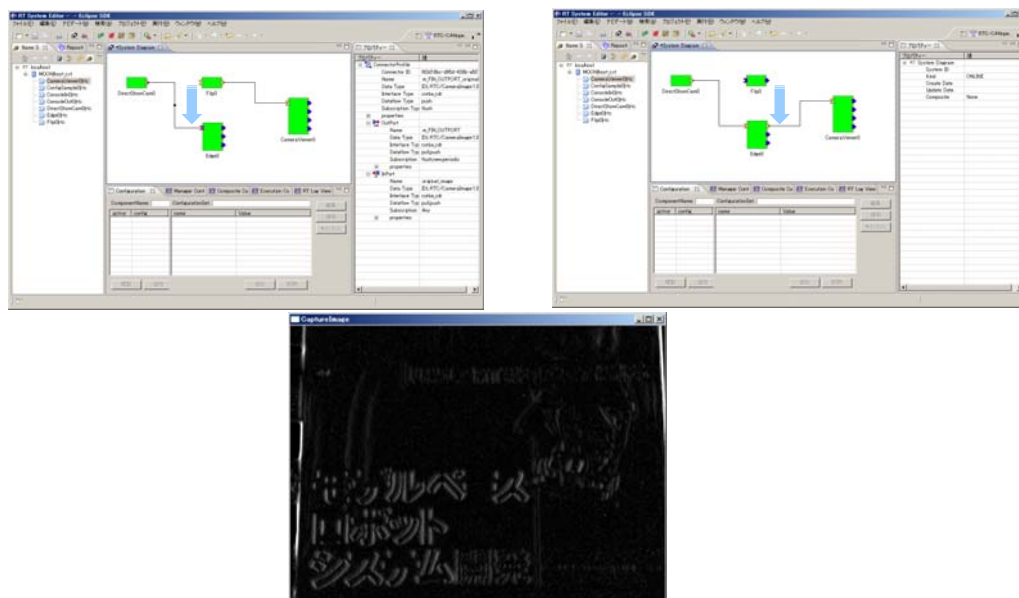


flip_mode=-1



システム構成の変更

- Edge側への差し替え
 - Flipに繋がっている接続線を選択
 - Flip側のPort部分に表示されているハンドルをEdge側のPortに繋ぎ替え
 - 接続プロファイルはデフォルト設定のまま



RTCBuilder補足説明



ドキュメント作成(Windows,CMake利用)



※binaryにて指定したディレクトリ以下のdoc/html/doxygen/html以下にドキュメント



■ 生成されたドキュメントの例

flip 1.0.0

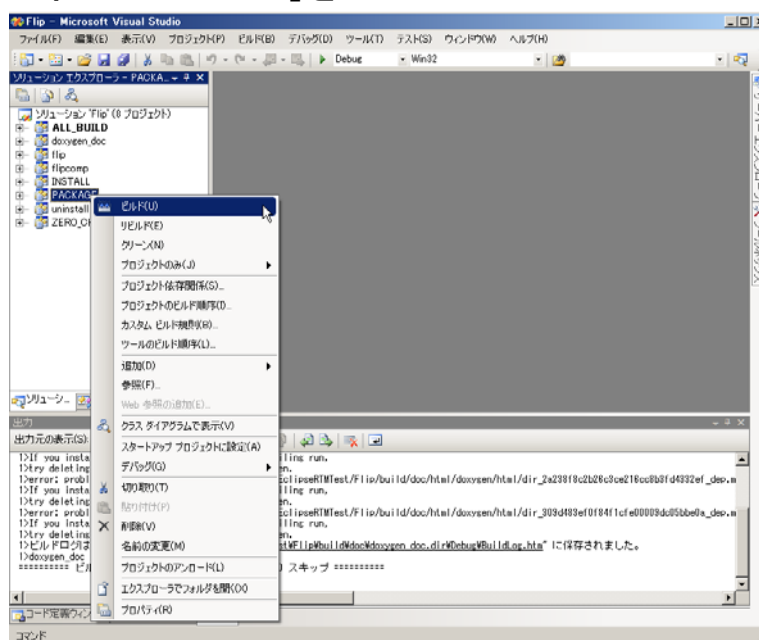
メンバ変数	クラス	ファイル
生成	生成済み	生成済み
クラス Flip		
Flip image component. (314)		
+Flip.h		
すべてのメンバー		
Public メソッド		
Flip (RTC::Manager *manager)		
+Flip ()		
virtual RTC::ReturnCode_t initialize ()		
virtual RTC::ReturnCode_t deactivate (RTC::UniqId ec_id)		
virtual RTC::ReturnCode_t deactivate (RTC::UniqId ec_id)		
virtual RTC::ReturnCode_t deactivate (RTC::UniqId ec_id)		
Protected 変数		
int m_flipMode		
CameraImage m_originImage		
InPort< CameraImage > m_originImageIn		
CameraImage m_flipImage		
OutPort< CameraImage > m_flipImageOut		
説明		
Flip image component.		
Initial値の入力画像を出力画像から読み出すコンポーネント。		
起動時の動作は、RTCの起動シーケンス中に初期化を行い、FlipModeの初期値を設定します。		
FlipModeは、初期値として0で設定されており、起動時に設定されています。		
+ 上下反転した画像: 0		
+ 左右反転した画像: 1		
+ 上下左右反転した画像: -1		
作成するRTCの入出力は、初期化時に設定されます。		

関数
RTC::ReturnCode_t Flip::onActivated (RTC::UniqId ec_id) [virtual]
データ取得の要求 + データ取得の要求の応答 + 出力ポートのデータサイズが異なる
RTC::ReturnCode_t Flip::onDeactivated (RTC::UniqId ec_id) [virtual]
データ取得の要求 + データ取得の要求の応答
RTC::ReturnCode_t Flip::onExecute (RTC::UniqId ec_id) [virtual]
Flip処理 + 初期化シーケンス + 初期化の完了待ち + 初期化の完了待ち + 初期化の完了待ち
RTC::ReturnCode_t Flip::onInitialize () [virtual]
コンポーネント自身の初期化処理
変数
let FlipMode FlipMode (protected)
画像の反転方法を指定するフラグ
Name: FlipMode + Default値: 0 + Unit: 0

flip 1.0.0

メインページ	クラス	ファイル
ファイル一覧		
C:/GlobalAssist/EclipseTest2/Flip/Flip.h		
説明を見る。		
<pre>00001 // * * * * * 00002 #ifndef FLIP_H 00003 #define FLIP_H 00004 00005 #include <rtc/Manager.h> 00006 #include <rtc/DataFlowComponentBase.h> 00007 #include <rtc/DataOutPort.h> 00008 #include <rtc/DataInPort.h> 00009 #include <rtc/DataFlowComponentBase.h> 00010 #include <rtc/DataFlowComponentBase.h> 00011 #include <rtc/DataFlowComponentBase.h> 00012 #include <rtc/DataFlowComponentBase.h> 00013 #include <rtc/DataFlowComponentBase.h> 00014 #include <rtc/DataFlowComponentBase.h> 00015 #include <rtc/DataFlowComponentBase.h> 00016 // Service implementation headers 00017 // c++-template block "service_impl.h" 00018 // c++-template 00019 // c++-template 00020 // Service Consumer stub headers 00021 // c++-template block "consumer_stub.h" 00022 // c++-template 00023 // c++-template 00024 using namespace RTC; 00025 00026 class Flip 00027 { 00028 public: 00029 Flip(RTC::Manager* manager): 00030 { 00031 Flip(); 00032 // c++-template block "public_attribute" 00033 // c++-template 00034 // c++-template 00035 } 00036 // c++-template 00037 // c++-template 00038 // c++-template 00039 // c++-template 00040 // c++-template 00041 // c++-template 00042 // c++-template 00043 // c++-template 00044 // c++-template 00045 // c++-template 00046 // c++-template 00047 // c++-template 00048 // c++-template 00049 // c++-template 00050 // c++-template 00051 // c++-template 00052 // c++-template 00053 // c++-template 00054 // c++-template 00055 // c++-template 00056 // c++-template 00057 // c++-template 00058 // c++-template 00059 // c++-template 00060 // c++-template 00061 // c++-template 00062 // c++-template 00063 // c++-template 00064 // c++-template 00065 // c++-template 00066 // c++-template 00067 // c++-template 00068 // c++-template 00069 // c++-template 00070 // c++-template 00071 // c++-template 00072 // c++-template 00073 // c++-template 00074 // c++-template 00075 // c++-template 00076 // c++-template 00077 // c++-template 00078 // c++-template 00079 // c++-template 00080 // c++-template 00081 // c++-template 00082 // c++-template 00083 // c++-template 00084 // c++-template 00085 // c++-template 00086 // c++-template 00087 // c++-template 00088 // c++-template 00089 // c++-template 00090 // c++-template 00091 // c++-template 00092 // c++-template 00093 // c++-template 00094 // c++-template 00095 // c++-template 00096 // c++-template 00097 // c++-template 00098 // c++-template 00099 // c++-template 00100 // c++-template 00101 // c++-template 00102 // c++-template 00103 // c++-template 00104 // c++-template 00105 // c++-template 00106 // c++-template 00107 // c++-template 00108 // c++-template 00109 // c++-template 00110 // c++-template 00111 // c++-template 00112 // c++-template 00113 // c++-template 00114 // c++-template 00115 // c++-template 00116 // c++-template 00117 // c++-template 00118 // c++-template 00119 // c++-template 00120 // c++-template 00121 // c++-template 00122 // c++-template 00123 // c++-template 00124 // c++-template 00125 // c++-template 00126 // c++-template 00127 // c++-template 00128 // c++-template 00129 // c++-template 00130 // c++-template 00131 // c++-template 00132 // c++-template 00133 // c++-template 00134 // c++-template 00135 // c++-template 00136 // c++-template 00137 // c++-template 00138 // c++-template 00139 // c++-template 00140 // c++-template 00141 // c++-template 00142 // c++-template 00143 // c++-template 00144 // c++-template 00145 // c++-template 00146 // c++-template 00147 // c++-template 00148 // c++-template 00149 // c++-template 00150 // c++-template 00151 // c++-template 00152 // c++-template 00153 // c++-template 00154 // c++-template 00155 // c++-template 00156 // c++-template 00157 // c++-template 00158 // c++-template 00159 // c++-template 00160 // c++-template 00161 // c++-template 00162 // c++-template 00163 // c++-template 00164 // c++-template 00165 // c++-template 00166 // c++-template 00167 // c++-template 00168 // c++-template 00169 // c++-template 00170 // c++-template 00171 // c++-template 00172 // c++-template 00173 // c++-template 00174 // c++-template 00175 // c++-template 00176 // c++-template 00177 // c++-template 00178 // c++-template 00179 // c++-template 00180 // c++-template 00181 // c++-template 00182 // c++-template 00183 // c++-template 00184 // c++-template 00185 // c++-template 00186 // c++-template 00187 // c++-template 00188 // c++-template 00189 // c++-template 00190 // c++-template 00191 // c++-template 00192 // c++-template 00193 // c++-template 00194 // c++-template 00195 // c++-template 00196 // c++-template 00197 // c++-template 00198 // c++-template 00199 // c++-template 00200 // c++-template 00201 // c++-template 00202 // c++-template 00203 // c++-template 00204 // c++-template 00205 // c++-template 00206 // c++-template 00207 // c++-template 00208 // c++-template 00209 // c++-template 00210 // c++-template 00211 // c++-template 00212 // c++-template 00213 // c++-template 00214 // c++-template 00215 // c++-template 00216 // c++-template 00217 // c++-template 00218 // c++-template 00219 // c++-template 00220 // c++-template 00221 // c++-template 00222 // c++-template 00223 // c++-template 00224 // c++-template 00225 // c++-template 00226 // c++-template 00227 // c++-template 00228 // c++-template 00229 // c++-template 00230 // c++-template 00231 // c++-template 00232 // c++-template 00233 // c++-template 00234 // c++-template 00235 // c++-template 00236 // c++-template 00237 // c++-template 00238 // c++-template 00239 // c++-template 00240 // c++-template 00241 // c++-template 00242 // c++-template 00243 // c++-template 00244 // c++-template 00245 // c++-template 00246 // c++-template 00247 // c++-template 00248 // c++-template 00249 // c++-template 00250 // c++-template 00251 // c++-template 00252 // c++-template 00253 // c++-template 00254 // c++-template 00255 // c++-template 00256 // c++-template 00257 // c++-template 00258 // c++-template 00259 // c++-template 00260 // c++-template 00261 // c++-template 00262 // c++-template 00263 // c++-template 00264 // c++-template 00265 // c++-template 00266 // c++-template 00267 // c++-template 00268 // c++-template 00269 // c++-template 00270 // c++-template 00271 // c++-template 00272 // c++-template 00273 // c++-template 00274 // c++-template 00275 // c++-template 00276 // c++-template 00277 // c++-template 00278 // c++-template 00279 // c++-template 00280 // c++-template 00281 // c++-template 00282 // c++-template 00283 // c++-template 00284 // c++-template 00285 // c++-template 00286 // c++-template 00287 // c++-template 00288 // c++-template 00289 // c++-template 00290 // c++-template 00291 // c++-template 00292 // c++-template 00293 // c++-template 00294 // c++-template 00295 // c++-template 00296 // c++-template 00297 // c++-template 00298 // c++-template 00299 // c++-template 00300 // c++-template 00301 // c++-template 00302 // c++-template 00303 // c++-template 00304 // c++-template 00305 // c++-template 00306 // c++-template 00307 // c++-template 00308 // c++-template 00309 // c++-template 00310 // c++-template 00311 // c++-template 00312 // c++-template 00313 // c++-template 00314 // c++-template 00315 // c++-template 00316 // c++-template 00317 // c++-template 00318 // c++-template 00319 // c++-template 00320 // c++-template 00321 // c++-template 00322 // c++-template 00323 // c++-template 00324 // c++-template 00325 // c++-template 00326 // c++-template 00327 // c++-template 00328 // c++-template 00329 // c++-template 00330 // c++-template 00331 // c++-template 00332 // c++-template 00333 // c++-template 00334 // c++-template 00335 // c++-template 00336 // c++-template 00337 // c++-template 00338 // c++-template 00339 // c++-template 00340 // c++-template 00341 // c++-template 00342 // c++-template 00343 // c++-template 00344 // c++-template 00345 // c++-template 00346 // c++-template 00347 // c++-template 00348 // c++-template 00349 // c++-template 00350 // c++-template 00351 // c++-template 00352 // c++-template 00353 // c++-template 00354 // c++-template 00355 // c++-template 00356 // c++-template 00357 // c++-template 00358 // c++-template 00359 // c++-template 00360 // c++-template 00361 // c++-template 00362 // c++-template 00363 // c++-template 00364 // c++-template 00365 // c++-template 00366 // c++-template 00367 // c++-template 00368 // c++-template 00369 // c++-template 00370 // c++-template 00371 // c++-template 00372 // c++-template 00373 // c++-template 00374 // c++-template 00375 // c++-template 00376 // c++-template 00377 // c++-template 00378 // c++-template 00379 // c++-template 00380 // c++-template 00381 // c++-template 00382 // c++-template 00383 // c++-template 00384 // c++-template 00385 // c++-template 00386 // c++-template 00387 // c++-template 00388 // c++-template 00389 // c++-template 00390 // c++-template 00391 // c++-template 00392 // c++-template 00393 // c++-template 00394 // c++-template 00395 // c++-template 00396 // c++-template 00397 // c++-template 00398 // c++-template 00399 // c++-template 00400 // c++-template 00401 // c++-template 00402 // c++-template 00403 // c++-template 00404 // c++-template 00405 // c++-template 00406 // c++-template 00407 // c++-template 00408 // c++-template 00409 // c++-template 00410 // c++-template 00411 // c++-template 00412 // c++-template 00413 // c++-template 00414 // c++-template 00415 // c++-template 00416 // c++-template 00417 // c++-template 00418 // c++-template 00419 // c++-template 00420 // c++-template 00421 // c++-template 00422 // c++-template 00423 // c++-template 00424 // c++-template 00425 // c++-template 00426 // c++-template 00427 // c++-template 00428 // c++-template 00429 // c++-template 00430 // c++-template 00431 // c++-template 00432 // c++-template 00433 // c++-template 00434 // c++-template 00435 // c++-template 00436 // c++-template 00437 // c++-template 00438 // c++-template 00439 // c++-template 00440 // c++-template 00441 // c++-template 00442 // c++-template 00443 // c++-template 00444 // c++-template 00445 // c++-template 00446 // c++-template 00447 // c++-template 00448 // c++-template 00449 // c++-template 00450 // c++-template 00451 // c++-template 00452 // c++-template 00453 // c++-template 00454 // c++-template 00455 // c++-template 00456 // c++-template 00457 // c++-template 00458 // c++-template 00459 // c++-template 00460 // c++-template 00461 // c++-template 00462 // c++-template 00463 // c++-template 00464 // c++-template 00465 // c++-template 00466 // c++-template 00467 // c++-template 00468 // c++-template 00469 // c++-template 00470 // c++-template 00471 // c++-template 00472 // c++-template 00473 // c++-template 00474 // c++-template 00475 // c++-template 00476 // c++-template 00477 // c++-template 00478 // c++-template 00479 // c++-template 00480 // c++-template 00481 // c++-template 00482 // c++-template 00483 // c++-template 00484 // c++-template 00485 // c++-template 00486 // c++-template 00487 // c++-template 00488 // c++-template 00489 // c++-template 00490 // c++-template 00491 // c++-template 00492 // c++-template 00493 // c++-template 00494 // c++-template 00495 // c++-template 00496 // c++-template 00497 // c++-template 00498 // c++-template 00499 // c++-template 00500 // c++-template 00501 // c++-template 00502 // c++-template 00503 // c++-template 00504 // c++-template 00505 // c++-template 00506 // c++-template 00507 // c++-template 00508 // c++-template 00509 // c++-template 00510 // c++-template 00511 // c++-template 00512 // c++-template 00513 // c++-template 00514 // c++-template 00515 // c++-template 00516 // c++-template 00517 // c++-template 00518 // c++-template 00519 // c++-template 00520 // c++-template 00521 // c++-template 00522 // c++-template 00523 // c++-template 00524 // c++-template 00525 // c++-template 00526 // c++-template 00527 // c++-template 00528 // c++-template 00529 // c++-template 00530 // c++-template 00531 // c++-template 00532 // c++-template 00533 // c++-template 00534 // c++-template 00535 // c++-template 00536 // c++-template 00537 // c++-template 00538 // c++-template 00539 // c++-template 00540 // c++-template 00541 // c++-template 00542 // c++-template 00543 // c++-template 00544 // c++-template 00545 // c++-template 00546 // c++-template 00547 // c++-template 00548 // c++-template 00549 // c++-template 00550 // c++-template 00551 // c++-template 00552 // c++-template 00553 // c++-template 00554 // c++-template 00555 // c++-template 00556 // c++-template 00557 // c++-template 00558 // c++-template 00559 // c++-template 00560 // c++-template 00561 // c++-template 00562 // c++-template 00563 // c++-template 00564 // c++-template 00565 // c++-template 00566 // c++-template 00567 // c++-template 00568 // c++-template 00569 // c++-template 00570 // c++-template 00571 // c++-template 00572 // c++-template 00573 // c++-template 00574 // c++-template 00575 // c++-template 00576 // c++-template 00577 // c++-template 00578 // c++-template 00579 // c++-template 00580 // c++-template 00581 // c++-template 00582 // c++-template 00583 // c++-template 00584 // c++-template 00585 // c++-template 00586 // c++-template 00587 // c++-template 00588 // c++-template 00589 // c++-template 00590 // c++-template 00591 // c++-template 00592 // c++-template 00593 // c++-template 00594 // c++-template 00595 // c++-template 00596 // c++-template 00597 // c++-template 00598 // c++-template 00599 // c++-template 00600 // c++-template 00601 // c++-template 00602 // c++-template 00603 // c++-template 00604 // c++-template 00605 // c++-template 00606 // c++-template 00607 // c++-template 00608 // c++-template 00609 // c++-template 00610 // c++-template 00611 // c++-template 00612 // c++-template 00613 // c++-template 00614 // c++-template 00615 // c++-template 00616 // c++-template 00617 // c++-template 00618 // c++-template 00619 // c++-template 00620 // c++-template 00621 // c++-template 00622 // c++-template 00623 // c++-template 00624 // c++-template 00625 // c++-template 00626 // c++-template 00627 // c++-template 00628 // c++-template 00629 // c++-template 00630 // c++-template 00631 // c++-template 00632 // c++-template 00633 // c++-template 00634 // c++-template 00635 // c++-template 00636 // c++-template 00637 // c++-template 00638 // c++-template 00639 // c++-template 00640 // c++-template 00641 // c++-template 00642 // c++-template 00643 // c++-template 00644 // c++-template 00645 // c++-template 00646 // c++-template 00647 // c++-template 00648 // c++-template 00649 // c++-template 00650 // c++-template 00651 // c++-template 00652 // c++-template 00653 // c++-template 00654 // c++-template 00655 // c++-template 00656 // c++-template 00657 // c++-template 00658 // c++-template 00659 // c++-template 00660 // c++-template 00661 // c++-template 00662 // c++-template 00663 // c++-template 00664 // c++-template 00665 // c++-template 00666 // c++-template 00667 // c++-template 00668 // c++-template 00669 // c++-template 00670 // c++-template 00671 // c++-template 00672 // c++-template 00673 // c++-template 00674 // c++-template 00675 // c++-template 00676 // c++-template 00677 // c++-template 00678 // c++-template 00679 // c++-template 00680 // c++-template 00681 // c++-template 00682 // c++-template 00683 // c++-template 00684 // c++-template 00685 // c++-template 00686 // c++-template 00687 // c++-template 00688 // c++-template 00689 // c++-template 00690 // c++-template 00691 // c++-template 00692 // c++-template 00693 // c++-template 00694 // c++-template 00695 // c++-template 00696 // c++-template 00697 // c++-template 00698 // c++-template 00699 // c++-template 00700 // c++-template 00701 // c++-template 00702 // c++-template 00703 // c++-template 00704 // c++-template 00705 // c++-template 00706 // c++-template 00707 // c++-template 00708 // c++-template 00709 // c++-template 00710 // c++-template 00711 // c++-template 00712 // c++-template 00713 // c++-template 00714 // c++-template 00715 // c++-template 00716 // c++-template 00717 // c++-template 00718 // c++-template 00719 // c++-template 00720 // c++-template 00721 // c++-template 00722 // c++-template 00723 // c++-template 00724 // c++-template 00725 // c++-template 00726 // c++-template 00727 // c++-template 00728 // c++-template 00729 // c++-template 00730 // c++-template 00731 // c++-template 00732 // c++-template 00733 // c++-template 00734 // c++-template 00735 // c++-template 00736 // c++-template 00737 // c++-template 00738 // c++-template 00739 // c++-template 00740 // c++-template 00741 // c++-template 00742 // c++-template 00743 // c++-template 00744 // c++-template 00745 // c++-template 00746 // c++-template 00747 // c++-template 00748 // c++-template 00749 // c++-template 00750 // c++-template 00751 // c++-template 00752 // c++-template 00753 // c++-template 00754 // c++-template 00755 // c++-template 00756 // c++-template 00757 // c++-template 00758 // c++-template 00759 // c++-template 00760 // c++-template 00761 // c++-template 00762 // c++-template 00763 // c++-template 00764 // c++-template 00765 // c++-template 00766 // c++-template 00767 // c++-template 00768 // c++-template 00769 // c++-template 00770 // c++-template 00771 // c++-template 00772 // c++-template 00773 // c++-template 00774 // c++-template 00775 // c++-template 00776 // c++-template 00777 // c++-template 00778 // c++-template 00779 // c++-template 00780 // c++-template 00781 // c++-template 00782 // c++-template 00783 // c++-template 00784 // c++-template 00785 // c++-template 00786 // c++-template 00787 // c++-template 00788 // c++-template 00789 // c++-template 00790 // c++-template 00791 // c++-template 00792 // c++-template 00793 // c++-template 00794 // c++-template 00795 // c++-template 00796 // c++-template 00797 // c++-template 00798 // c++-template 00799 // c++-template 00800 // c++-template 00801 // c++-template 00802 // c++-template 00803 // c++-template 00804 // c++-template 00805 // c++-template 00806 // c++-template 00807 // c++-template 00808 // c++-template 00809 // c++-template 00810 // c++-template 00811 // c++-template 00812 // c++-template 00813 // c++-template 00814 // c++-template 00815 // c++-template 00816 // c++-template 00817 // c++-template 00818 // c++-template 00819 // c++-template 00820 // c++-template 00821 // c++-template 00822 // c++-template 00823 // c++-template 00824 // c++-template 00825 // c++-template 00826 // c++-template 00827 // c++-template 00828 // c++-template 00829 // c++-template 00830 // c++-template 00831 // c++-template 00832 // c++-template 00833 // c++-template 00834 // c++-template 00835 // c++-template 00836 // c++-template 00837 // c++-template 00838 // c++-template 00839 // c++-template 00840 // c++-template 00841 // c++-template 00842 // c++-template 00843 // c++-template 00844 // c++-template 00845 // c++-template 00846 // c++-template 00847 // c++-template 00848 // c++-template 00849 // c++-template 00850 // c++-template 00851 // c++-template 00852 // c++-template 00853 // c++-template 00854 // c++-template 00855 // c++-template 00856 // c++-template 00857 // c++-template 00858 // c++-template 00859 // c++-template 00860 // c++-template 00861 // c++-template 00862 // c++-template 00863 // c++-template 00864 // c++-template 00865 // c++-template 00866 // c++-template 00867 // c++-template 00868 // c++-template 00869 // c++-template 00870 // c++-template 00871 // c++-template 00872 // c++-template 00873 // c++-template 00874 // c++-template 00875 // c++-template 00876 // c++-template 00877 // c++-template 00878 // c++-template 00879 // c++-template 00880 // c++-template 00881 // c++-template 00882 // c++-template 00883 // c++-template 00884 // c++-template 00885 // c++-template 00886 // c++-template 00887 // c++-template 00888 // c++-template 00889 // c++-template 00890 // c++-template 00891 // c++-template 00892 // c++-template 00893 // c++-template 00894 // c++-template 00895 // c++-template 00896 // c++-template 00897 // c++-template 00898 // c++-template 00899 // c++-template 00900 // c++-template 00901 // c++-template 00902 // c++-template 00903 // c++-template 00904 // c++-template 00905 // c++-template 00906 // c++-template 00907 // c++-template 00908 // c++-template 00909 // c++-template 00910 // c++-template 00911 // c++-template 00912 // c++-template 00913 // c++-template 00914 // c++-template 00915 // c++-template 00916 // c++-template 00917 // c++-template 00918 // c++-template 00919 // c++-template 00920 // c++-template 00921 // c++-template 00922 // c++-template 00923 // c++-template 00924 // c++-template 00925 // c++-template 00926 // c++-template 00927 // c++-template 00928 // c++-template 00929 // c++-template 00930 // c++-template 00931 // c++-template 00932 // c++-template 00933 // c++-template 00934 // c++-template 00935 // c++-template 00936 // c++-template 00937 // c++-template 00938 // c++-template 00939 // c++-template 00940 // c++-template 00941 // c++-template 00942 // c++-template 00943 // c++-template 00944 // c++-template 00945 // c++-template 00946 // c++-template 00947 // c++-template 00948 // c++-template 00949 // c++-template 00950 // c++-template 00951 // c++-template 00952 // c++-template 00953 // c++-template 00954 // c++-template 00955 // c++-template 00956 // c++-template 00957 // c++-template 00958 // c++-template 00959 // c++-template 00960 // c++-template 00961 // c++-template 00962 // c++-template 00963 // c++-template 00964 // c++-template 00965 // c++-template 00966 // c++-template 00967 // c++-template 00968 // c++-template 00969 // c++-template 00970 // c++-template 00971 // c++-template 00972 // c++-template 00973 // c++-template 00974 // c++-template 00975 // c++-template 00976 // c++-template 00977 // c++-template 00978 // c++-template 00979 // c++-template 00980 // c++-template 00981 // c++-template 00982 // c++-template 00983 // c++-template 00984 // c++-template 00985 // c++-template 00986 // c++-template 00987 // c++-template 00988 // c++-template 00989 // c++-template 00990 // c++-template 00991 // c++-template 00992 // c++-template 00993 // c++-template 00994 // c++-template 00995 // c++-template 00996 // c++-template 00997 // c++-template 00998 // c++-template 00999 // c++-template 01000 // c++-template 01001 // c++-template 01002 // c++-template 01003 // c++-template 01004 // c++-template 01005 // c++-template 01006 // c++-template 01007 // c++-template 01008 // c++-template 01009 // c++-template 01010 // c++-template 01011 // c++-template 01012 // c++-template 01013 // c++-template 01014 // c++-template 01015 // c++-template 01016 // c++-template 01017 // c++-template 01018 // c++-template 01019 // c++-template 01020 // c++-template 01021 // c++-template </pre>		

■ ソリューション中の「PACKAGE」をビルド



- binaryにて指定したディレクトリ直下にmsi形式のインストールパッケージを生成
- コンポーネントのインストール先
C:\Program Files\OpenRTM-aist\1.1\components\言語>/<パッケージ名>

各種設定

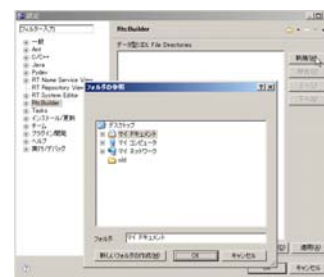
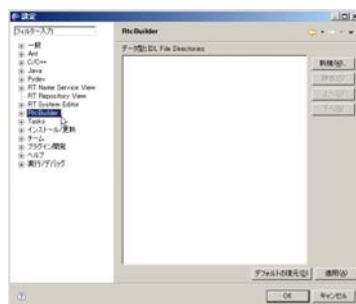
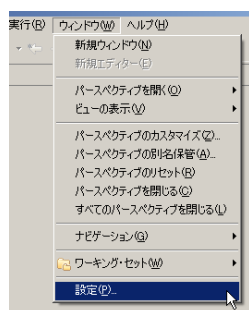
■ DataPortにて利用するデータ型の指定

→データ型を定義したIDLファイルが格納されているディレクトリを指定

①メニューから
「ウインドウ」→「設定」

②「RtcBuilder」を選択

③「新規」ボタンにて表示される
ディレクトリ選択ダイアログ
にて場所を指定



※独自に定義したデータ型を使用する場合のみ必要な設定

OpenRTM-aistにて標準で用意されている型のみを使用する場合には設定不要

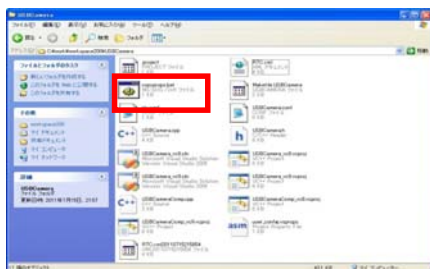
・標準型の定義内容格納位置：[RTM_Root]rtm/idl

→BasicDataType.idl, ExtendedDataTypes.idlなど

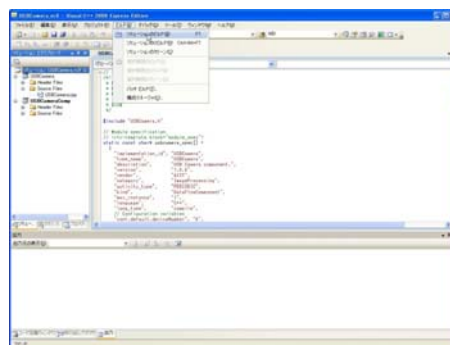
→デフォルト設定では, [RTM_Root]=C:/Program Files/OpenRTM-aist/1.1/

コンパイル・実行(Windows)

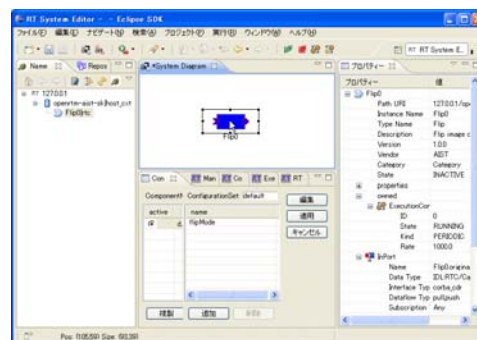
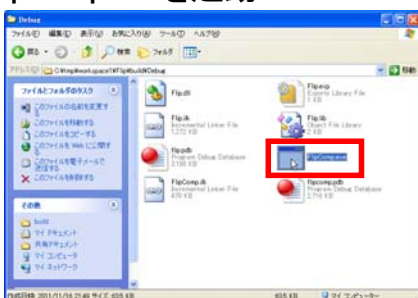
- ①コード生成先ディレクトリ内の「copyprops.bat」をダブルクリックして、設定ファイルをコピー



- ②VisualStudioを用いたビルド

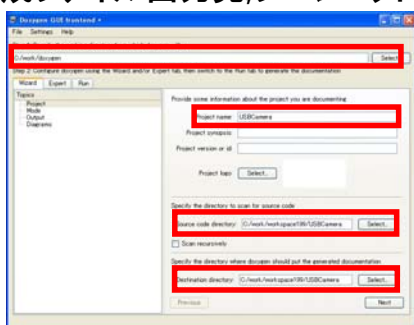


- ③FlipComp¥¥Debug内のFlipComp.exeを起動

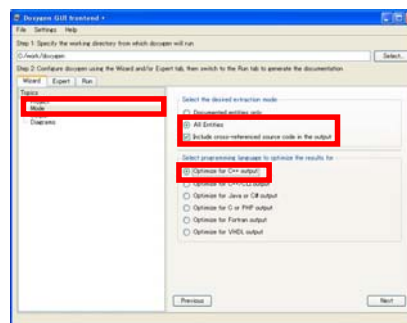


ドキュメント作成(Windows)

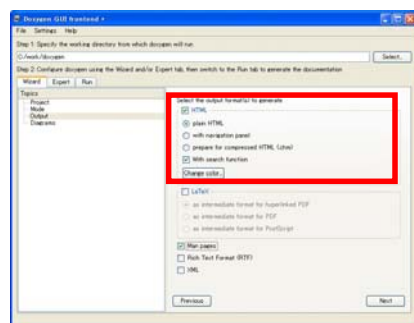
- ①Doxygen用GUIツールを起動
作業用ディレクトリ,ソース格納場所,
生成ファイル出力先,プロジェクト名を指定



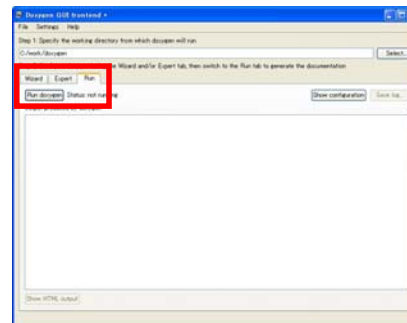
- ②「Mode」セクションにて、
出力内容,使用言語を指定



- ③「Output」セクションにて、html出力を指定



- ③「Run」タブにて、「Run doxygen」を実行



RTSystemEditor補足説明



既存コンポーネントの再利用

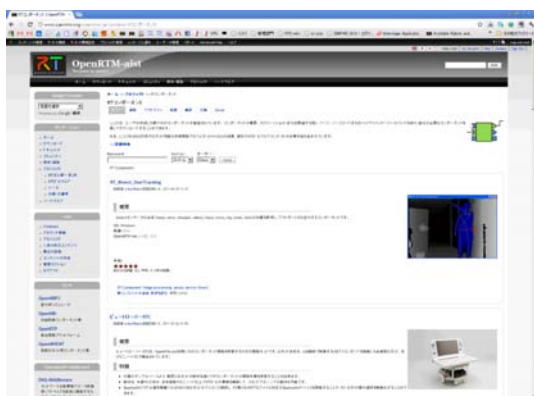


■ プロジェクトとは

- ユーザが作成した様々なコンポーネントやツールの公開場所
- ユーザ登録すれば、誰でも自分の成果物の紹介ページを作成可能
- 他のユーザに自分のコンポーネント等を紹介することができる

■ プロジェクトのカテゴリ

- RTコンポーネント: 1つのコンポーネントまたは複数のコンポーネント群などが登録されています。
- RTミドルウェア: OpenRTM-aistや他のミドルウェア、ミドルウェア拡張モジュール等が登録されています。
- ツール: 各種ツール(RTSystemEditorやrtshellを含む)ツールはこのカテゴリになります。
- 関連ドキュメント: 関連ドキュメントとは、各種インターフェースの仕様書やマニュアル等を含みます。



タイプ	登録数
RTコンポーネント群	638
RTミドルウェア	29
ツール	39
仕様・文書	4
ハードウェア	30

既存コンポーネントの再利用

■ プロジェクトから対象コンポーネントを取得

■ 「顔検出コンポーネント」

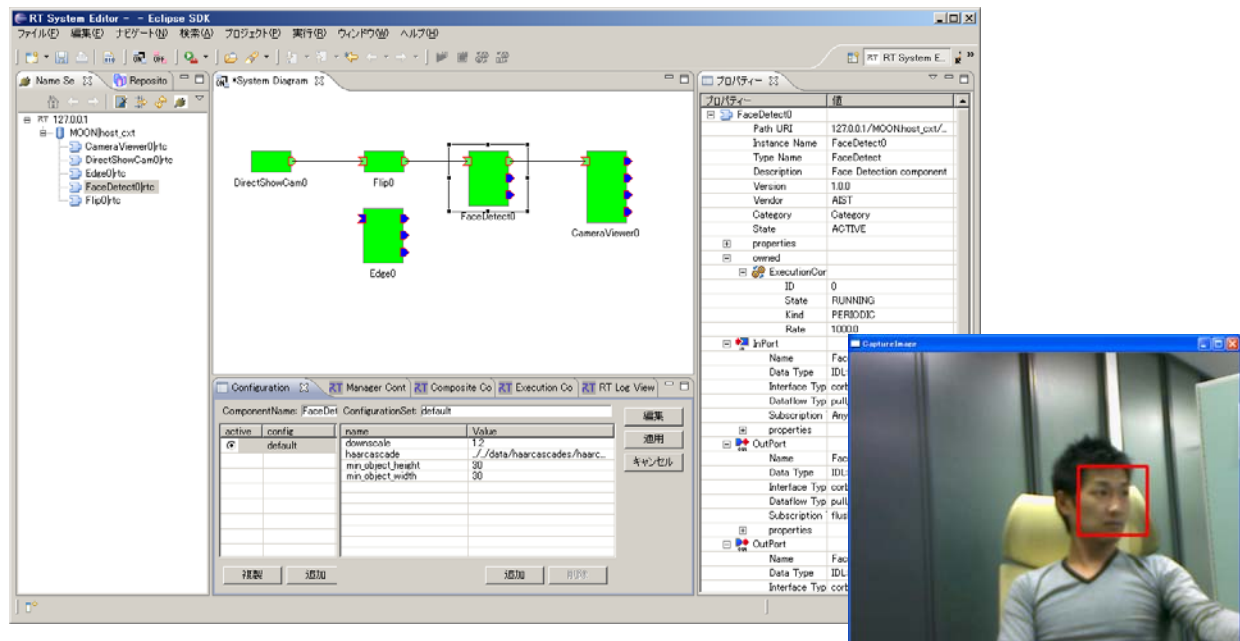
<http://www.openrtm.org/openrtm/ja/project/facedetect>

対象コンポーネントをダウンロード



既存コンポーネントの再利用

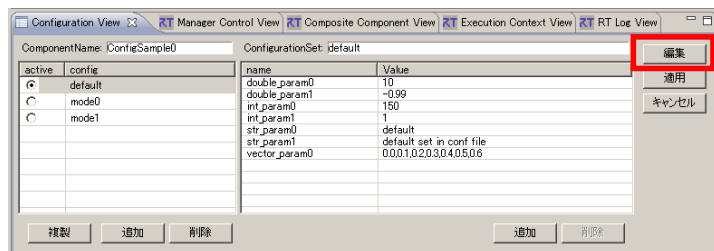
- ダウンロードしたファイル(FaceDetect.zip)を解凍
- 解凍したディレクトリ内の以下のファイルを実行し、システムエディタ上に配置
\$(FaceDetect_Root)/build/Release/FaceDetectComp.exe



ネットワーク上の他のRTCとの接続

- IPアドレスの確認
 - スタートメニュー中の「全てのプログラム」-「アクセサリ」-「コマンドプロンプト」
 - コマンド「ipconfig」を実行
- 他PC上で動作するRTCとの接続
 - 隣の方のIPアドレスを聞く
 - RTSystemEditorの「ネームサーバを追加(コンセントのアイコン)」をクリックして、上記のIPアドレスを入力する
 - 隣の方のネームサーバ内の階層化にあるDirectShowCamをシステムエディタにDnDする
 - 上記でDnDしたDirectShowCamと自分のPC上で起動したCameraViewerのデータポートを接続する

■ RTコンポーネントのコンフィギュレーション情報の確認/編集



- ※「編集」ボタンにより、各種コントロールを用いた一括編集が可能
- ※「Apply」チェックボックスがONの場合、設定値を変更すると即座にコンポーネントに反映
→テキストボックスからフォーカス外れる、ラジオボタンを選択する、スライダーを操作する、スピナを変更する、などのタイミング
- ※コンフィギュレーション情報を複数保持している場合、上部のタブで編集対象を切り替え

コンフィギュレーション情報の設定方法

● rtc.conf内

[カテゴリ名]. [コンポーネント名]. config_file: [コンフィギュレーションファイル名]

※例) example.ConfigSample.config_file: configsample.conf

● コンフィギュレーションファイル内

● コンフィギュレーション情報

conf. [コンフィグセット名]. [コンフィグパラメータ名] : [デフォルト値]

※例) conf.mode0.int_param0: 123

● Widget情報

conf. __widget__. [コンフィグパラメータ名] : [Widget名]

※例) conf.__widget__.str_param0: radio

● 制約情報

conf. __constraints__. [コンフィグパラメータ名] : [制約情報]

※例) conf.__constraints__.str_param0: (bar,foo,foo,dara)

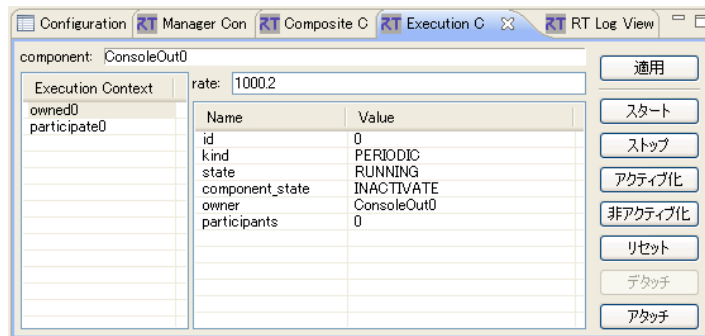
conf. __[コンフィグセット名]. [コンフィグパラメータ名] : [制約情報]

※例) conf._mode1.str_param0: (bar2,foo2,dara2)

RTCの利用者が設定するのではなく、RTC開発者、RTC管理者が設定することを想定。

RTCBUILDERを使用することで設定可能

■ RTコンポーネントが属する実行コンテキスト(EC)を一覧表示

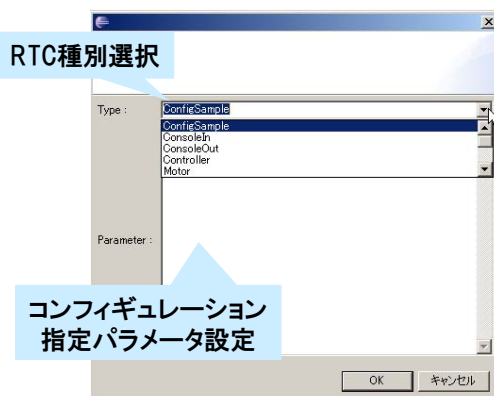
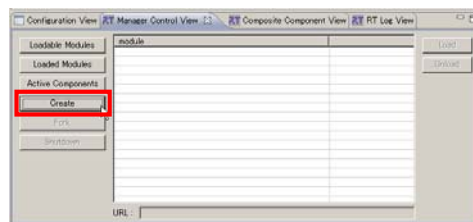


属性名	説明
id	ECのID. オンラインの場合には, context_handleを表示
kind	ECの種別(PERIODIC/EVENT_DRIVEN/OTHER)
state	ECの状態(RUNNING/STOPPING)
component state	対象RTCの状態(ACTIVE/INACTIVE/ERROR)
owner	対象ECを所有しているオーナーRTCのインスタンス名
participants	対象ECに参加中のRTCの数

※対象ECの実行周期の変更, EC自身の動作開始/終了, 新規RTCへのアタッチ, アタッチ済みRTCのデタッチも可能

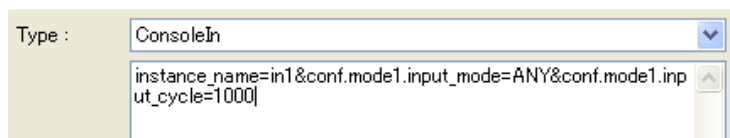
マネージャビュー

■ RTコンポーネントの新規インスタンスの生成

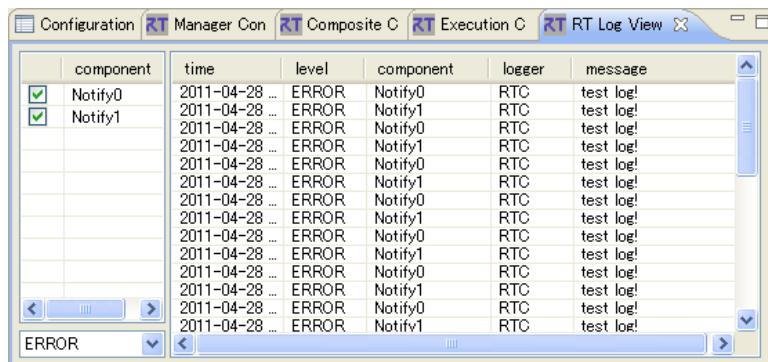


● コンフィギュレーション指定パラメータ

- conf. [ConfigSet名]. [Configパラメータ名]=[設定値]の形式にてConfigurationSetの値も設定可能

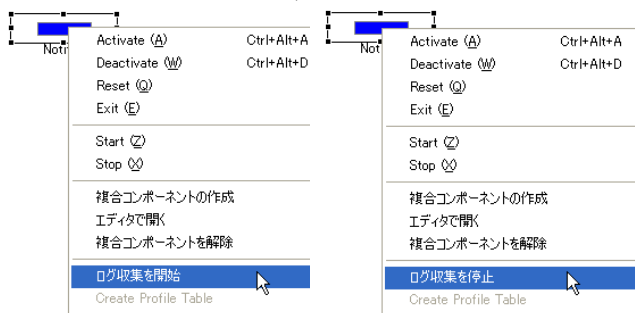


■ 選択したRTCから収集したログ情報を一覧表示

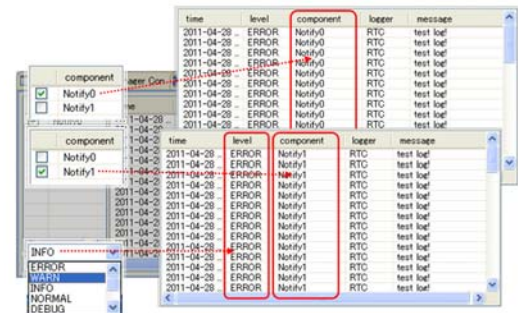


※近日機能追加予定

● ログ収集の開始/停止



● ログ情報のフィルタリング

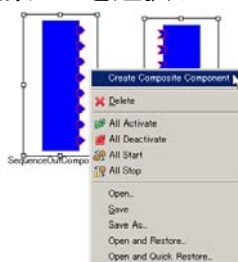


複合コンポーネント

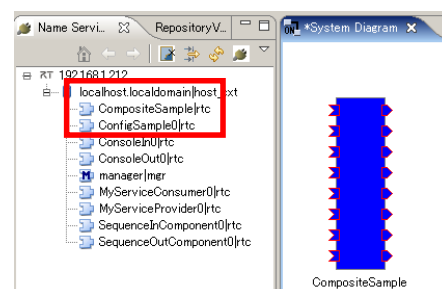
■ 複数のRTCをまとめて、1つのRTCとして扱うための仕組み

● 複合コンポーネントの作成方法

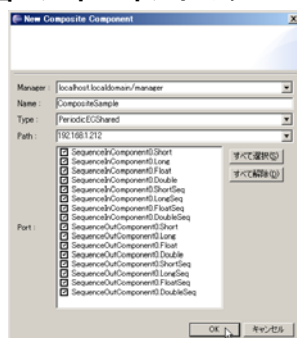
① 複数RTCを選択している状態で右クリック



③ 複合コンポーネントを生成



② 複合コンポーネントのプロパティを設定



項目	設定内容
Manager	複合コンポーネントを制御するマネージャを選択
Name	複合コンポーネントのインスタンス名を入力
Type	複合コンポーネントの型を選択
Path	複合コンポーネントのパスを入力
Port	外部に公開するポートを選択

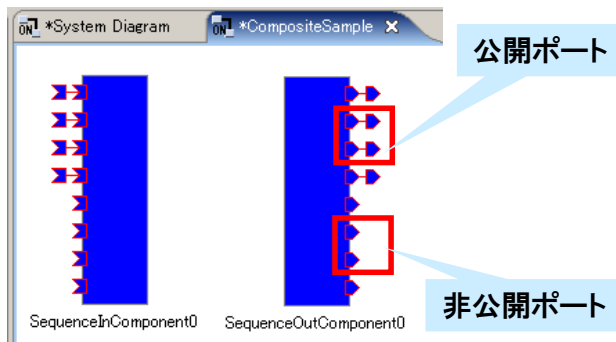
※生成対象複合コンポーネント外部と接続されているPortは強制的に公開されます

■ 複合コンポーネントのタイプについて

タイプ名	説明
PeriodicECShared	実行主体であるExecutionContextのみを共有. 各子コンポーネントはそれぞれの状態を持つ
PeriodicStateShared	実行主体であるExecutionContextと状態を共有
Grouping	便宜的にツール上のみでグループ化

■ 複合コンポーネントエディタ

- 複合コンポーネントをダブルクリックすることで表示



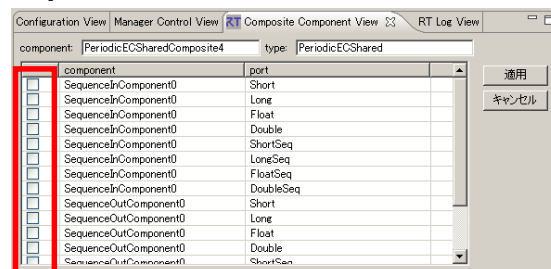
- ※エディタ内に別RTCをDnDすることで、子コンポーネントの追加が可能
→ 追加したRTCのポートは全て非公開に設定
- ※エディタ内のRTCを削除することで、子コンポーネントの削除が可能
→ 削除されたRTCは、親エディタに表示

複合コンポーネント

■ 公開ポートの設定

- 複合コンポーネントビュー

ポート公開情報

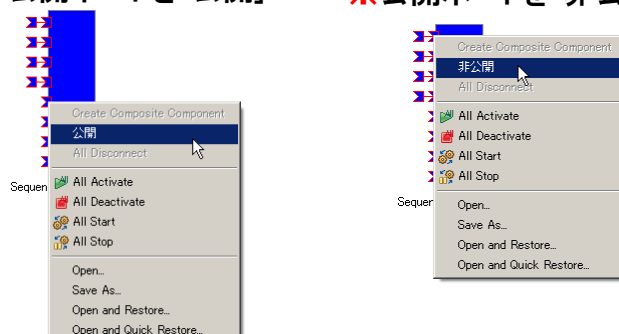


- ※ポート公開情報を変更し、「適用」をクリック

- 複合コンポーネントエディタ

※非公開ポートを「公開」

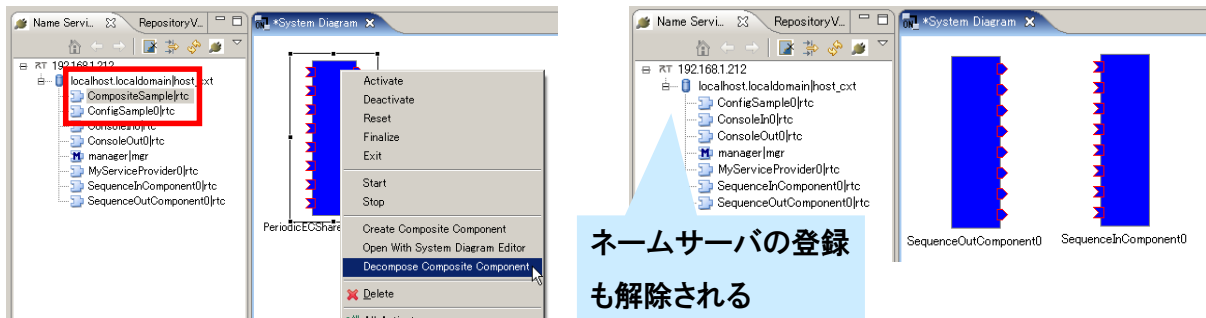
※公開ポートを「非公開」



外部コンポーネントと接続されているポートを「非公開」に設定することはできません

■ 複合コンポーネントの解除

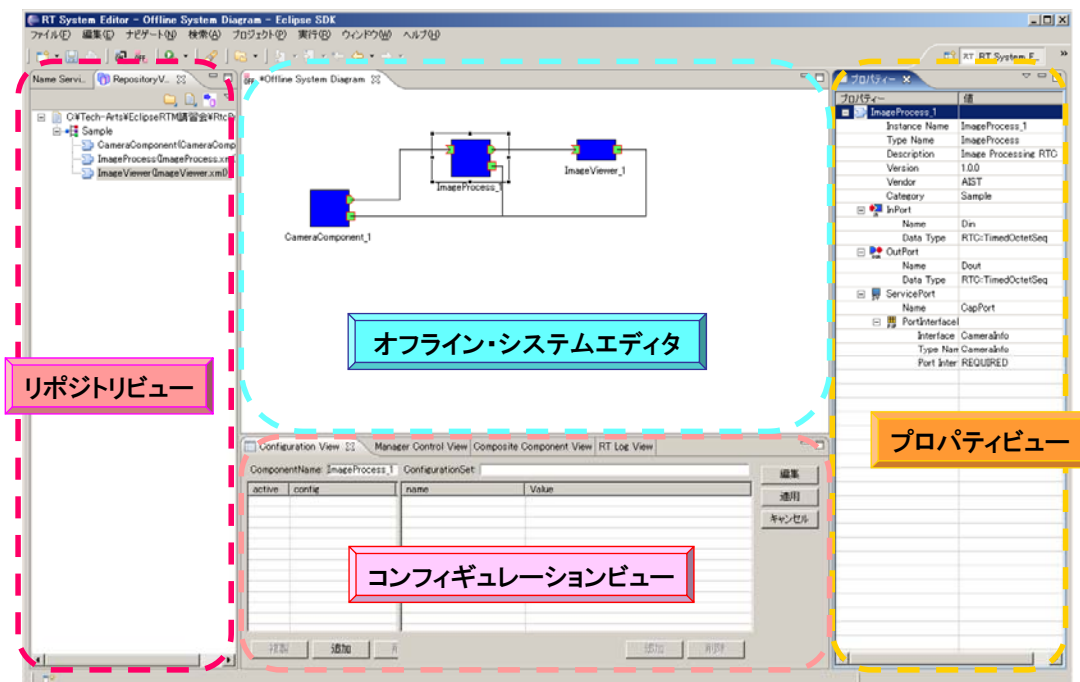
- ① 複合RTCを右クリックし、複合コンポーネントの解除を選択
- ② 複合コンポーネントが分解され、内部のRTCが表示



※エディタ上で、(Deleteキーなどで)単純に削除した場合は、エディタから表示が消えるのみ複合コンポーネントは解除されない

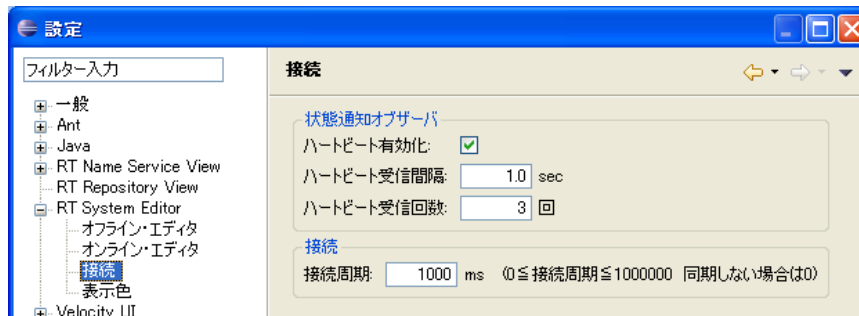
オフラインエディタ

- RTコンポーネントの仕様を用いてRTシステムを構築
 - 実際のRTコンポーネントが動作している必要はない



■ 接続一状態通知オブザーバ

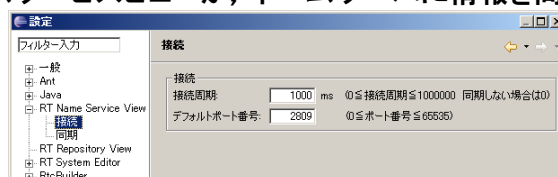
- RTCの生存確認用オブザーバに関する設定
 - RTSE側から生存確認を行うのではなく、RTC側から通知(ハートビート)を行う形
 - OpenRTM-aist-1.1以降で対応



- ハートビート有効化: ハートビートによる生存確認機能の有効化
- ハートビート受信間隔: ハートビートの受信間隔. この間隔以内にRTC側からハートビートが送られてこないで生存確認失敗と判断
- ハートビート受信回数: この回数を超えて生存確認に失敗した場合, 対象RTCに異常が発生したと判断

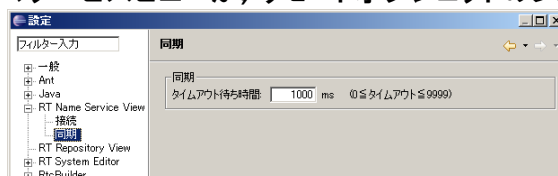
■ 「RT Name Service View」-「接続」【接続周期】

- ネームサービスビューが、ネームサーバに情報を問い合わせる周期



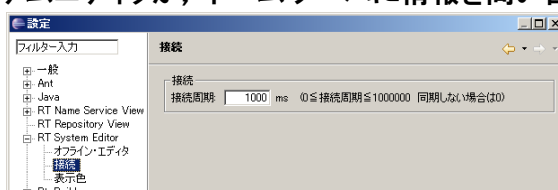
■ 「RT Name Service View」-「同期」【タイムアウト待ち時間】

- ネームサービスビューが、リモートオブジェクトのレスポンスを待つ時間



■ 「RT System Editor」-「接続」【接続周期】

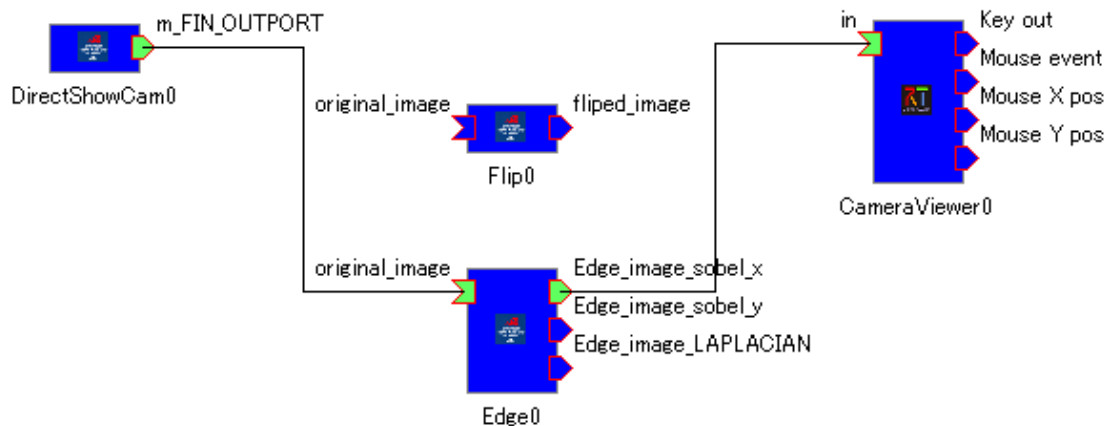
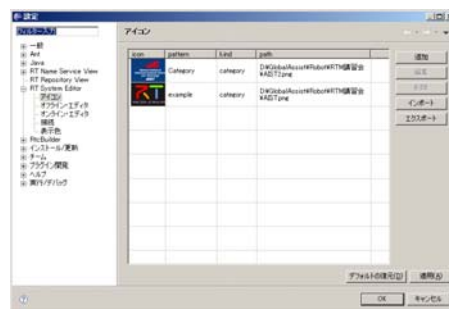
- システムエディタが、ネームサーバに情報を問い合わせる周期



【接続周期】をゼロに設定すると
ネームサーバとの同期を行わない

■ 「RT System Editor」-「アイコン」【表示アイコン】

- RTC内に表示するアイコンを指定可能
 - カテゴリ単位, RTC名称単位で設定が可能



RTミドルウェア講習会

日時: 2012年5月22日(水) 10:00~16:45
場所: つくば国際会議場 小会議室303