



HASC Tutorial 2011

<http://hasc.jp/>

スライドコンテンツ

- HASC Loggerの設定
- HASC Tool（ダウンロード版）の使い方
- HASC Tool（開発版）のセットアップ

Part 2

名古屋大学 河口研究室
修士2年 小川延宏

HASC Loggerの設定

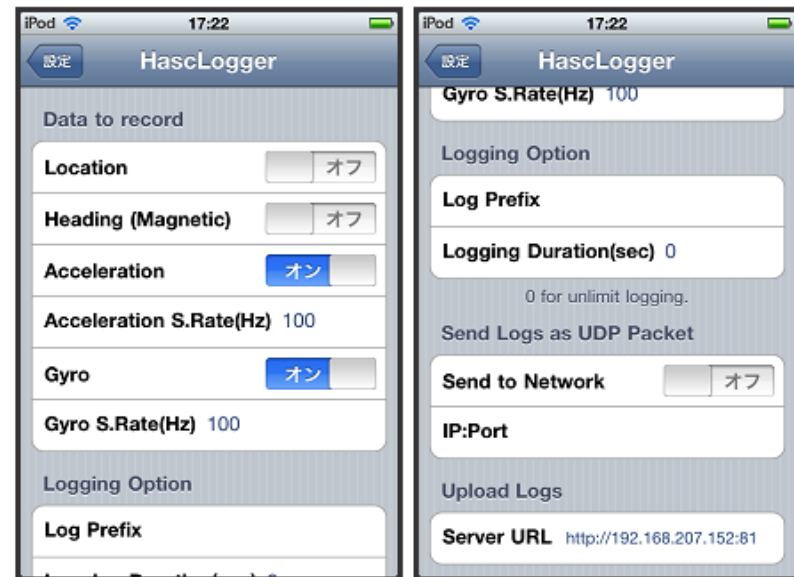


- HASCChallenge2011 > HASC Loggerを参照

- <http://hasc.jp/hc2011/hasclogger.html>
- HASC Toolの説明>利用画面を参照

- 設定アプリ内のHASCLoggerを選択

- 収集データ
- サンプリングレート
- データ収集時間
- 行動データのリアルタイム転送先
- HASC LoggerServerへのデータ転送先



行動データ収集設定

- Data to record
 - Location : 位置情報
 - Heading : 地磁気情報
 - Acceleration : 加速度情報
 - Acceleration S. Rate (Hz) : 加速度計測周波数
 - Gyro : 角速度情報
 - Gyro S. Rate (Hz) : 角速度計測周波数
- Logging Option
 - Logging Duration(sec) : データ計測時間
 - 0の場合 : 制限なし

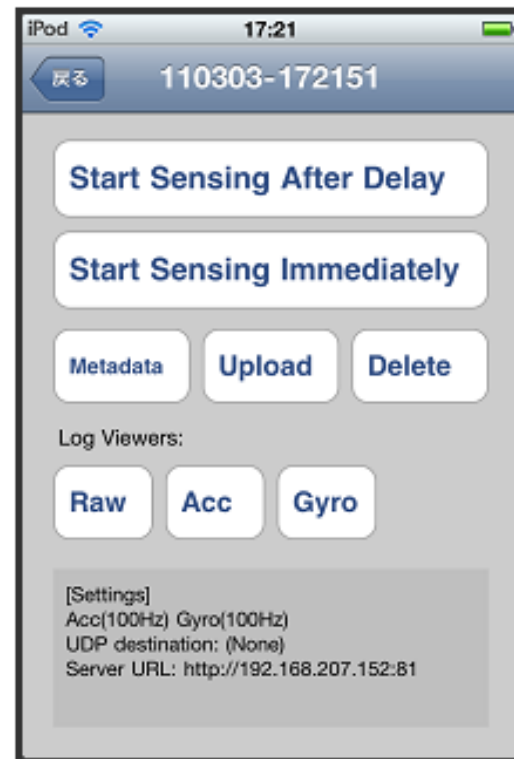
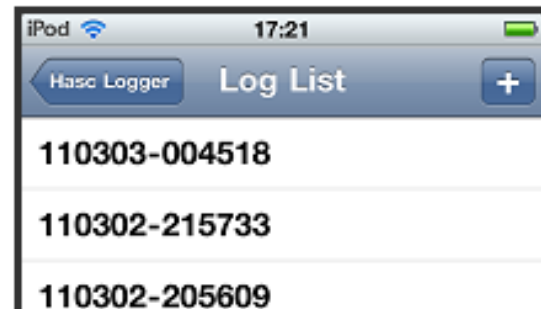
行動データの送信

- Send Logs as UDP Packet
 - HASCToolへ行動データをリアルタイム送信
 - Send to Network
 - IP:Port
 - HASC Toolを動かしているPCのIPアドレス+ポート番号6666
 - Ex) 192.168.100.10:6666
- Upload Logs
 - 収集した行動データをHASCToolにアップロード
 - Server URL
 - "http://" + HASC Toolを動かしているPCのIPアドレス+":"+ポート番号+"/ "
 - Ex) http://192.168.100.10:80/

HASC Logger起動画面



- HASC Loggerの起動
 - Log List
 - New
- データ収集の設定
 - Start Sensing After Delay
 - Start Sensing Immediately
 - Metadata
 - Upload
 - Delete
 - Log Viewers: Raw, Acc, Gyro



メタ情報の設定

- データ収集環境情報を入力
- Comment
 - 直接打ち込み
 - 自由にコメントをかけます
- Tags
 - リストから選択・追加
 - 直接打ち込みも可能
 - カンマ “,” で区切る

100808-190344 Metadata

Comment:
test

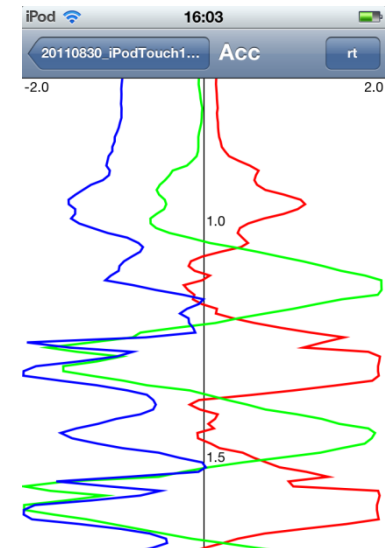
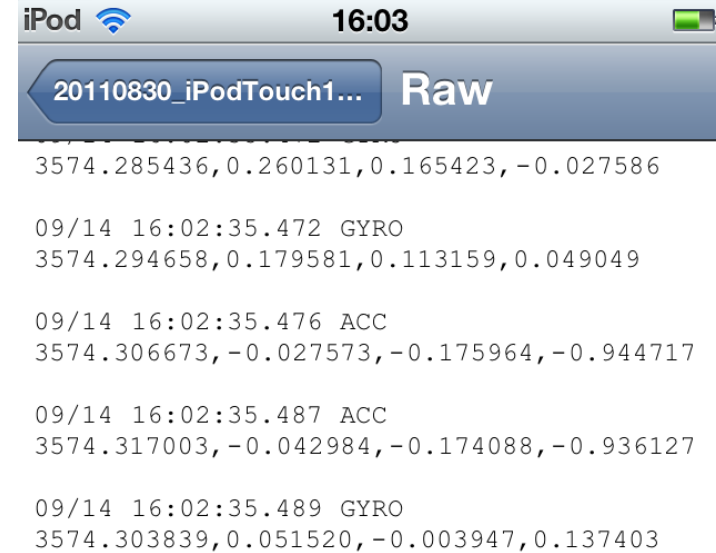
Tags: (separated by "," character)
Walk,InBag

Add Remove Last

InHand
InPocket
InBag
RideCar
RideTrain

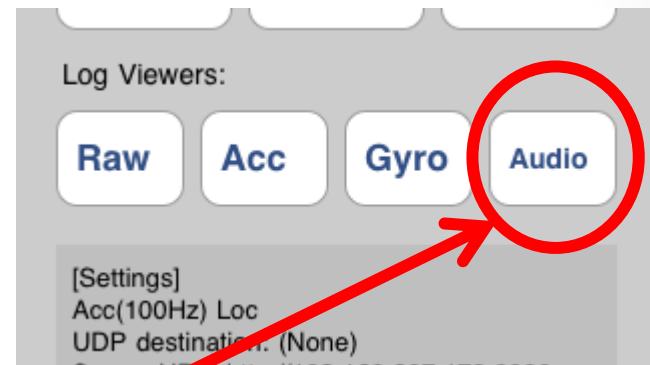
データ収集画面

- データ収集中
 - 現在のセンサ値を表示
- データ収集の終了
 - 手動：Stop Sensingボタンを押す
 - 自動：事前にデータ収集時間を指定
- 収集データの形式
 - 日時 センサの種類
 - 時間(sec), センサ値 1, センサ値 2, . . .
- 収集データの閲覧
 - HASC Logger起動画面
 - Log Viewers項目で、Raw, Acc, Gyro ボタンを選択



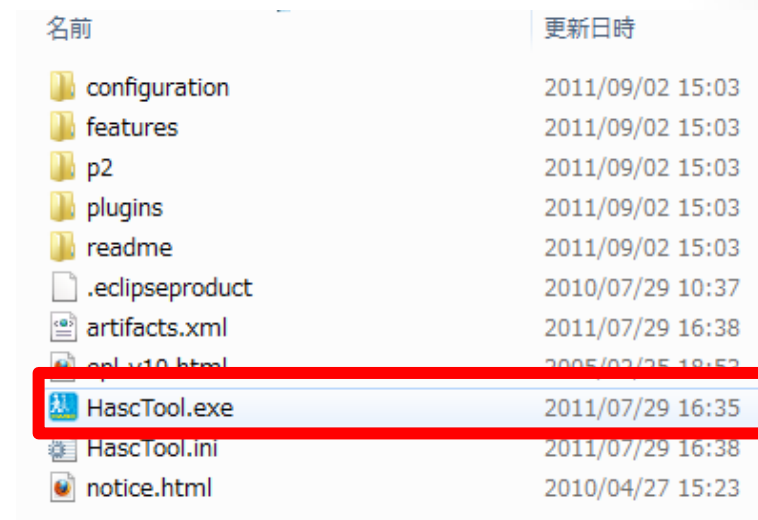
HASC Logger [更新情報]

- 録音機能の追加
 - 音情報の取得
 - 環境音
 - 音声
 - 音情報の再生
“Audio” ボタン
- 設定方法
 - 設定 > HascLogger
> Audio を ON



HASC Tool(ダウンロード版)

- <http://hasc.jp/hc2011/download.html>
- HASC Tool (ダウンロード版)の入手
 - HASC Tool
 - Mac版
 - Windows版
 - Sample Data Project
 - RealWorldActivityData
- HASC Toolの起動
- それぞれのファイルを解凍
- HASC Toolディレクトリを開く
- HASC Toolの起動

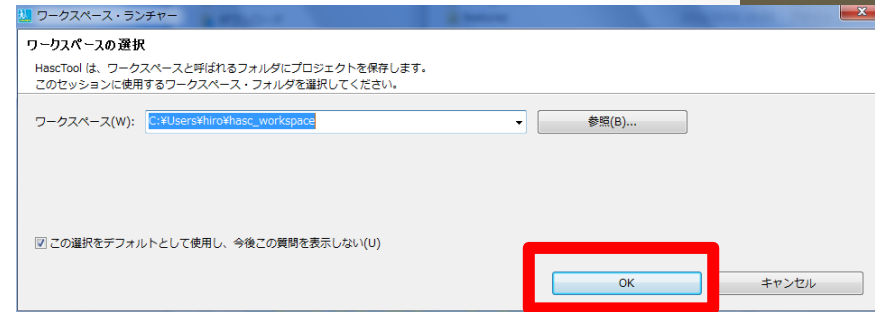


名前	更新日時
configuration	2011/09/02 15:03
features	2011/09/02 15:03
p2	2011/09/02 15:03
plugins	2011/09/02 15:03
readme	2011/09/02 15:03
.eclipseproduct	2010/07/29 10:37
artifacts.xml	2011/07/29 16:38
apl_v10.html	2005/02/25 10:53
HascTool.exe	2011/07/29 16:35
HascTool.ini	2011/07/29 16:38
notice.html	2010/04/27 15:23

HASC Toolの起動

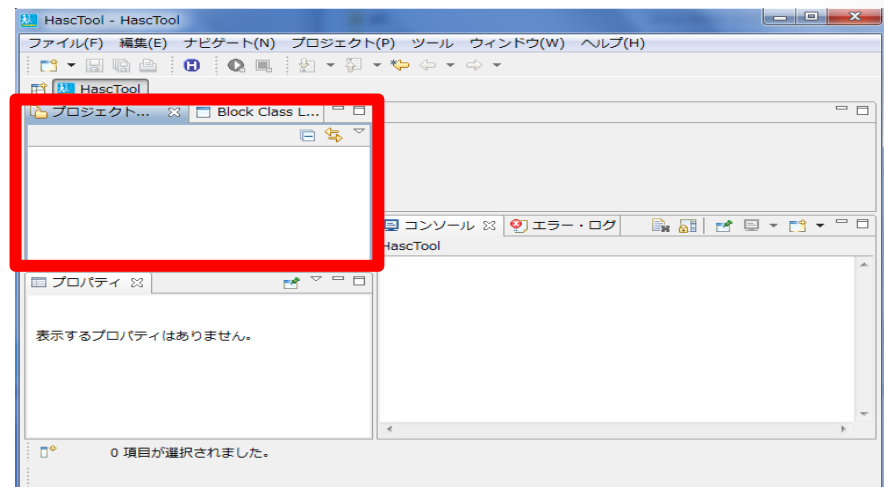
- ワークスペースの選択

- 任意に設定可
- デフォルトでもOK！



- 新規プロジェクトの作成

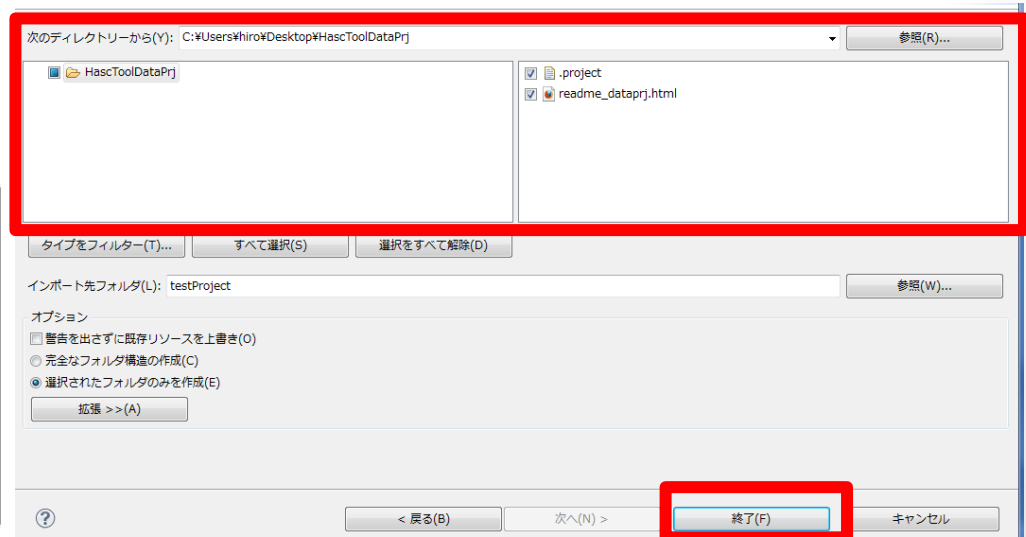
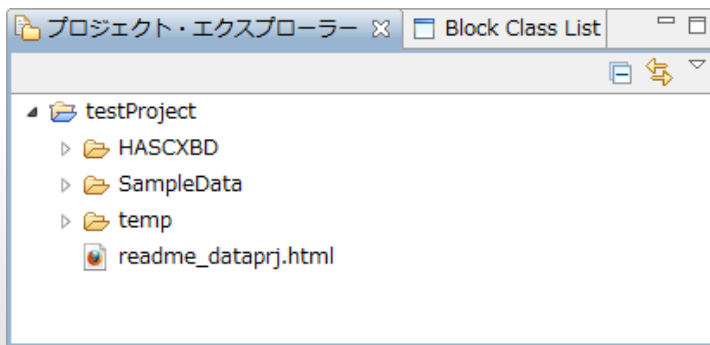
- プロジェクト・エクスプローラ内で右クリック
- 新規>プロジェクト
- 一般>プロジェクト
- >次へ
- プロジェクト名
HASCtestPrj



Sample Data Projectのインポート

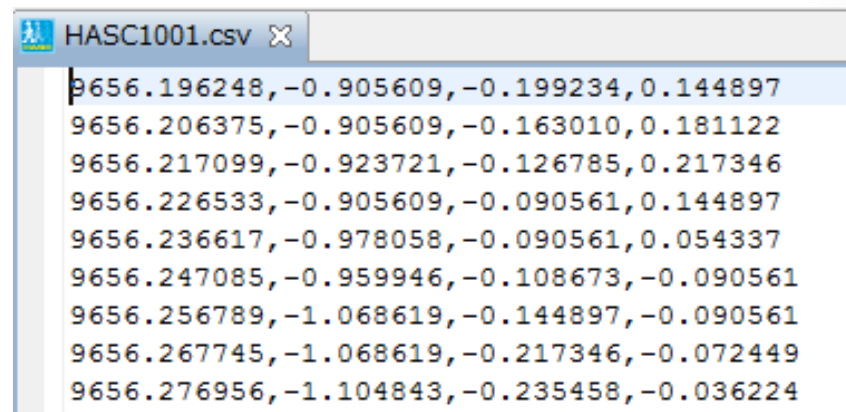
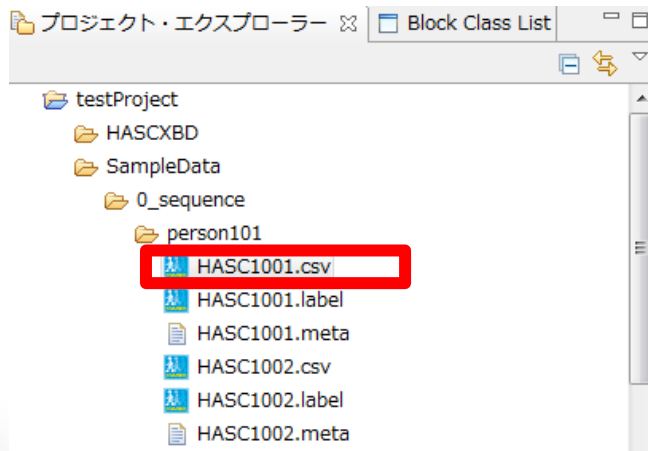
SampleDataProject : サンプル行動データセット+処理スクリプト

- 作成したプロジェクト
 - 右クリック>インポート>一般>ファイルシステム
 - 参照元の選択 (SampleDataProjectを選択)
 - 出てきたすべての項目にチェック
 - 終了を選択
 - インポート完了



行動データ (.csv)の閲覧

- データ構成
 - 加速度データ
 - [時間(sec)], [x軸], [y軸], [z軸]
 - データフォーマットについては、
<http://hasc.jp/hc2011/dataFormat.html>
に記載されています。



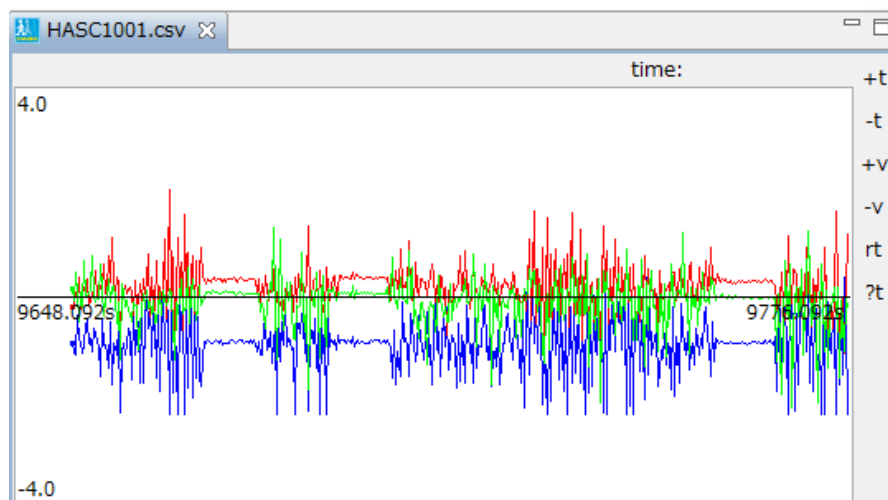
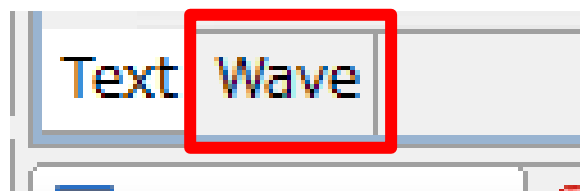
行動データのViewer

- 波形の表示

- 縦軸：加速度 (G)，横軸：時間 (sec)
- センサX軸：青色，センサY軸：赤色，センサZ軸：緑色

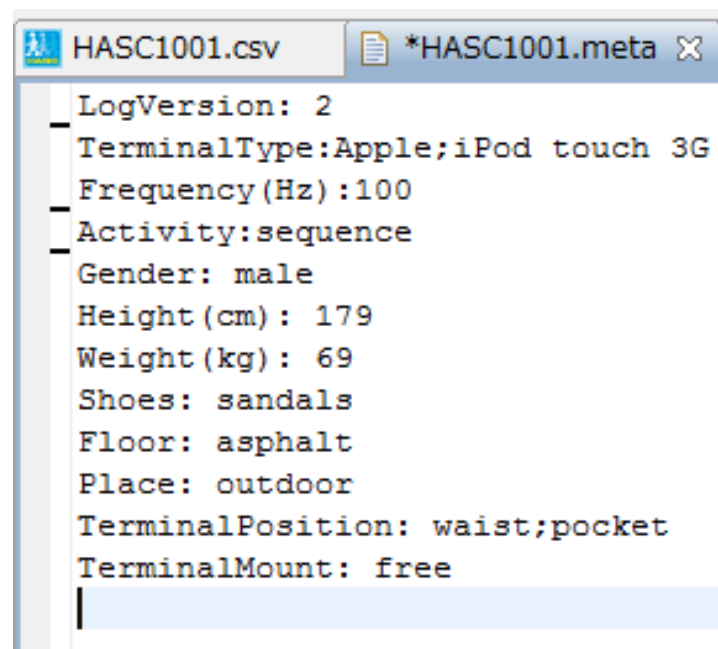
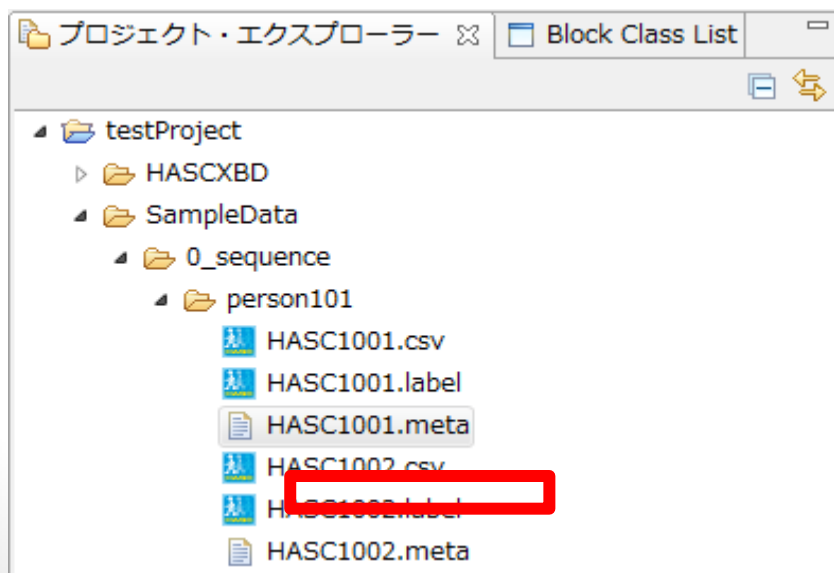
- 波形ビューの操作

- 表示する範囲の変更：“+t”，“-t”，“+v”，“-v”
- 時刻の左端を0とする：“rt”
- 選択した時間範囲をテキストエディタ上に表示：“?t”



メタデータ (.meta)

- データの要素
 - TerminalType, Frequency (Hz), Activity, Gender, Height (cm) など
 - key: value



メタデータ [更新情報]

- データの要素を変更

- 必須要素

- TerminalType, Frequency (Hz), Gender, Generation, Height (cm), Weight (kg), Shoes, TerminalMount, TerminalPosition, AttachmentDirection

- 6種類の行動データで必要

- Activity, Floor, Place

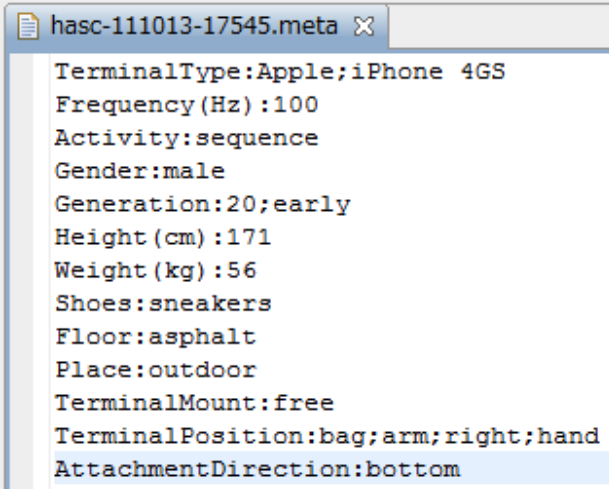
- 実環境データで必要

- Weather, Start, Goal

- 参考URL

- HASC Challenge 2011

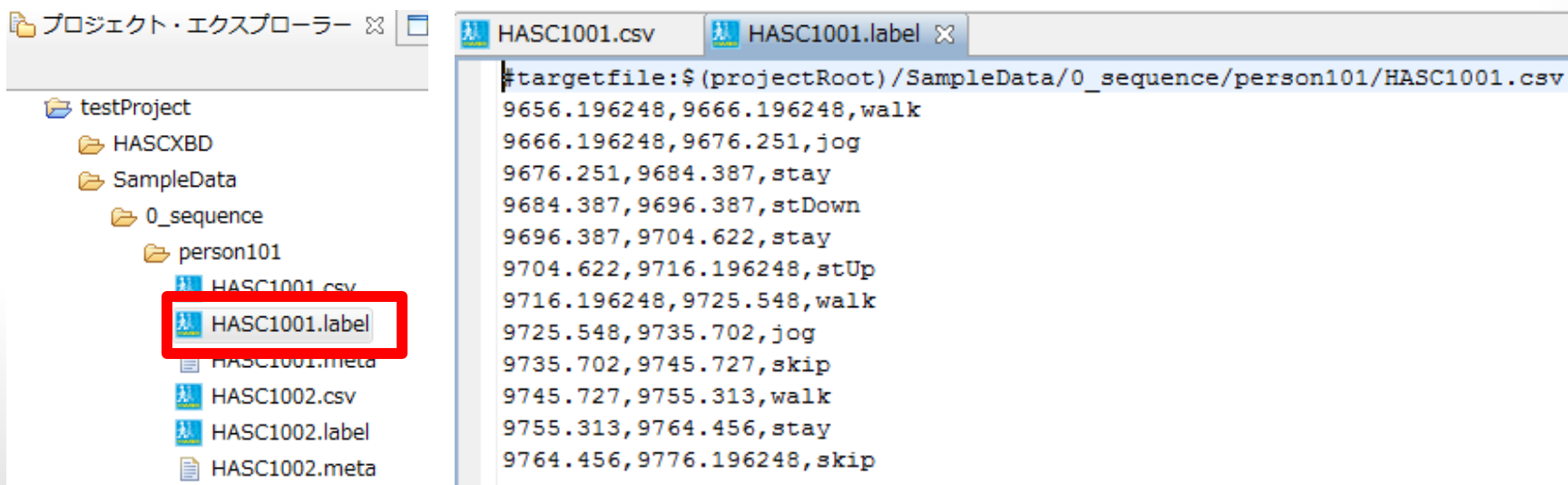
- <http://hasc.jp/hc2011/dataFormat.html#DF2>



```
hasc-111013-17545.meta
TerminalType:Apple;iPhone 4GS
Frequency(Hz):100
Activity:sequence
Gender:male
Generation:20;early
Height(cm):171
Weight(kg):56
Shoes:sneakers
Floor:asphalt
Place:outdoor
TerminalMount:free
TerminalPosition:bag;arm;right;hand
AttachmentDirection:bottom
```

ラベルデータ (.label)

- データ構成
 - 1行目：対象とするcsvデータの絶対パス
 - 2行目以降
 - [開始時刻(sec)][終了時刻(sec)][Activity]



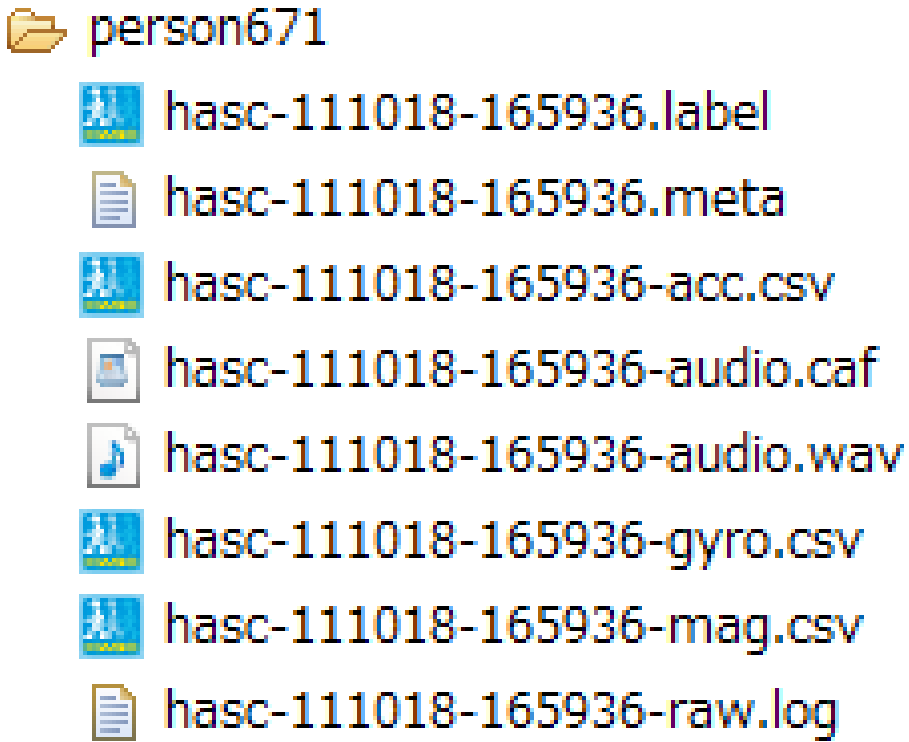
The screenshot shows a project explorer on the left and a text editor on the right. The project explorer displays a tree structure with the following items:

- testProject
 - HASXBD
 - SampleData
 - 0_sequence
 - person101
 - HASC1001.csv
 - HASC1001.label** (highlighted with a red box)
 - HASC1001.meta
 - HASC1002.csv
 - HASC1002.label
 - HASC1002.meta

The text editor shows the content of the selected file, HASC1001.label, which contains the following text:

```
#targetfile:$(projectRoot)/SampleData/0_sequence/person101/HASC1001.csv
9656.196248,9666.196248,walk
9666.196248,9676.251,jog
9676.251,9684.387,stay
9684.387,9696.387,stDown
9696.387,9704.622,stay
9704.622,9716.196248,stUp
9716.196248,9725.548,walk
9725.548,9735.702,jog
9735.702,9745.727,skip
9745.727,9755.313,walk
9755.313,9764.456,stay
9764.456,9776.196248,skip
```

行動データセット [更新情報]

- 行動データ
 - 拡張子 : csv
 - 加速度情報, 角速度情報, 地磁気情報など
 - メタデータ
 - 拡張子 : meta
 - 計測環境の情報
 - ラベルデータ
 - 拡張子 : label
 - 音データ
 - 拡張子 : caf, wav
- 

person671

 - hasc-111018-165936.label
 - hasc-111018-165936.meta
 - hasc-111018-165936-acc.csv
 - hasc-111018-165936-audio.caf
 - hasc-111018-165936-audio.wav
 - hasc-111018-165936-gyro.csv
 - hasc-111018-165936-mag.csv
 - hasc-111018-165936-raw.log

ラベルデータ [更新情報]

- 音声データの絶対パス
 - 下図 2 行目を参考
 - #audiofile: “ファイルパス”
- イベント情報
 - 下図 4 行目を参考
 - Ex) 6071.934, , move; floor; B1F

音声データを再生する
場合の遅延オフセット

```
hasc-111018-165936.label x
#targetfile:$(projectRoot)/rwActData/person671/hasc-111018-165936-acc.csv
#audiofile:$(projectRoot)/rwActData/person671/hasc-111018-165936-audio.wav; offsettime:10.0
5057.661,5091.26,walk;floor;B2F;steps;44
5071.934,,walk;turn
5098.502,5126.499,escalatorUp;stay
5125.117,,move;floor;B1F
5127.665,5143.411,walk;floor;B1F;steps;22
5154.309,5162.703,escalatorUp;stair
5165.451,,move;floor;1F
```

ラベルデータ [更新情報]

- データの要素

- 必須

- 対象としている行動データの絶対パス
 - 行動ラベル情報

- 任意

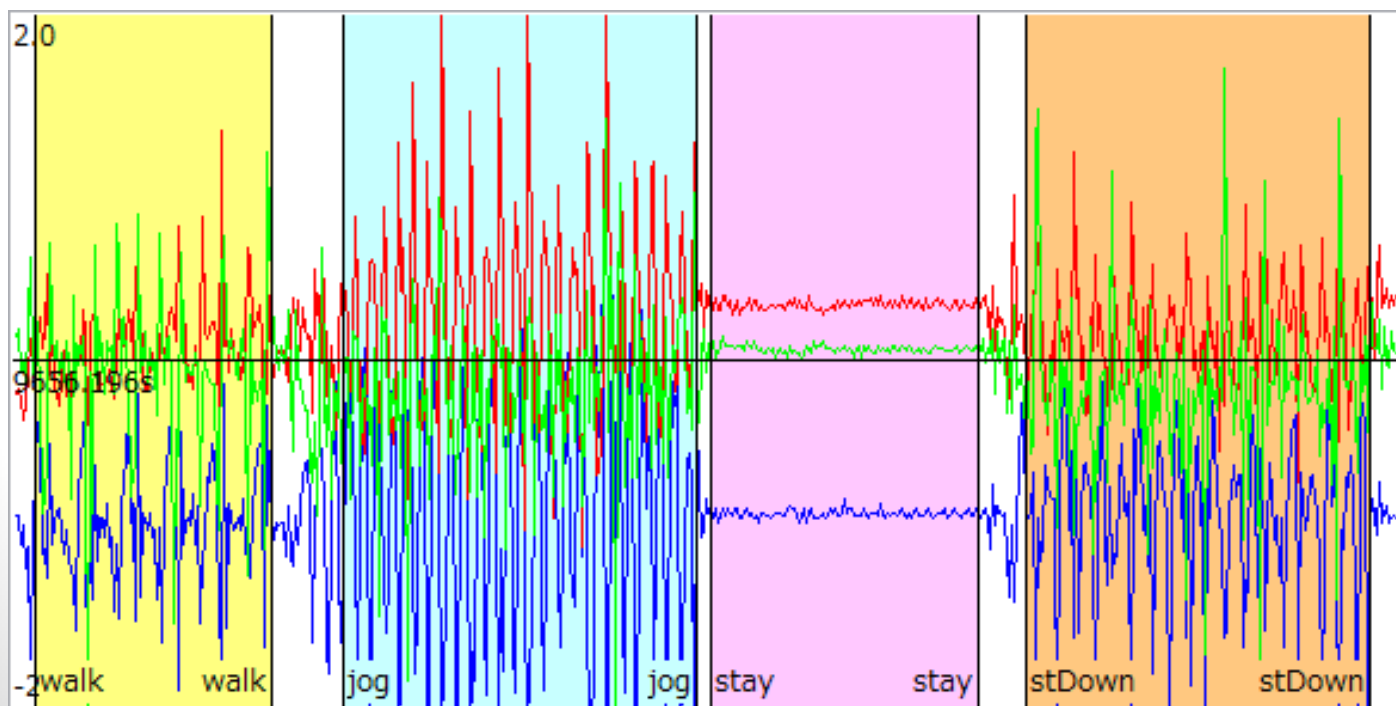
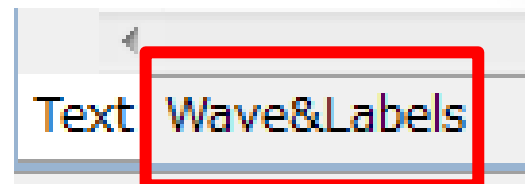
- 音声データの絶対パス
 - イベント情報

- 参考URL

- <http://hasc.jp/hc2011/dataFormat.html#DF3>

ラベルデータのViewer

- ラベル付き波形表示
 - Wave&Labelsをクリック
- ラベル付け：“+L”

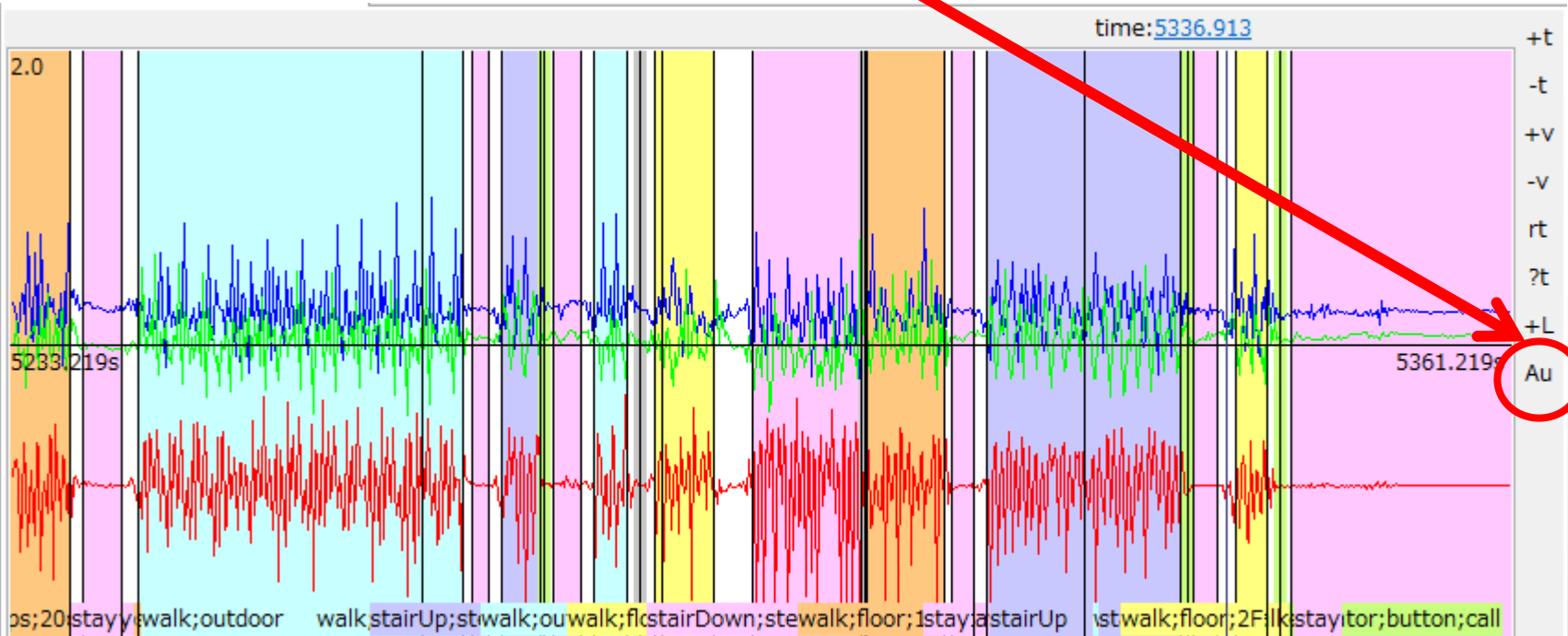


+t
-t
+V
-V
rt
?t
+L

ラベルデータのViewer

[更新情報]

- 音声情報と行動波形が同期
- 音の再生：“Au”



行動信号のセグメンテーション

[更新情報]

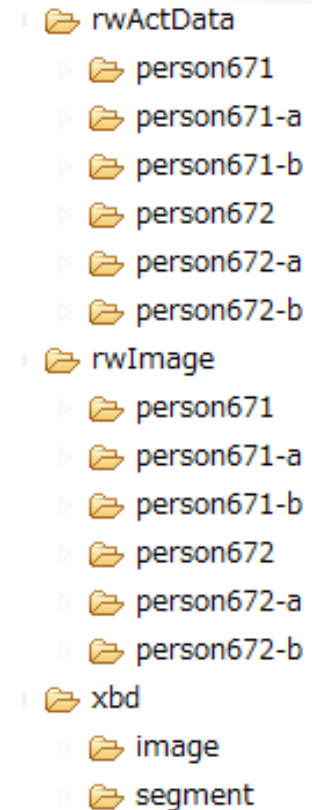
- 行動情報を利用した情報システム開発
 - 要素技術
 - 1. 自動ラベリング
 - 2. 自動セグメンテーション
 - 3. 自動経路推定
 - 実環境データ
 - ランドマーク間を移動
 - ランドマーク間には経路がある

実環境データを収集！

RealWorldActivityDataのインポート

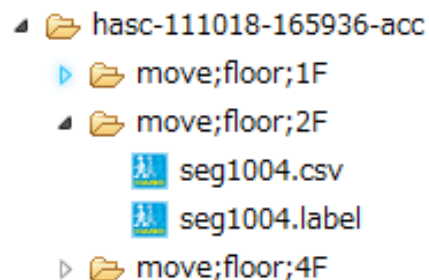
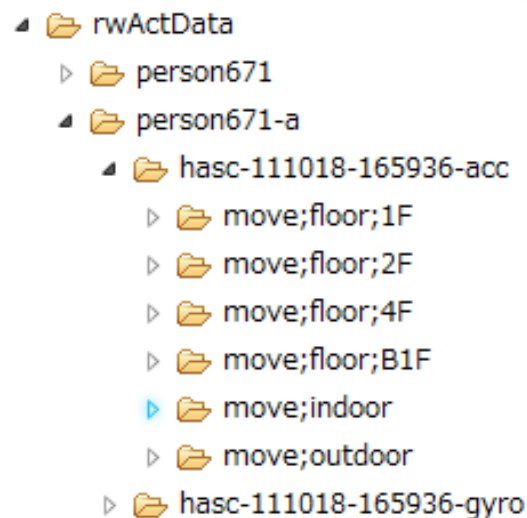
[更新情報]

- プロジェクトを新規作成
 - プロジェクト名 : HASCrwPrj
- 作成したプロジェクト
 - 右クリック > インポート > 一般 > ファイルシステム
 - 参照元の選択 (HASCrwdataを選択)
 - 出てきたすべての項目にチェック
 - アラームがでたら “OK” を押してください
 - 終了を選択
 - インポート完了



実環境サンプルデータセット [更新情報]

- rwActData
 - 2人分の実環境データ
 - -a, -bはラベル情報を変更
- rwImage
 - イメージ画像セット
- xbd
 - 処理スクリプト



実環境データ (-aの場合)

Acceleration

walk > escalatorUp;stay

walk > escalatorUp;stair

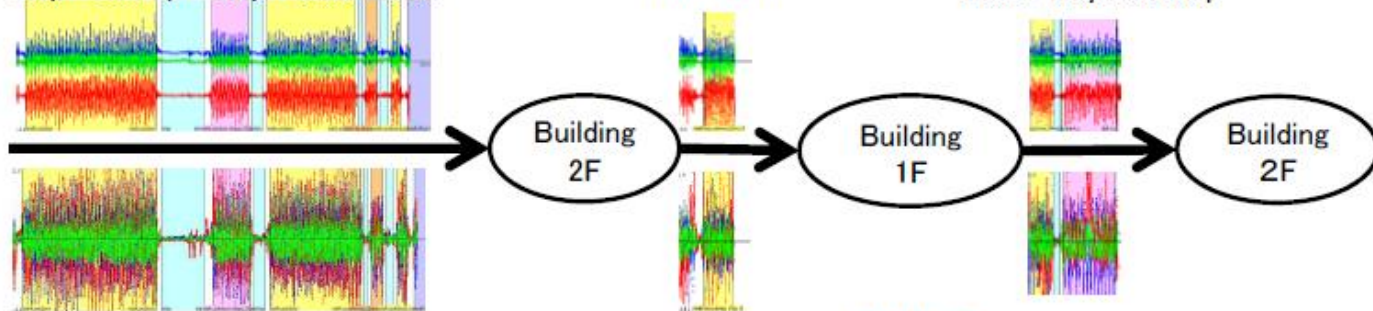


Gyro

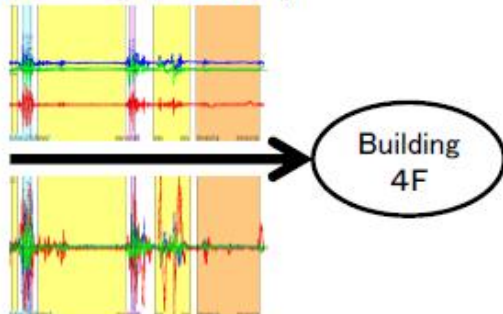
walk > stay > walk > stay > walk > stay > stairUp > stay > walk > walk

stairDown

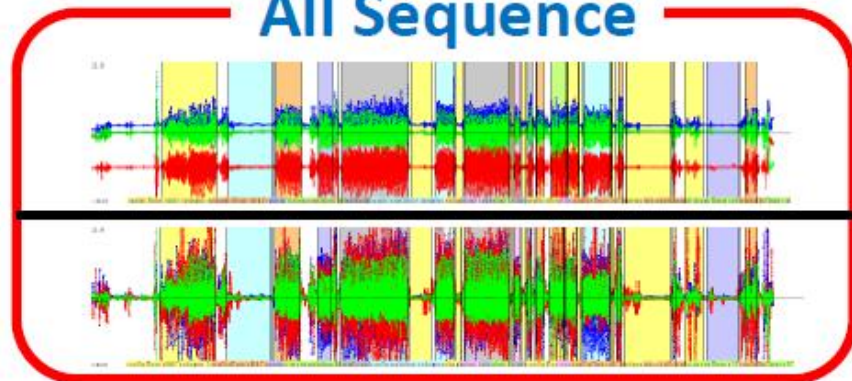
walk > stay > stairUp



stay > walk > stay
walk > stay > elevatorUp



All Sequence



処理スクリプト [更新情報]

- segment
 - label
 - xbdExe_SegmentLabel.hascxbd
 - ラベルファイルをイベント情報で区切る
 - signal
 - xbdExe_SegmentCSV.hascxbd
 - 行動データファイル(csvファイル)をイベント情報で区切る
- image
 - xbdExe_imageLabelFile
 - 区切ったデータのイメージを作成

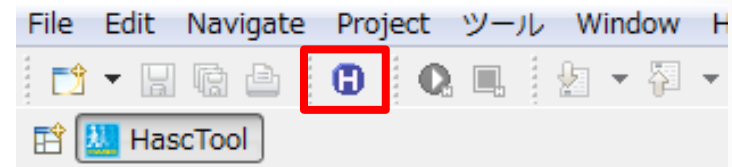
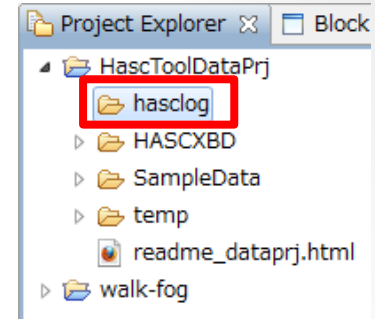
HASC Toolの機能説明

- HASC Loggerとの連携機能
 - HASCLoggerServerへデータ送信
- ラベル付与機能
 - ラベルファイルの編集
 - ラベル付の手順紹介
- 行動データ処理機能
 - HASCXBDディレクトリの内部構成
 - HASCXBDファイルの説明

HASCLoggerServerへデータ送信

- HASC Tool側

- ログの保存先ディレクトリを選択
- ツールバーの (H) ボタン
 - 起動, 終了
 - bindするTCPポートの設定
 - 任意選択 (デフォルトのままでも可)



- HASC Logger側

- Uploadボタン
- 確認画面 (右図)
- アップロード内容
 - 各種センサデータ, メタデータ



ラベル付与機能

- ラベル付手順

- ラベルファイルを用意する

- 拡張子 : .label

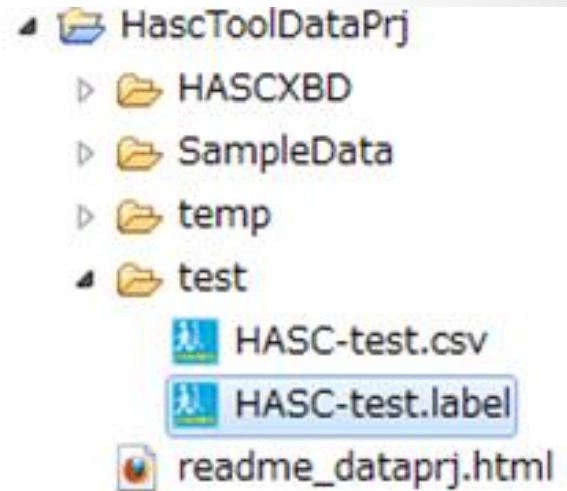
- ターゲットファイルのPathを入力

- ラベルファイルのフォーマットは以下に記載

<http://www.hasc.jp/hc2011/dataFormat.html>

- ラベルファイルを編集

- ラベル付を実行

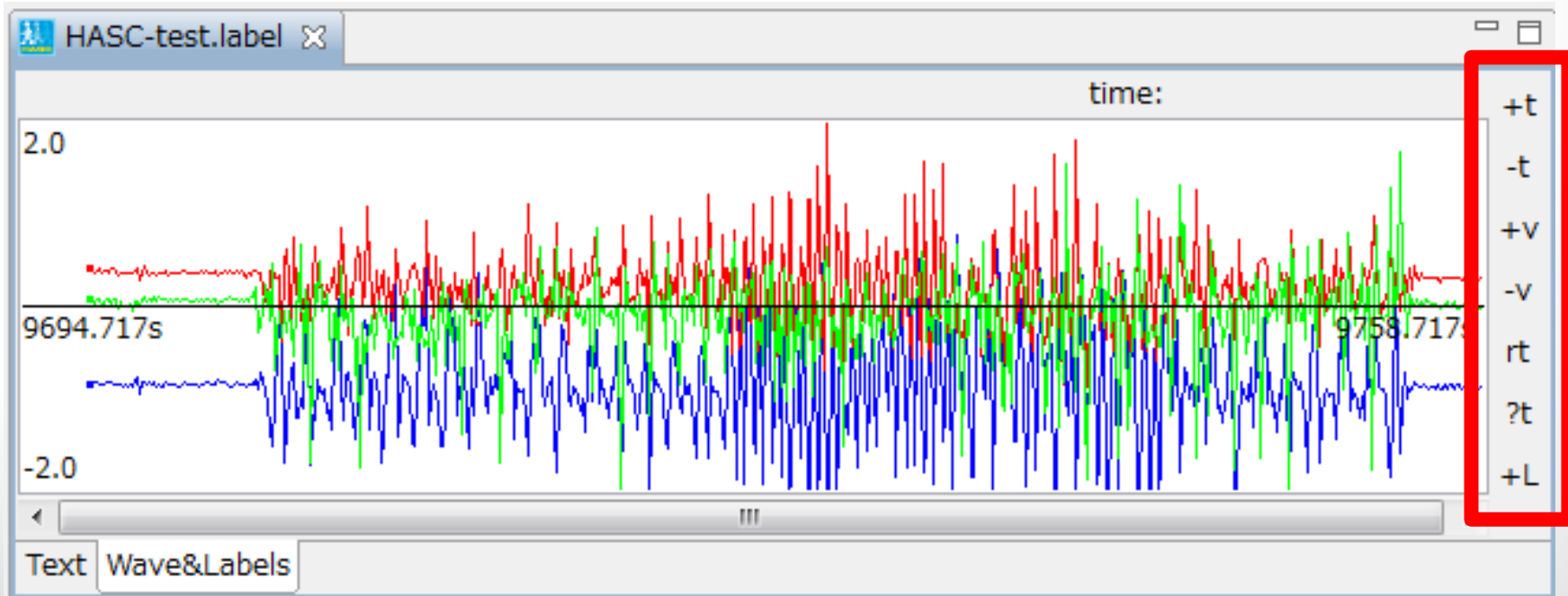


HASC-test.label ⌵

```
#targetfile:$(projectRoot)/test/HASC-test.csv
```

ラベルファイルの編集

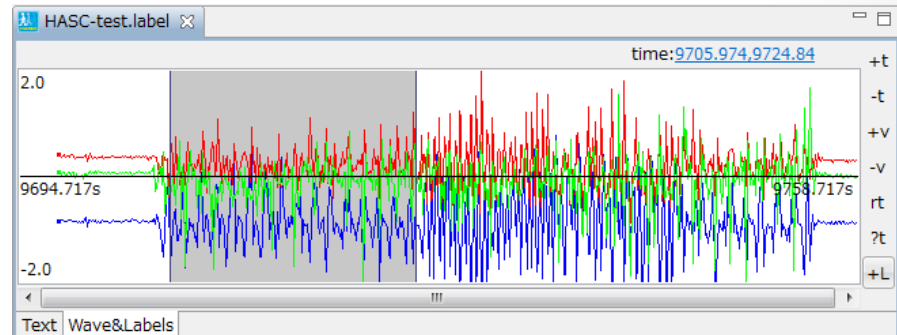
- 編集ボタンは右側に配置
 - 表示する範囲の変更：“+t”，“-t”，“+v”，“-v”
 - 時刻の左端を0とする：“rt”
 - 選択した時間範囲をテキストエディタ上に表示：“?t”
 - ラベル付ボタン：“+L”



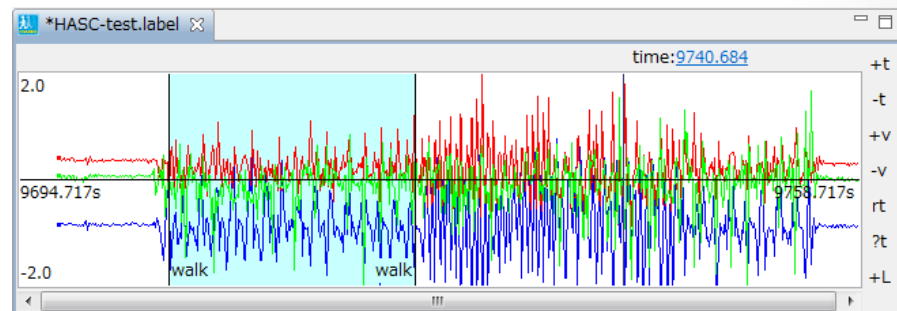
ラベル付の手順

- ラベル範囲を選択
 - ドラッグドロップ
- +L ボタンを押す
- テキスト編集
 - ラベル情報の入力
 - 画像ではwalk
- ラベル付け完成

上記作業を繰り返す



The screenshot shows the text editor window for "HASC-test.label". The text content is: `#targetfile:$(projectRoot)/test/HASC-test.csv` and `9705.974,9724.84,walk`. The word "walk" is highlighted in blue.



行動データ処理機能

- HASCXBDディレクトリ

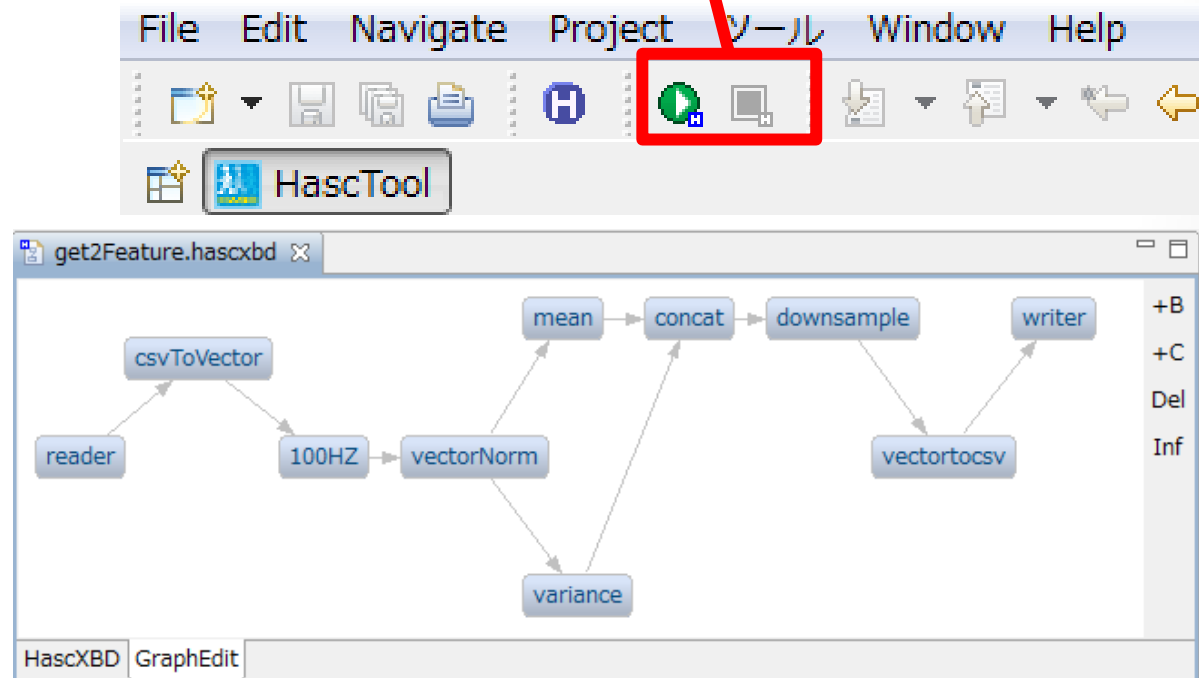
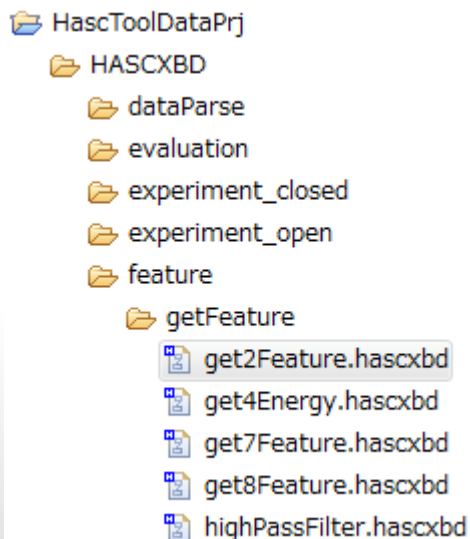
- 特徴量抽出を実行
 - feature
- 機械学習 & 評価を実行
 - experiment_~
 - weka (機械学習)
 - evaluation (評価)
- 閾値処理
 - Threshold
- HASC Loggerからリアルタイムデータ送信
 - Udp
- ファイルのカウント & メタデータテーブルの作成
 - file

- 📁 HASCXBD 405
 - 📁 dataParse 191
 - 📁 evaluation 379
 - 📁 experiment_closed 405
 - 📁 experiment_open 405
 - 📁 feature 405
 - 📁 file 283
 - 📁 frequency 253
 - 📁 label 191
 - 📁 sample 381
 - 📁 table 253
 - 📁 threshold 377
 - 📁 udp 294
 - 📁 view 200
 - 📁 weka 385

HASCXBDファイル

- HASC Tool上で様々なデータ処理を実行
- ダイアグラムの階層化
- ブロック（フィルタ）
- コネクタ

実行／停止



ブロック（フィルタ）

- 実際にデータ処理を行います
- ブロックの新規作成ボタン：“+B”
- クラスファイルの場所を入力
- プロパティに値をセット

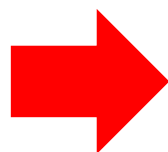


作成するブロックのクラス名を入力して下さい

```
jp.hasc.hasctool.core.runtime.filter.MeanFilter|
```

Property	Value
▲ _block	
_Type	BeanBlock
Class	jp.hasc.hasctool.core.runtime.filter.MeanFilter
Comment	
Name	mean
▲ bean properties	
sampleNum	256

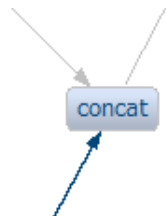
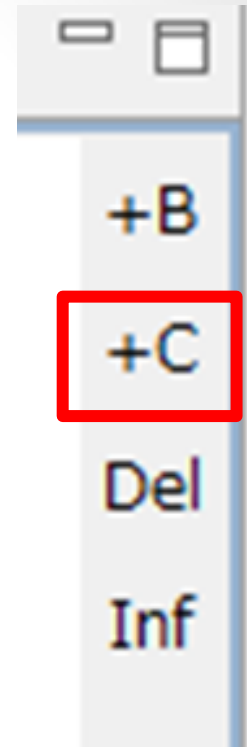
参考



<http://hasc.jp/hc2011/hasctool.html>
“HASC Toolの説明”

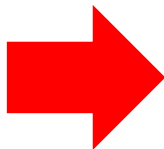
コネクタ

- ブロック間のデータ送信
- コネクタの新規作成ボタン: “+C”
 - 事前に2つのブロックを選択
 - “ctrl” を押しながら選択
- 複数入力ブロック
 - コネクタのプロパティを設定



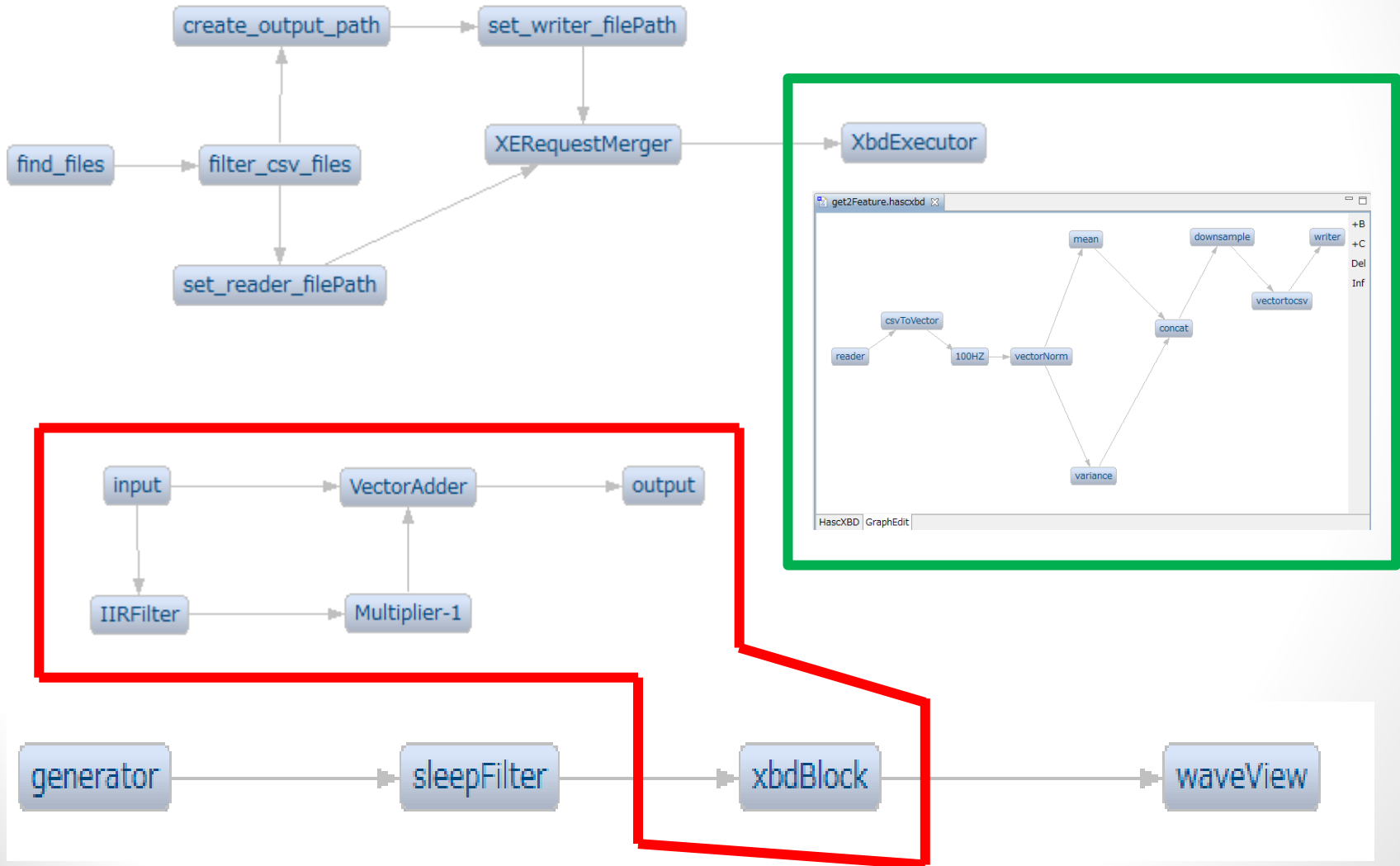
_Type	Connection
Comment	
InputPort	inputPort[1]
OutputPort	outputPort

参考



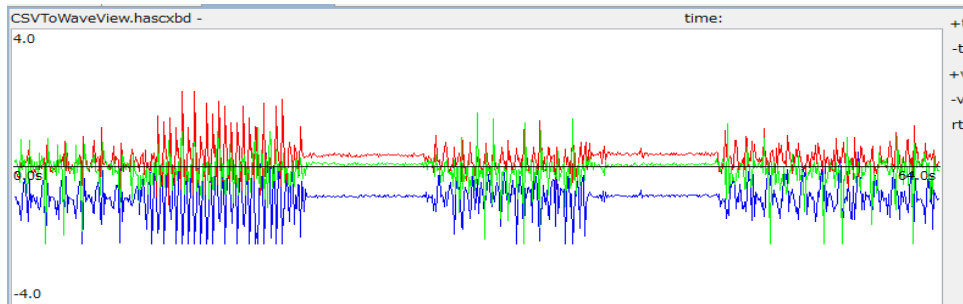
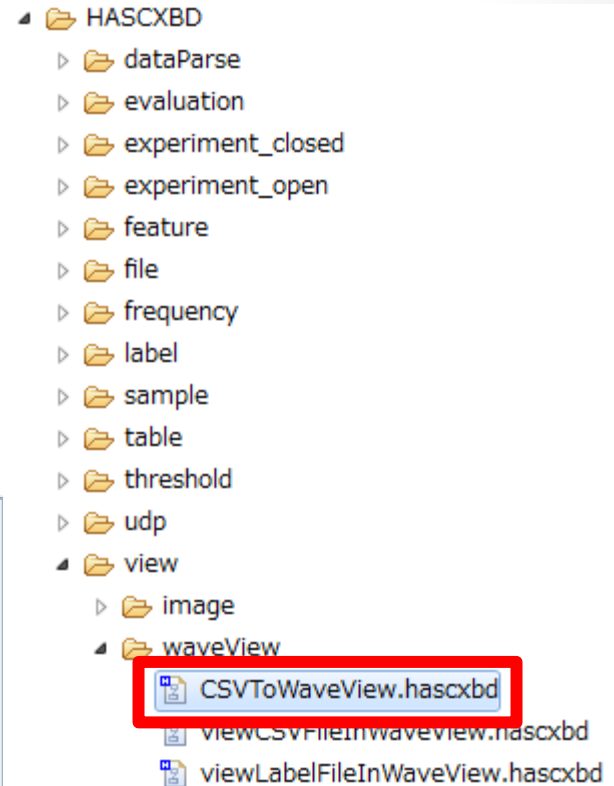
<http://hasc.jp/hc2011/hasctool.html>
“HASC Toolの説明”

ダイアグラムの階層化



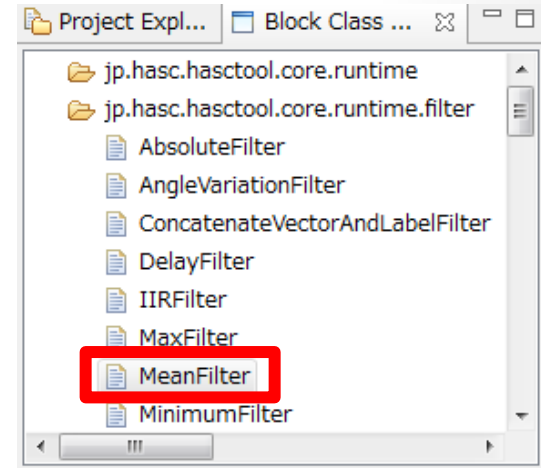
行動データの読込

- HASCXBD>view>waveView>CSVToWaveView.hascxbd
- csvファイルを読み込む
- 行動データ⇒時系列波形



フィルタの追加

- 追加するフィルタの決定
 - Block Class Listから選択
 - 右図ではMeanFilterを選択
 - 平均を計算するフィルタ
 - 場所：jp.hasc.hasctool.core.runtime.filter
 - 選択したらブロックダイアグラム内にできます
 - 出てきたブロックを選択
 - プロパティを設定

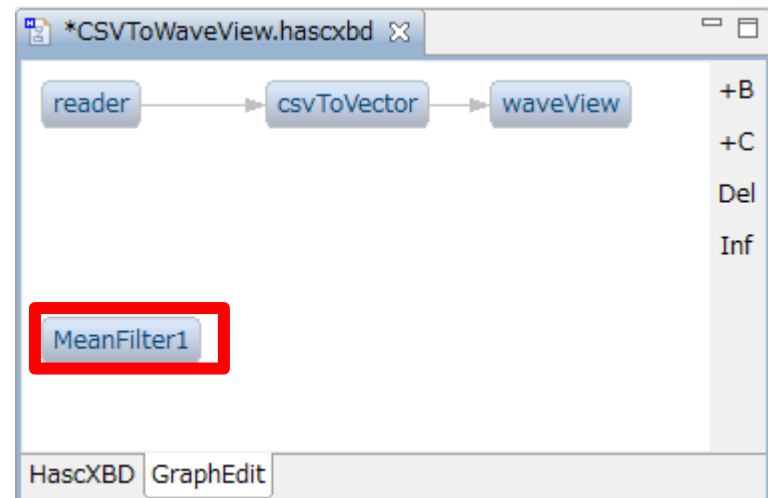


名前も
編集で
きます

The Properties window shows the following table:

Property	Value
_block	
_Type	BeanBlock
Class	jp.hasc.hasctool.core.runtime.filter.MeanFilter
Comment	
Name	MeanFilter1
bean properties	
sampleNum	100

対象データ
の数を設定

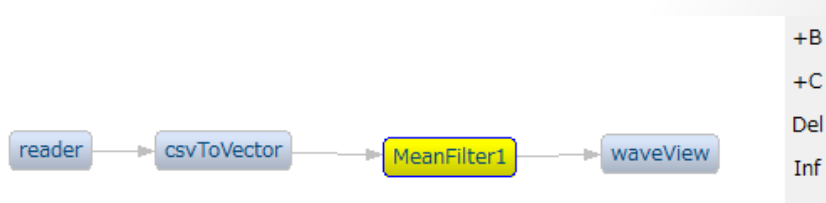
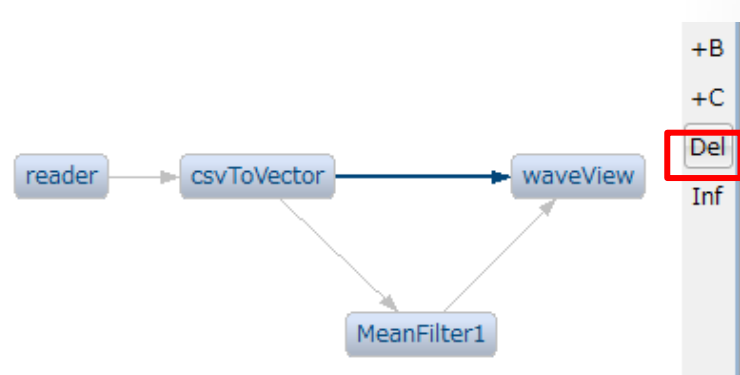
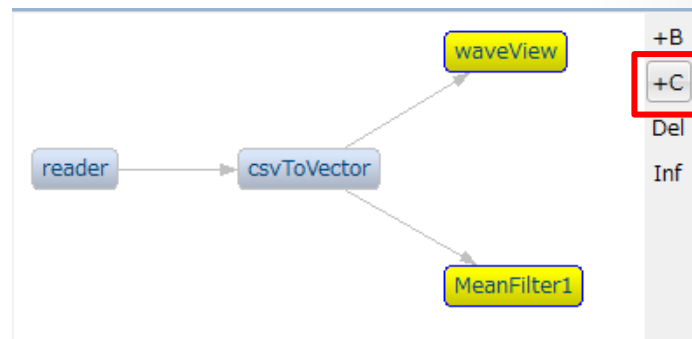


ブロックダイアグラムの編集

- csvToVector と waveView の間に MeanFilter をつなげる

- 編集手順

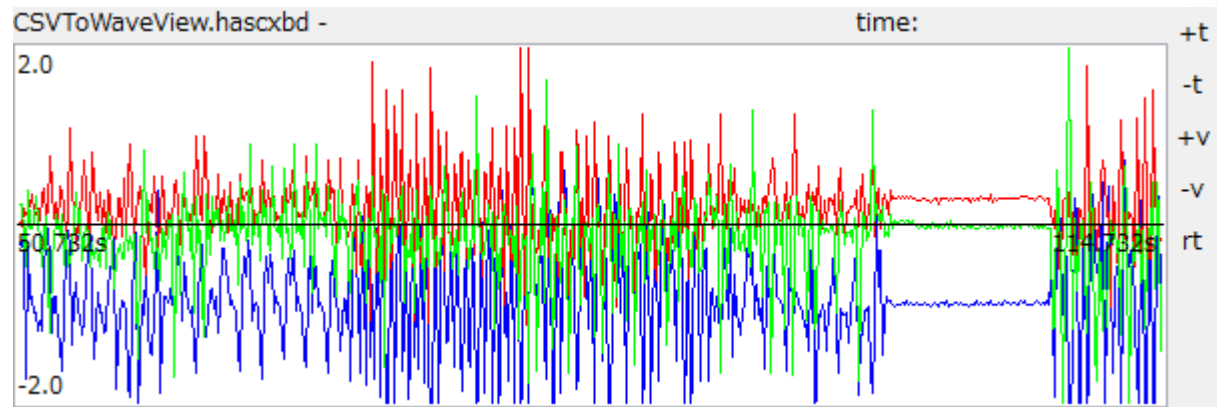
- “Ctrl” キーを押しながら連結するブロックを選択
- “+C” ボタンを押す
- 右図の状態では +C ボタンを押すとエラーが出るが、OK ボタンを押す
- 余分なコネクトを消去



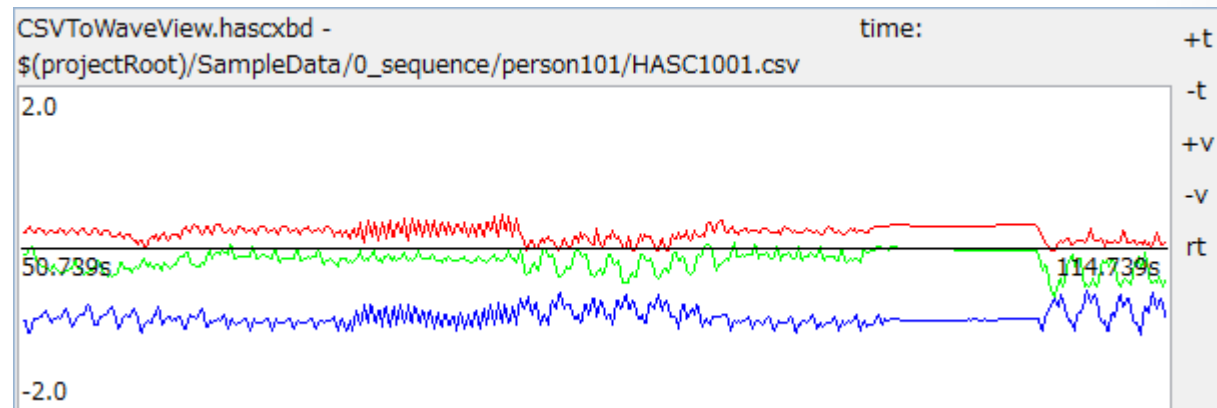
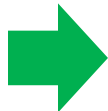
波形の変換

- 作成したブロックダイアグラムを保存
- HASCXBDファイルの実行

波形変換前

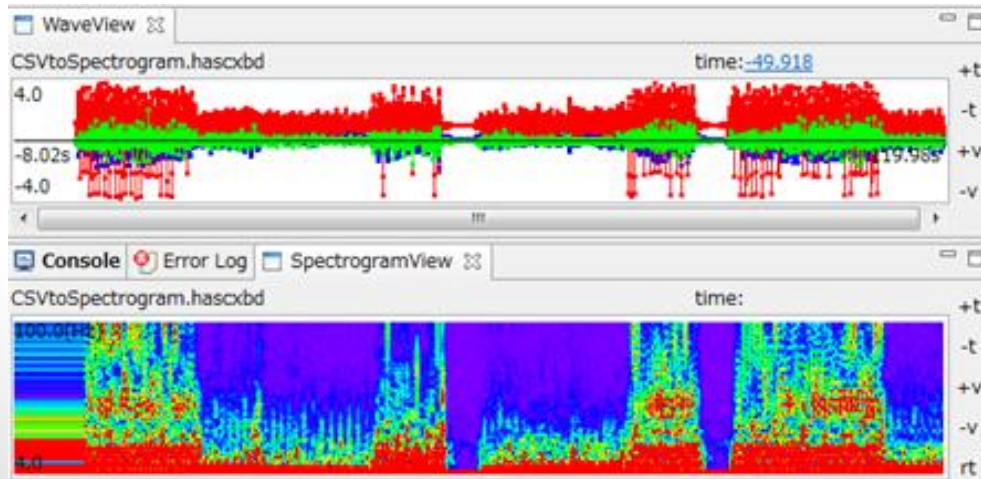
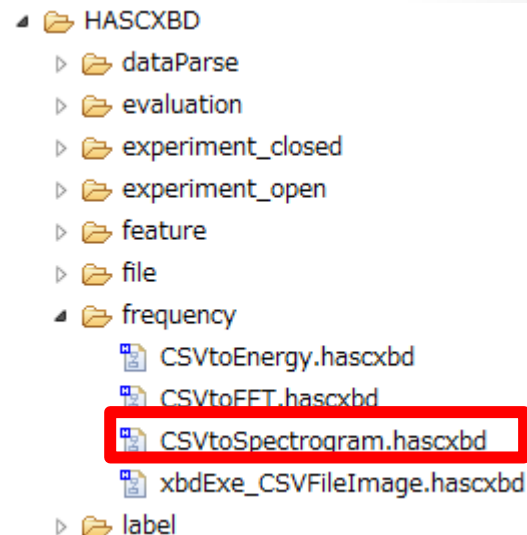


波形変換後



スペクトログラムの表示

- HASCXBD>frequency>
CSVtoSpectrogram.hascxbd
- csvファイルを読み込む
 - 行動データ
⇒スペクトログラム



Hanning : ハニング窓

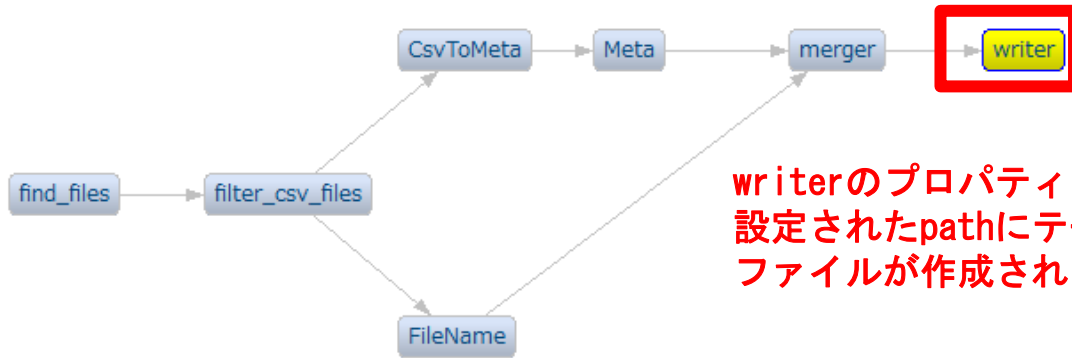
FFT : フーリエ変換

CompToVec : 複素数⇒絶対値

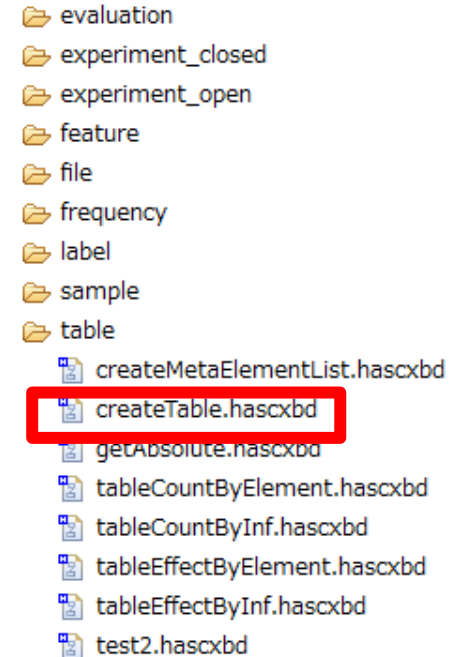
SampleDataテーブルの作成

- HASCXBD>table>
createTable.hascxbd

- 行動データファイル毎の計測環境が一覧できる



writerのプロパティで
設定されたpathにテーブル
ファイルが作成される

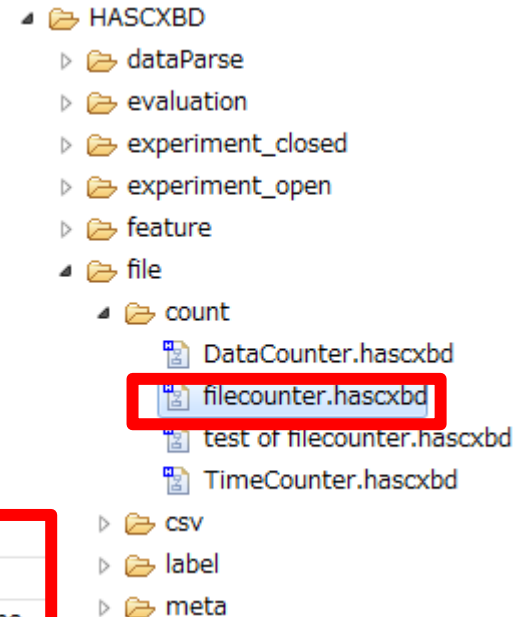


```
createTable.hascxbd | table_output.csv x
```

```
FilePath, FileName, Person, Activity, Activity, TerminalType, Frequency (Hz), Gender, Height (cm), Weight (kg), Floor, Place, Shoes, TerminalMount, TerminalPosi  
$(projectRoot)/SampleData/0_sequence/person101/HASC1001,HASC1001.csv, person101, 0_sequence, sequence, Apple; iPod touch 3G, 100, male, 179, 69, asphalt,  
$(projectRoot)/SampleData/0_sequence/person101/HASC1002,HASC1002.csv, person101, 0_sequence, sequence, Apple; iPhone3Gs, 100, male, 179, 69, asphalt, outc  
$(projectRoot)/SampleData/0_sequence/person102/HASC1003,HASC1003.csv, person102, 0_sequence, sequence, ATR; WAA-001, 100, male, 171, 56, asphalt, outdoor,  
$(projectRoot)/SampleData/0_sequence/person102/HASC1004,HASC1004.csv, person102, 0_sequence, sequence, Apple; iPod touch 3G, 100, male, 171, 56, asphalt,  
$(projectRoot)/SampleData/0_sequence/person102/HASC1005,HASC1005.csv, person102, 0_sequence, sequence, Apple; iPhone3Gs, 100, male, 171, 56, asphalt, outc  
$(projectRoot)/SampleData/0_sequence/person103/HASC1006,HASC1006.csv, person103, 0_sequence, sequence, Apple; iPod touch 3G, 100, male, 170, 54, asphalt,  
$(projectRoot)/SampleData/0_sequence/person103/HASC1007,HASC1007.csv, person103, 0_sequence, sequence, Apple; iPhone3Gs, 100, male, 170, 54, asphalt, outc  
$(projectRoot)/SampleData/0_sequence/person104/HASC1008,HASC1008.csv, person104, 0_sequence, sequence, Apple; iPod touch 3G, 100, male, 166, 54, asphalt,  
$(projectRoot)/SampleData/0_sequence/person104/HASC1009,HASC1009.csv, person104, 0_sequence, sequence, Apple; iPhone3Gs, 100, male, 166, 54, asphalt, outc  
$(projectRoot)/SampleData/0_sequence/person105/HASC1010,HASC1010.csv, person105, 0_sequence, sequence, ATR; WAA-001, 100, male, 169, 46, asphalt, outdoor,  
$(projectRoot)/SampleData/0_sequence/person105/HASC1011,HASC1011.csv, person105, 0_sequence, sequence, Apple; iPod touch 3G, 100, male, 169, 46, asphalt,  
$(projectRoot)/SampleData/0_sequence/person105/HASC1012,HASC1012.csv, person105, 0_sequence, sequence, Apple; iPhone3Gs, 100, male, 169, 46, asphalt, outc  
$(projectRoot)/SampleData/0_sequence/person106/HASC1013,HASC1013.csv, person106, 0_sequence, sequence, ATR; WAA-001, 100, male, 167, 67, asphalt, outdoor,  
$(projectRoot)/SampleData/0_sequence/person106/HASC1014,HASC1014.csv, person106, 0_sequence, sequence, Apple; iPod touch 3G, 100, male, 167, 67, asphalt,  
$(projectRoot)/SampleData/0_sequence/person106/HASC1015,HASC1015.csv, person106, 0_sequence, sequence, Apple; iPhone3Gs, 100, male, 167, 67, asphalt, outc  
$(projectRoot)/SampleData/0_sequence/person107/HASC1016,HASC1016.csv, person107, 0_sequence, sequence, ATR; WAA-001, 100, male, 174, 52, asphalt, outdoor,  
$(projectRoot)/SampleData/0_sequence/person107/HASC1017,HASC1017.csv, person107, 0_sequence, sequence, Apple; iPod touch 3G, 100, male, 174, 52, asphalt,
```

データファイル数のカウント

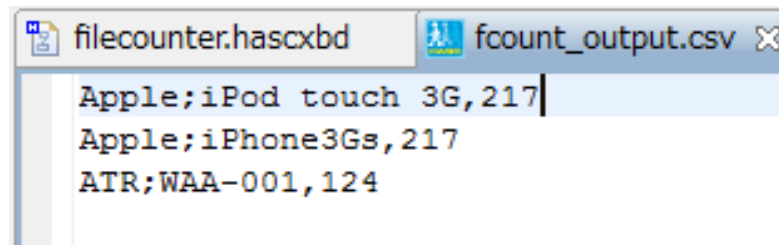
- HASCXBD>file>count>
filecounter.hascxbd
 - コーパス内に含まれる行動データファイル数をカウント
 - 設定したメタ情報の分布も閲覧可能



bean properties	
mergeMetaInf	
metaElement	TerminalType



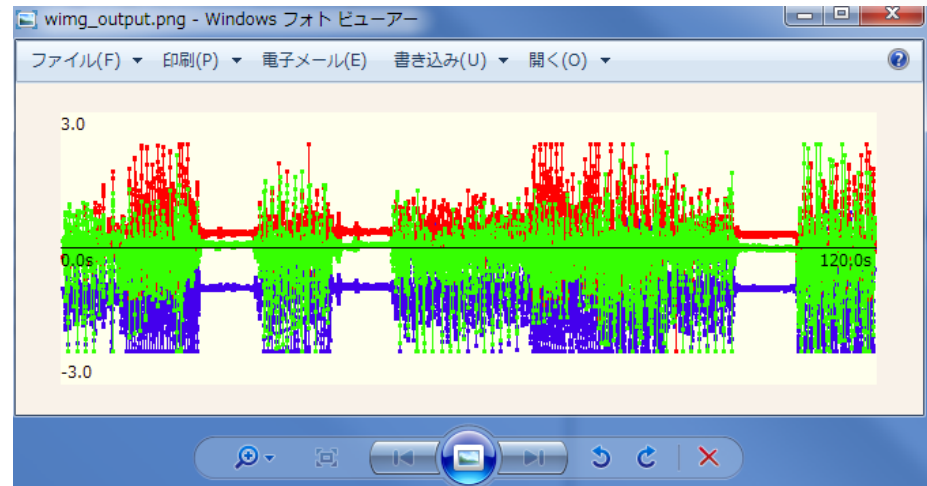
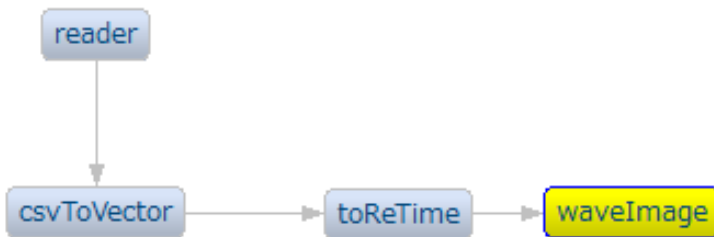
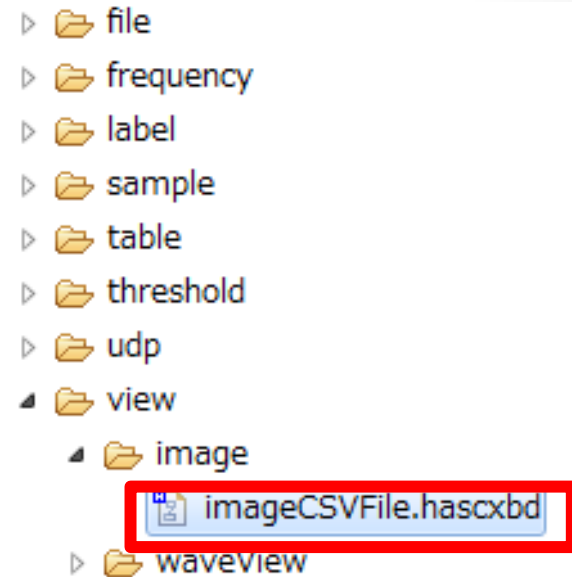
TerminalTypeに関するメタ情報のファイル数分布が閲覧できる



writerのプロパティで設定されたpathにテーブルファイルが作成される

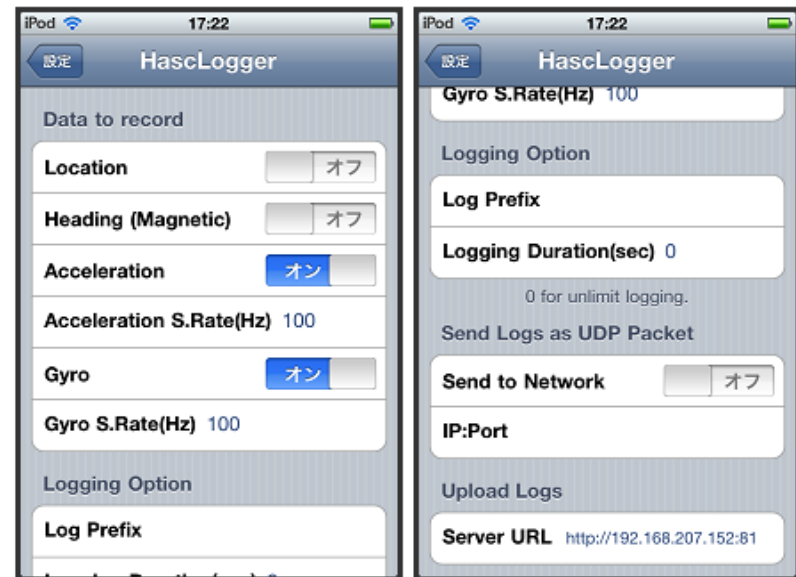
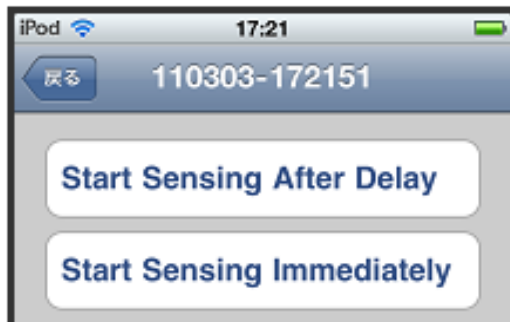
画像ファイルの取得

- HASCXBD>view>image>
imageCSVFile.hascxbd
- csvファイルを読み込む
 - 行動データ
⇒時系列波形
⇒画像ファイル



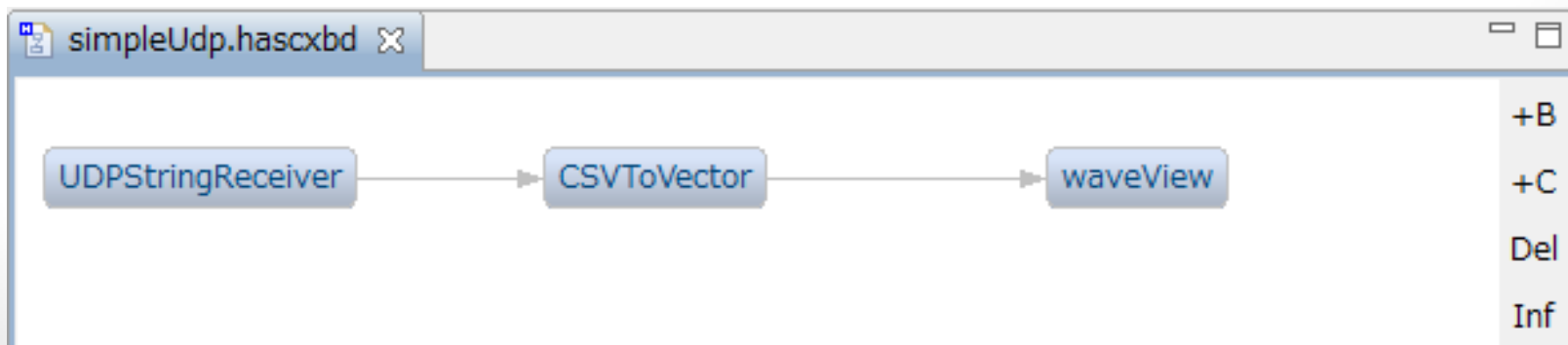
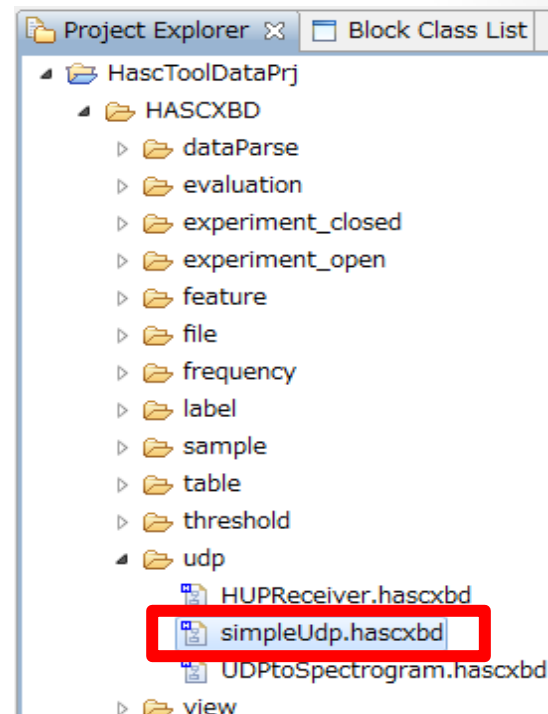
HASCToolへリアルタイム送信 ～HASC Logger側～

- 設定アプリ > HASCLogger
> Send Logs as UDP Packet > IP:Port
 - HASC Toolを動かしているPCのIPアドレス+ポート番号6666
 - Ex) 192.168.100.10:6666
- HASC Loggerを起動
 - Start Sensing ～～



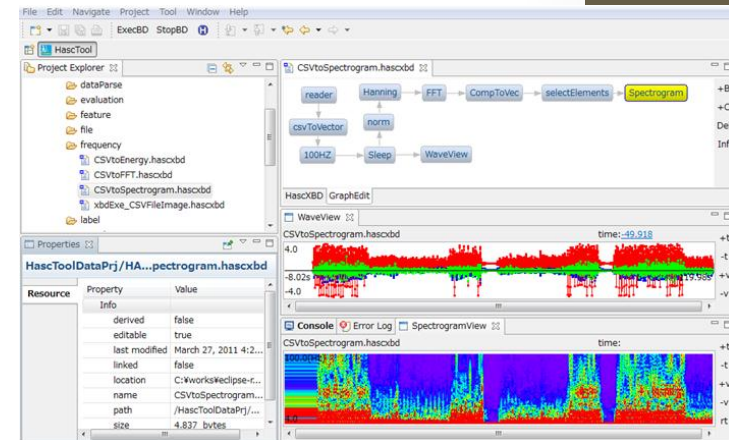
HASCToolへリアルタイム送信 ～HASC Tool側～

- HASCXBD>udp>
simpleUdp.hascxbd
- UDPでHASCLoggerから
行動データを受信
 - リアルタイムに波形を表示




HASC Toolのセットアップ

- 1. Eclipse RCPのインストール
- 2. プラグインの入手
 - Zest plugin
 - Subversive
- 3. Sourceforgeにアクセス
 - HASC Tool
 - HASCLoggerServerPlugin
 - HASC Weka
 - SampleData
- 4. HASC Toolの起動設定



Eclipse RCPのインストール

- Java SE を事前にインストール
- Eclipse RCPをダウンロード
 - <http://www.eclipse.org/downloads/packages/eclipse-rcp-and-rap-developers/heliosr1>
 - 64bit版と32bit版があるので気を付けて選択してください
- Eclipse RCPをインストール
 - ダウンロードしたファイルを解凍



The screenshot shows the Eclipse website's download page for "Eclipse for RCP and RAP Developers". The page includes a navigation menu with links like Home, Downloads, Users, Members, Committers, Resources, Projects, and About Us. A search bar is also present. The main content area features a "Package Details" section with a description: "A complete set of tools for developers who want to create Eclipse plug-ins, Rich Client or Rich Ajax Applications (RCP+RAP), plus Mylyn, and an XML editor." Below this is a "Feature List" table with two entries: "org.eclipse.cvs" and "org.eclipse.epp.package.common.feature", both at version "1.1.0". To the right, a "Download Links" section lists various operating system and architecture combinations: "Windows 32-bit", "Windows 64-bit", "Mac OS X(Cocoa 32)", "Mac OS X(Cocoa 64)", "Linux 32-bit", and "Linux 64-bit". The "Downloaded 75,444 Times" statistic is also visible.

ワークスペース
を作成する！

ex)
C:\works¥eclipse

Eclipse プラグイン

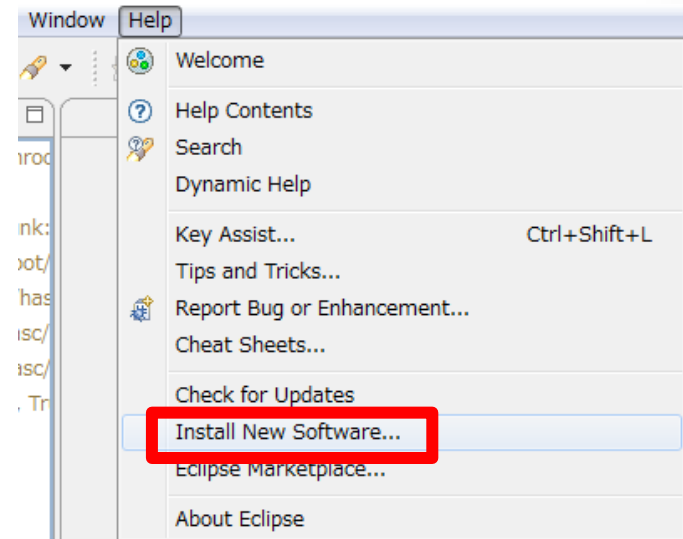
- HASCCChallenge2011>HASC Toolページ
 - **開発環境のセットアップ**を参照

- 各種プラグインを入手

- Zest plugin
- Subversive

- プラグインの入手方法

- メニューバー>Help>
Install New Software...



Zest pluginのインストール

- メニューバー>Help>Install New Software...

- Work with タブ



- Helios -

<http://download.eclipse.org/releases/helios>

- Group items by category

- チェックをはずす

- 検索ボックスに“zest”と入力

- 見つかった3つのpluginを全てinstall

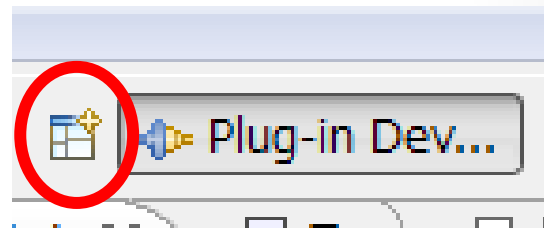
- Next⇒Next⇒acceptにチェックを入れてFinish

Subversiveのインストール

- メニューバー > Help > Install New Software...
- Work with タブ
 - Helios -
<http://download.eclipse.org/releases/helios>
- Collaboration > Subversive SVN Team Provider (Incubation)
 - チェックを入れる
 - Nextをクリックしてインストールを完了します
- Eclipseを再起動

Subversiveのインストール

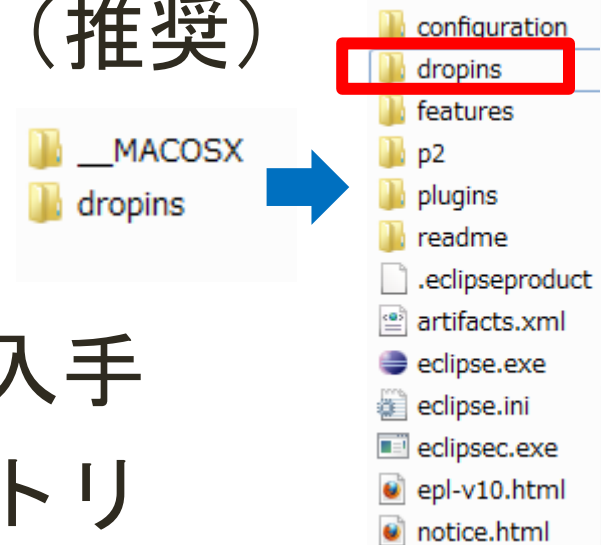
- 右上のパーспекティブボタンを押す
 - Other >
SVN Repository Exploring
 - OKボタンを押す
- **SVNKit 1.*.* Implementation** を選択
- 指示に従ってインストールを完了
- Eclipseを再起動



各種プラグインの入手

- 紹介した上記2つのプラグインは以下の方法で簡単に入手できます。（推奨）

- Zest plugin
- Subversive



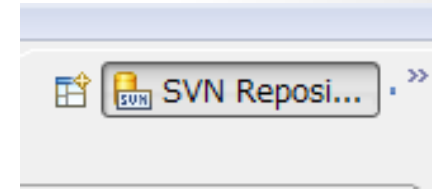
- <http://bit.ly/n3xoZD>から入手

- Eclipseのdropinsディレクトリ

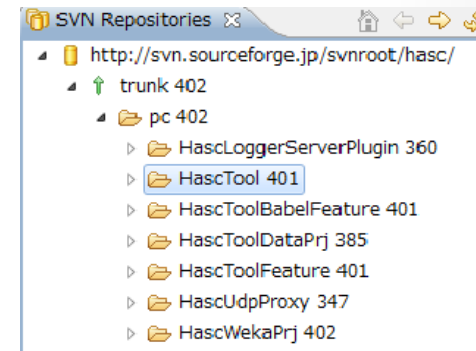
- 解凍したデータからdropinsディレクトリをdrag&drop

- Mac版とWin版があるので注意してください！
- データ上書き（データ統合）を選択

Sourceforgeにアクセス

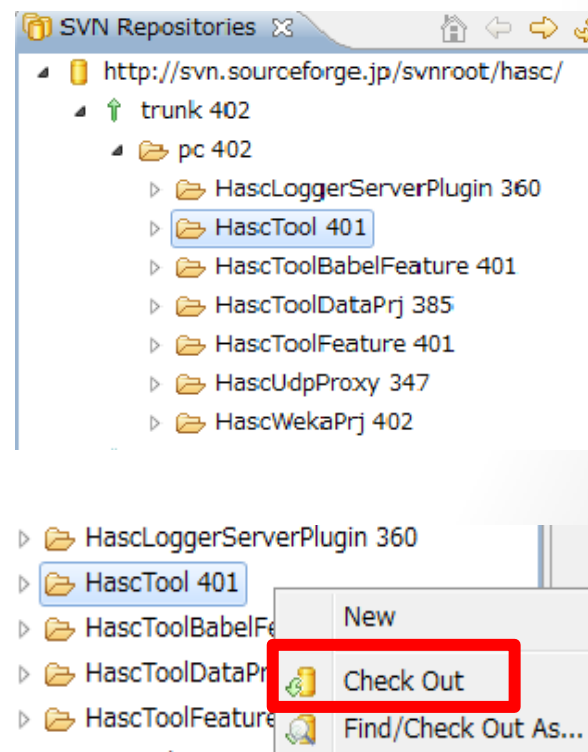


- 右上に生成されたSVN Repositoryををクリック
- SVN Repository Exploring
 - リポジトリロケーションの追加
 - File>New>Repository Location
 - New Repository Locationダイアログ
 - URL : <http://svn.sourceforge.jp/svnroot/hasc/>
 - リポジトリを登録 (Finishを押す)
 - もし何かメッセージがでた場合OK(trust)を選択
 - 登録したリポジトリの閲覧
 - trunk>pcの中を見る



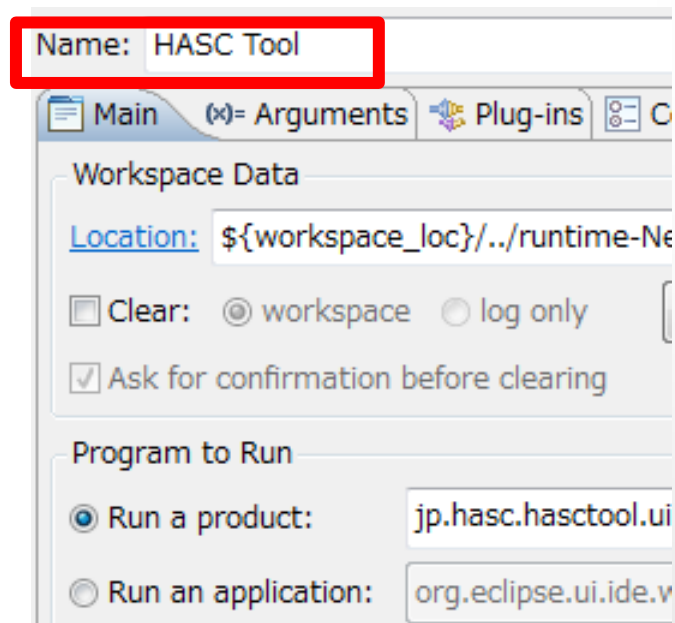
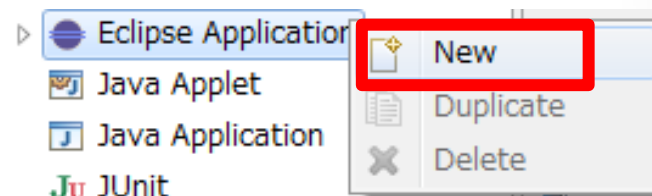
各種プロジェクトの入手

- HASCリポジトリにアクセス
 - 追加したロケーションを開く
- 以下の **4 つ** のProjectをチェックアウト
 - HascTool
 - HascLoggerServerPlugin
 - HascToolDataPrj
 - HascWekaPrj
- チェックアウトの方法
 - プロジェクトを右クリック
 - Check Outを選択



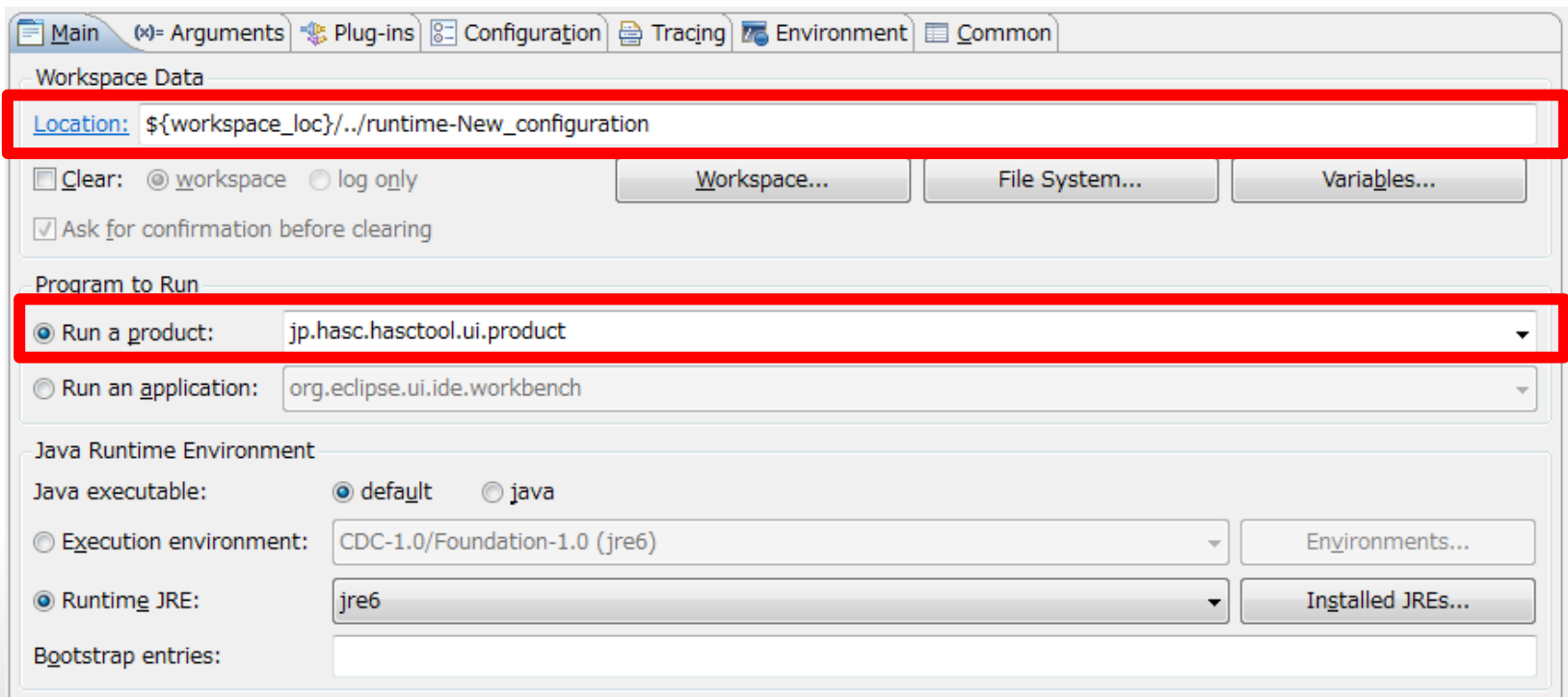
HASC Toolの起動設定

- “HASC Tool” プロジェクトを右クリック > Run As > Run Configurations を選択
 - Eclipse Application を右クリック > **New**
 - **Name** を設定
 - HASC Tool
- 実行の構成を設定
 - **Main** タブ
 - **Arguments** タブ
 - **Plug-ins** タブ



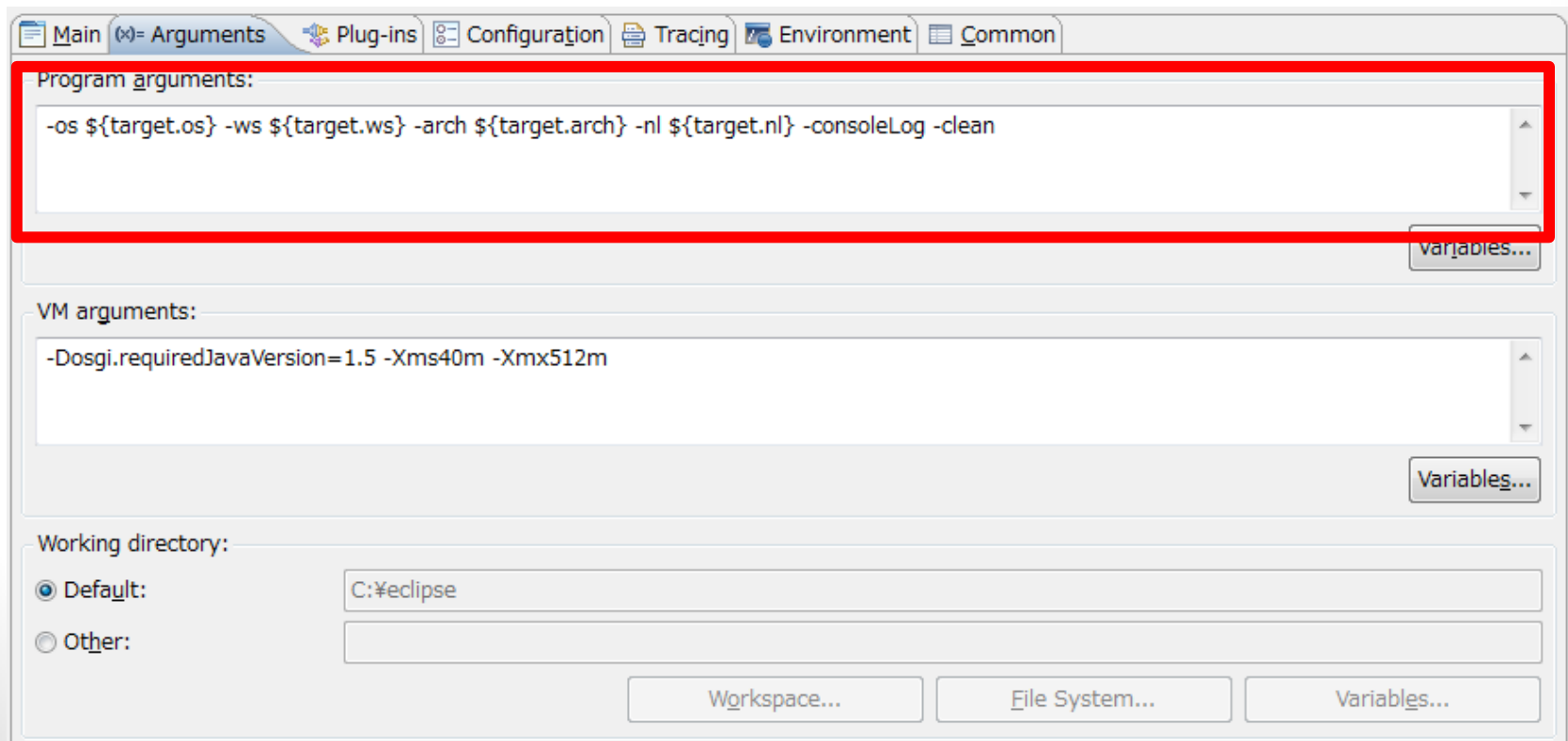
Mainタブ

- **Location**を任意設定
 - デフォルトのままでもOK!
- **Program to Run**
 - **jp.hasc.hasctool.ui.product** を設定



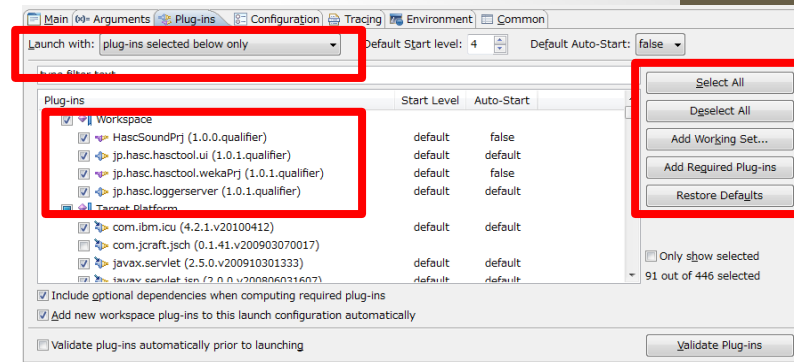
Argumentsタブ

- Program arguments
 - `-consoleLog -clean` を追加
 - 直前に空白文字も必要



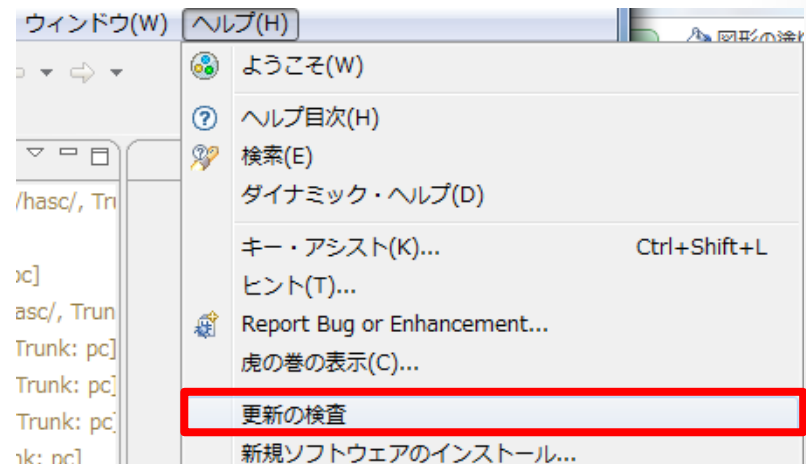
Plug-insタブ

- Launch with
 - Plug-ins selected below only
- Deselect All を選択
- Workspace内のPluginをチェック
 - `jp.hasc.hasctool.ui`
 - `jp.hasc.loggerserver`
 - `jp.hasc.hasctool.wekaPrj`
- Add Required Plug-ins を選択
- Apply を選択
- Run で実行



HASC Toolの更新

- Sourceforge版の場合
 - Eclipse側で . . .
 - 入手した各プロジェクトを右クリック
 - Team>Updateを選択
- ダウンロード版の場合
 - HASC Tool上で . . .
 - Help>更新の検査



**最新のバージョンに
更新されます！**

HASCToolDataPrjのImport

- Project Explore (左上のウィンドウ)
 - ウィンドウ内を右クリック > Import
 - General > Existing Pro~~を選択
 - Next
 - Browse
 - HasctoolDataPrjの場所を選択
 - Finishを選択

