

DAQミドルウェアのシステム・コンフィグレーション

仲吉一男

KEK

July 2009

概要

DAQミドルウェアにおけるシステム・コンフィグレーション機能について説明します。DAQミドルウェアでは、コンフィグレーション・ファイルとよばれるXML文書によるDAQシステムの記述を行い、それにより必要なDAQコンポーネントが起動され、コンポーネント間の接続が行われます。はじめに、コンフィグレーション・ファイルのスキーマについて説明し、次にJ-PARC MLF中性子実験における使用例を示します。

目次

1	はじめに	2
2	コンフィグレーション・ファイルのスキーマ	2
2.1	スキーマのモデル	2
2.2	config.xmlのスキーマ	3
2.3	コンフィグレーション・ファイルの検証	4
3	J-PARC MLF 中性子の使用例	4
3.1	param エレメント	4
A	config.xsd	6
B	コンフィグレーション・ファイル生成スクリプト	7
B.1	ファイル構成等	7
B.2	制限事項	8
B.3	使用方法	8
B.3.1	スクリプトの起動	8
B.3.2	テキストボックスへの入力	9
B.3.3	コンフィグレーション・ファイルの生成	9
B.3.4	生成したコンフィグレーションの確認	9
B.3.5	スクリプトの終了	9
B.3.6	生成したコンフィグレーション・ファイルの検証	10
C	スクリプトにより生成した config3.xml	10

1 はじめに

DAQ ミドルウェアでは、DAQ システムの構成を XML 文書（デフォルトの名前は config.xml）で記述します。XML は現在、広く普及している技術で、構造化された文書やデータを異なる情報システム間で共有することが可能です。DAQ ミドルウェアにおいては、DAQ システムを記述する広義のデータベースとして重要な意味を持ちます。コンフィギュレーション・ファイルの記述により DAQ システムに使用する DAQ コンポーネントを選択し、また DAQ コンポーネント間の接続を変更することで柔軟に DAQ システムを構築することができます。コンフィギュレーション・ファイルのスキーマは W3C の XML Schema で記述されています。コンフィギュレーション・ファイルが持っている情報は、使用する DAQ コンポーネントの名前やコンポーネントのポート情報、動作させる計算機の IP アドレス等です。DAQ コンポーネントを制御する DAQ オペレータは、システム起動時このファイルの情報により CORBA Naming Service に問い合わせることでコンポーネントのオブジェクト・リファレンスを取得します。そしてコンポーネント間のデータ入出力ポート間を接続し、コンポーネント間のデータストリームの経路を確立します。各 DAQ コンポーネントは、複数のパラメータ情報を持つことができ、“Configure” コマンド投入時に DAQ オペレータから各 DAQ コンポーネントへ送信されます。このパラメータの情報も config.xml に記述します。

2 コンフィギュレーション・ファイルのスキーマ

XML におけるスキーマとは、XML 文書の構造を記述したものです。このスキーマにより XML 文書の妥当性を検証することができます。この節ではコンフィギュレーション・ファイルのスキーマについて説明します。

2.1 スキーマのモデル

現在 config.xml のスキーマのモデルとなっているのは、J-PARC MLF 中性子 DAQ システムです。MLF 中性子の PSD 検出器系の DAQ システムでは、1 台の計算機上に図 1 のような Gatherer, Dispatcher, Logger, Monitor と呼ばれる 4 つの DAQ コンポーネントが動作します。Gatherer は複数のリードアウト・モジュールからデータを取得し、Dispatcher へ送信します。Dispatcher は取得したデータを Logger と Monitor へ送信します。Logger は取得したデータをディスクへ保存し、モニターはオンラインのモニタリングを行います。この文書では、この 4 つのコンポーネントのことを DAQ Unit と呼びます。MLF 中性子 PSD 検出器系の DAQ システムでは、1 つの DAQ Unit で 1 台の VME クレート内のリードアウト・モジュール（最大 20 台）からデータを取るよう設計されています。検出器の数が多い場合は、計算機とそこで動作する DAQ Unit を増設して対応します。config.xml のスキーマは上記のシステム構成をモデル化したものです。config.xml のスキーマにおいては、この DAQ Unit は daqGroup エレメント（要素）として記述します。DAQ Unit は、同一ホスト内で

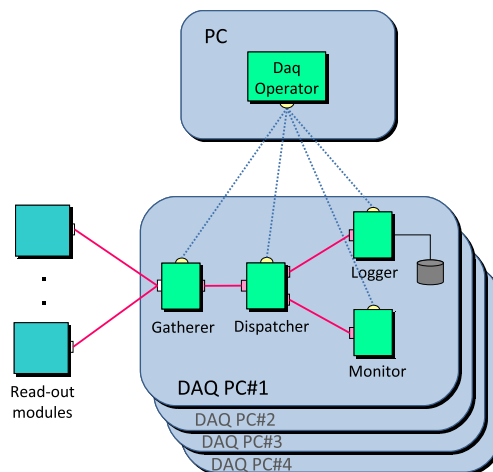


図 1: MLF 中性子の DAQ モデル

動作しますが、DAQ コンポーネントの性質からそれぞれがネットワーク上の分散した計算機上にあっても動作可能です。また、config.xml の daqGroup においても同一ホスト上という規定はなく、それぞれのコンポーネントが実際はネットワーク上に分散している場合でも、ひとつの daqGroup として記述することができます。

2.2 config.xml のスキーマ

XML のスキーマ言語としては数種類存在しますが、W3C の XML Schema を使用しています。config.xml のスキーマ config.xsd を付録 A に載せました。

エレメント名	属性	説明
configInfo	なし	ルート・エレメント
daqOperator	なし	子エレメントとして 1 個の hostAddr をもつ
daqGroups	なし	子エレメントとして 1 個以上の daqGroup をもつ
daqGroup	gid	グループ ID を任意の文字列で指定する 例: <daqGroup gid="group0">
components	なし	子エレメントとして 1 個以上の daqComponent をもつ
component	cid	MyModuleInit() の manager->createComponent("xxx") で使用した文字列 "xxx" に "0" を付加した文字列 例: <component cid="Gatherer0">
hostAddr	なし	コンポーネントを起動させるホストの IP Address 例: <hostAddr>192.168.1.206</hostAddr>
hostPort	なし	コンポーネントのリモート起動に使用する xinetd のポート番号 例: <hostPort>50000</hostPort> 50000 番を使用する
instName	なし	コンポーネントのインスタンス名。cid に ".rtc" を付加する 例: <instName>Gatherer0.rtc</instName>
execPath	なし	コンポーネントの実行形式ファイルの絶対パス
confFile	なし	コンポーネントの使用する rtc.conf ファイルの絶対パス
startOrd	なし	コンポーネントのスタートコマンド投入の際の順番
inPorts	なし	子エレメントとして 0 個以上の inPort をもつ
inPort	from	registerInPort ("xxx", m_InPort) で登録したの InPort の名前 "xxx" を指定 "from" には接続する OutPort を指定する。形式は cid : outPort 例: <inPort from="Gatherer0:gatherer_out">dispatcher_in</inPort>
outPorts	なし	子エレメントとして 0 個以上の outPort をもつ
outPort	なし	registerOutPort("xxx", m_OutPort) で指定した outPort の名前 "xxx" を指定 例: <outPort>gatherer_out</outPort>
params	なし	子エレメントとして 0 個以上の param をもつ
param	pid	pid 属性にユニークなパラメータ名を指定する 例: <param pid="srcAddr">192.168.0.80</param>

表 1: コンフィグレーション・ファイルに使用するエレメント

config.xml で使用可能なエレメントについて説明します。表 1 にエレメント名をまとめました。図 2 にその構造を示します。configInfo というルートエレメントの中に、daqOperator と daqGroups というエレメントがあります。daqGroups の中には複数の daqGroup が存在します。daqGroup は複数の component から構成されます。component には、hostAdd, hostPort, instName, execPat, configFile, startOrd, inPorts, outPorts, param エレメントがあります。inPorts, outPorts, params はそれぞれ、複数の inPort, outPort, param を持ちます。param エレメントは、各コンポーネントに特有なパラメータ等を記述する際に使用します。このパラメータは、DAQ オペレータから configure コマンドと一緒に、名前と値のリストとして各コンポーネントへ送られます。各コンポーネントでは、その名前をキーにリストから値を取得し、設定を行います。パラメータとその設定については後述します。

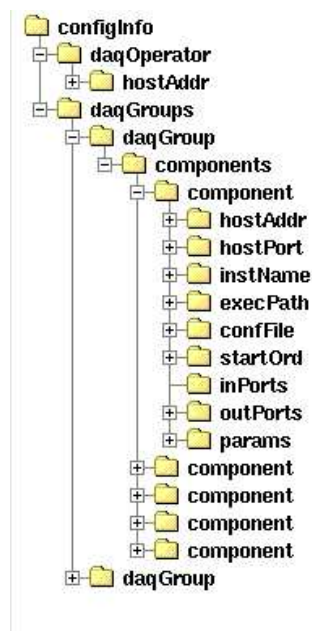


図 2: config.xml の構造

2.3 コンフィグレーション・ファイルの検証

次の節で説明する J-PARC MLF 中性子では、GUI ツールによるコンフィグレーション・ファイルの生成を推奨しています。使用するコンポーネントの数が少ない場合等は、テキストエディタによりコンフィグレーションを作成可能です。そのような場合は、下記のように xmllint というコマンドラインで使用する XML tool を使用してコンフィグレーション・ファイルの検証をしてください。作成したコンフィグレーション・ファイル名は myconfig.xml、またスキーマ・ファイル config.xsd と myconfig.xml はカレントディレクトリ内にあるとします。

```
xmllint --noout --schema ./config.xsd ./myconfig.xml
```

3 J-PARC MLF 中性子の使用例

前節のコンフィグレーション・ファイルの使用例として J-PARC MLF 中性子 PSD 検出器系 DAQ システムのコンフィグレーション・ファイルについて説明します。中性子 PSD 検出器系では、Gatherer, Dispatcher, Logger, Monitor という 4 つのコンポーネントで DAQ Unit という基本構成になっています。Gatherer は NEUNET というリードアウトモジュールへネットワーク接続を行いデータを取得して後段のコンポーネントへ送信します。

3.1 param エレメント

前述の param エレメントを使用して実験に特有なパラメータを記述しています。複数のパラメータは、属性 pid により区別します。Gatherer コンポーネントのパラメータの取得を例に説明します。Gatherer のパラメータとしては、表 2 にある、daqId, srcAddr があります。例えば、config.xml に次のような記述があった場合を考えます。

```

<component cid="Gatherer0">
  <hostAddr>192.168.1.206</hostAddr>
  <hostPort>50000</hostPort>
  <instName>Gatherer0.rtc</instName>
  ...
  <params>
    <param pid="daqId">0</param>
    <param pid="srcAddr">192.168.0.80</param>
  </params>
</component>

```

この場合、パラメータの NVList としては "daqId", "0", "srcAddr", "192.168.0.80" となります。"daqId" は、DAQ コンポーネントが動作する計算機の ID、"srcAddr" は、NEUNET という PSD 検出器 リードアウト用モジュールの IP アドレスです。DAQ オペレータから、NVList が Gatherer に "Configure" コマンドとともに送られ、Gatherer は、自分が必要なパラメータの名前を知っているので parse_params() という関数で "daqId", "srcAddr" をキーにリストを検索して値を取得します。

pid	値	意味
daqId	0 以上の整数	DAQ ID を指定する 例: <param pid="daqId">0</param>
srcAddr	文字列	NEUNET の IP アドレスを指定する。複数台指定することが可能 例: <param pid="srcAddr">192.168.0.80</param>
eventByteSize	0 以上の整数	1 イベントのバイトサイズを指定する。 例: <param pid="eventByteSize">8</param>
isLogging	文字列 (yes, no)	取得したデータを Logger で保存するかどうか指定する。 例: <param pid="isLogging">yes</param>
maxFileSizeInMegaByte	0 以上の整数	データ保存ファイル用の最大ファイルサイズ (MByte)。この値以上になるとファイルを保存し新たにファイル名の枝番が 1 つ増えたファイルを作る。 例: <param pid="maxFileSizeInMegaByte">1024</param>
instId	英大文字 3 文字	装置 ID 例: <param pid="instId">ABC</param>
dirName	文字列	取得データ保存用のディレクトリを指定する。ディレクトリがなければエラーとなる。 例: <param pid="dirName">/tmp</param>
samplingRate	1 以上の整数	Dispatcher から Monitor へ送るデータの割合を指定。全データを Monitor へ転送する場合は "1" を 10 分の 1 のデータを転送する場合は "10" を指定する。この値は Dispatcher および Monitor のパラメータとして同一のものを指定する 例: <param pid="samplingRate">dispatcher_out2/10</param>
gnuplot_path	文字列	gnuplot executable の絶対パス 例: <param pid="gnuplot_path">/home/daq/gnuplot/bin/gnuplot</param>
png_output_dir	文字列	各種ヒストグラム png ファイルを出力するディレクトリ名 例: <param pid="png_output_dir">/home/daq/Data/png</param>
num_of_psd_per_module	1 以上の整数	NEUNET モジュール 1 台あたりの PSD 数 例: <param pid="num_of_psd_per_module">8</param>

表 2: コンフィグレーション・ファイルに使用するパラメータ

使用するコンポーネントの数、リードアウト・モジュールの数が多くなると記述する内容が増え、テキストエディタを使って config.xml を作成すると不具合が生じることがあります。このような障害を減らすため、J-PARC MLF 中性子 PSD 検出器系のコンフィグレーション・ファイル生成スクリプトがあります。詳細は付録 B で説明します。

A config.xsd

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      DAQ-Components Configuration schema for DAQ-Middleware.
      Copyright 2008 Kazuo Nakayoshi. All rights reserved.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="configInfo" type="ConfigInfoType" />
  <xsd:element name="daqOperator" type="DaqOperatorType" />
  <xsd:element name="daqGroups" type="DaqGroupsType" />
  <xsd:element name="daqGroup" type="DaqGroupType" />
  <xsd:element name="components" type="ComponentsType" />
  <xsd:element name="component" type="ComponentType" />
  <xsd:element name="inPorts" type="InPortsType" />
  <xsd:element name="outPorts" type="OutPortsType" />
  <xsd:element name="params" type="ParamsType" />
  <xsd:element name="hostAddr" type="xsd:string" />
  <xsd:element name="hostPort" type="xsd:string" />
  <xsd:element name="instName" type="xsd:string" />
  <xsd:element name="execPath" type="xsd:string" />
  <xsd:element name="confFile" type="xsd:string" />
  <xsd:element name="startOrd" type="xsd:string" />
  <xsd:element name="inPort" type="InPortType" />
  <xsd:element name="outPort" type="xsd:string" />
  <xsd:element name="param" type="ParamType" />

  <xsd:complexType name="InPortType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="from" type="xsd:string"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:complexType name="ParamType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="pid" type="xsd:string" use="required" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:complexType name="InPortsType">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="inPort" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="OutPortsType">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="outPort" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="ParamsType">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="param" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="ComponentType">
    <xsd:sequence>
      <xsd:element ref="hostAddr" />
      <xsd:element ref="hostPort" />
      <xsd:element ref="instName" />
      <xsd:element ref="execPath" />
      <xsd:element ref="confFile" />
      <xsd:element ref="startOrd" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

```

    <xsd:element ref="inPorts" />
    <xsd:element ref="outPorts" />
    <xsd:element ref="params" />
  </xsd:sequence>
  <xsd:attribute name="cid" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="ComponentsType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element ref="component" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DaqGroupType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element ref="components" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="gid" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="DaqGroupsType">
  <xsd:sequence>
    <xsd:element ref="daqGroup" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DaqOperatorType">
  <xsd:sequence>
    <xsd:element ref="hostAddr" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConfigInfoType">
  <xsd:sequence>
    <xsd:element ref="daqOperator" />
    <xsd:element ref="daqGroups" />
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

B コンフィグレーション・ファイル生成スクリプト

これまで、ユーザが手動で config.xml を書いていました。この方法では、勘違いやミスにより DAQ コンポーネントが起動しない、エラーが起きる等の例がありました。そこで config.xml を生成する Python スクリプトを用います。このスクリプトを使用することで次の利点があります。

- このスクリプトにより生成したコンフィグレーション・ファイルの XML の構造、DAQ ミドルウェアに関する必要事項について保証します
- ユーザが入力すべき情報は最小限にし、入力ミスを最小限にします

B.1 ファイル構成等

コンフィグレーション・ファイル生成スクリプトは、次の 3 つのファイルから構成されています。

- mkconfig_gui.py: Tkinter による GUI 処理
- genconf.py: XML 生成処理
- dom_tree.py: 生成したファイルをツリー表示

この生成スクリプトを使用する場合は、Tkinter が必要です。下記の環境で開発、動作テストを行いました。

- python-2.4.3-21.el5
- tkinter-2.4.3-21.el5

B.2 制限事項

- このスクリプトは、DAQ ミドルウェア for MLF 7月版以降に対応した PSD 検出器系の config.xml を生成します。
- 既存の config.xml を読み込んで編集する機能はありません。
- GUI による入力項目を減らすため、次の項目は genconf.py 中で固定値となっています。使用する環境にあわせて変更してください。
 - MAX_FILE_BSIZE_IN_MB = '1024' # イベント保存用ファイルはこのサイズ（単位メガバイト）以上になると分割され枝番がつく
 - MON_GNUPLOT_PATH = '/home/daq/gnuplot/bin/gnuplot' # Monitor コンポーネントで称する gnuplot の絶対パス
 - MON_PNG_OUT_DIR = '/home/daq/Data/png' # Monitor コンポーネントが PNG 形式のヒストグラム・ファイルを書き出すディレクトリ
 - NUM_PSD_PER_MOD = '8' # 1 NEUNET の PSD のチャンネル数
 - MON_UPDATE_RATE = '500' # Monitor コンポーネントのヒストグラム更新レート（相対値）
 - GATENET_DEFAULT_ADDR = '192.168.0.15' # GATENET の IP アドレス
- このスクリプトは自由に修正・変更してください。また修正・変更した場合は、お知らせください。

B.3 使用方法

スクリプトの使用方法について説明します。

B.3.1 スクリプトの起動

\$ はプロンプト。CPU DAQ の数（必須）と出力ファイル名を指定する。出力ファイル名が指定されなければ config.xml.generated というファイルが生成されます。

```
$ ./mkconfig_gui.py -n CPU DAQ の数 -o 出力ファイル名
```

図 3 に、

```
./mkconfig_gui.py -n 3 -o config3.xml
```

を入力した際に現れるフレームを示します。

B.3.2 テキストボックスへの入力

CPU DAQ の数を”3”と指定したので、画面には DAQ#0, DAQ#1, DAQ#2 と 3つのグループに関する入力欄が現れます。フレーム上から 1 番目と 2 番目は、グループ共通の情報です。

- 1 番目は、装置 ID (各装置ですでに決まっている) を入力します。装置 ID はイベントデータを保存するディレクトリ名、ファイル名に使用されます。
- 2 番目は、DAQ オペレータを起動する計算機の IP アドレスをドット区切りで (192.168.1.11 のように) 指定します。
- 次に各 CPU DAQ の情報を入力します。”DAQ Group name”は、スペースを含まない任意の文字列を入れてください。例えば、daqgroup0 等。”CPU DAQ IP addr.”は、対応する IP アドレスをドット区切りで入力します。
- ”NEUNETs addr.”は、この CPU DAQ で読み出す NEUNET の IP アドレスの第 4 オクテットを入力します。NEUNET のフロントパネルのロータリースイッチで設定可能な値は第 4 オクテットのみだからです。残りのアドレスは”192.168.0”と固定です。指定の際、”,”(コンマ)”を使って”16,18,20”と飛び飛びの値の指定が可能です。連続する値は”-(ハイフン)”を使って”16-19”と指定できます。
- ”Directory of data logging”はイベントデータを保存するディレクトリを絶対パスで指定します。複数の Gatherer から NFS でマウントしたディスクにデータを保存する場合は、それぞれの CPU DAQ で指定するディレクトリがユニークになるようにします。例えば、/nfs/cpudaq0, /nfs/cpudaq1, /nfs/cpudaq2 等。
- フレームの最下段にあるボタンの上に GATENET の情報を指定する欄があります。GATENET コンポーネントの制御をどの CPU DAQ で行うかを指定します。この例だと、”not use”, ”DAQ#0”, ”DAQ#1”, ”DAQ#2”のうちから一つを選択します。

B.3.3 コンフィグレーション・ファイルの生成

全ての欄を入力または選択した後、”Create”ボタンを押します。カレントディレクトリに指定した config3.xml というファイルが作られます。

B.3.4 生成したコンフィグレーションの確認

ボタンを押した後、図 4 のような XML の構造を可視化した図が現れます。その構造で展開可能なノードは図 5 のように展開することができます。ノードを展開して生成したコンフィグレーションに問題がないか確かめてください。

B.3.5 スクリプトの終了

コンフィグレーション確認用のフレームの ”Close”ボタンを押してフレームを閉じます。テキストボックスのあるフレームの ”Quit”ボタンを押してスクリプト終了します。このスクリプトにより生成した config3.xml を C に示します。

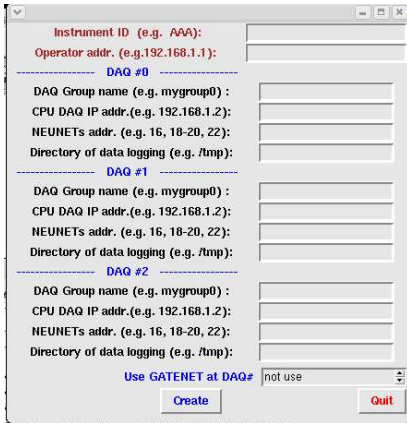


図 3: config.xml 生成用 GUI のフレーム

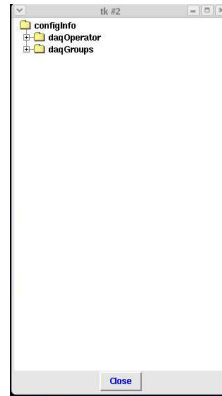


図 4: 生成ファイル確認用フレーム (展開前)

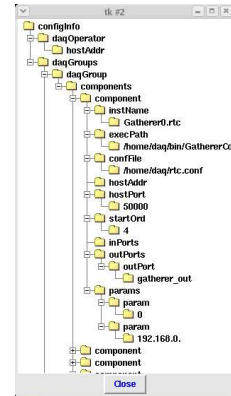


図 5: 生成ファイル確認用フレーム (展開後)

B.3.6 生成したコンフィグレーション・ファイルの検証

2.3 で示した方法で、生成したコンフィグレーション・ファイルの検証を行ってください。

C スクリプトにより生成した config3.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet href="config.xsl" type="text/xsl"?>
<configInfo>
  <daqOperator>
    <hostAddr>192.168.1.1</hostAddr>
  </daqOperator>
  <daqGroups>
    <daqGroup gid="group0">
      <components>
        <component cid="Gatherer0">
          <hostAddr>192.168.1.2</hostAddr>
          <hostPort>50000</hostPort>
          <instName>Gatherer0.rtc</instName>
          <execPath>/home/daq/DaqComponents/bin/GathererComp</execPath>
          <confFile>/home/daq/DaqComponents/rtc.conf</confFile>
          <startOrd>4</startOrd>
          <inPorts>
            </inPorts>
          <outPorts>
            <outPort>gatherer_out</outPort>
          </outPorts>
          <params>
            <param pid="daqId">0</param>
            <param pid="srcAddr">192.168.0.16</param>
            <param pid="srcAddr">192.168.0.18</param>
            <param pid="srcAddr">192.168.0.19</param>
            <param pid="srcAddr">192.168.0.20</param>
            <param pid="srcAddr">192.168.0.22</param>
          </params>
        </component>
        <component cid="Dispatcher0">
          <hostAddr>192.168.1.2</hostAddr>
          <hostPort>50000</hostPort>
          <instName>Dispatcher0.rtc</instName>
          <execPath>/home/daq/DaqComponents/bin/DispatcherComp</execPath>
          <confFile>/home/daq/DaqComponents/rtc.conf</confFile>
          <startOrd>3</startOrd>
          <inPorts>
            <inPort from="Gatherer0:gatherer_out">dispatcher_in</inPort>
          </inPorts>
          <outPorts>
            <outPort>dispatcher_out1</outPort>
            <outPort>dispatcher_out2</outPort>
          </outPorts>
          <params>
            <param pid="eventByteSize">8</param>
            <param pid="samplingRate">dispatcher_out2/10</param>
          </params>
        </component>
      </components>
    </daqGroup>
  </daqGroups>
</configInfo>
```

```

<component cid="Logger0">
  <hostAddr>192.168.1.2</hostAddr>
  <hostPort>50000</hostPort>
  <instName>Logger0.rtc</instName>
  <execPath>/home/daq/daqComponents/bin/LoggerComp</execPath>
  <confFile>/home/daq/daqComponents/rtc.conf</confFile>
  <startOrd>1</startOrd>
  <inPorts>
    <inPort from="Dispatcher0:dispatcher_out1">logger_in</inPort>
  </inPorts>
  <outPorts>
  </outPorts>
  <params>
    <param pid="daqId">0</param>
    <param pid="srcAddr">192.168.0.16</param>
    <param pid="srcAddr">192.168.0.18</param>
    <param pid="srcAddr">192.168.0.19</param>
    <param pid="srcAddr">192.168.0.20</param>
    <param pid="srcAddr">192.168.0.22</param>
    <param pid="instId">ABC</param>
    <param pid="dirName">/tmp/group0</param>
    <param pid="eventByteSize">8</param>
    <param pid="isLogging">yes</param>
    <param pid="maxFileSizeInMegaByte">1024</param>
  </params>
</component>
<component cid="Monitor0">
  <hostAddr>192.168.1.2</hostAddr>
  <hostPort>50000</hostPort>
  <instName>Monitor0.rtc</instName>
  <execPath>/home/daq/daqComponents/bin/MonitorComp</execPath>
  <confFile>/home/daq/daqComponents/rtc.conf</confFile>
  <startOrd>2</startOrd>
  <inPorts>
    <inPort from="Dispatcher0:dispatcher_out2">monitor_in</inPort>
  </inPorts>
  <outPorts>
  </outPorts>
  <params>
    <param pid="daqId">0</param>
    <param pid="srcAddr">192.168.0.16</param>
    <param pid="srcAddr">192.168.0.18</param>
    <param pid="srcAddr">192.168.0.19</param>
    <param pid="srcAddr">192.168.0.20</param>
    <param pid="srcAddr">192.168.0.22</param>
    <param pid="samplingRate">dispatcher_out2/10</param>
    <param pid="gnuplot_path">/home/daq/gnuplot/bin/gnuplot</param>
    <param pid="png_output_dir">/home/daq/Data/png</param>
    <param pid="num_of_psd_per_module">8</param>
  </params>
</component>
</components>
</daqGroup>
<daqGroup gid="group1">
  <components>
    <component cid="Gatherer0">
      <hostAddr>192.168.1.3</hostAddr>
      <hostPort>50000</hostPort>
      <instName>Gatherer0.rtc</instName>
      <execPath>/home/daq/daqComponents/bin/GathererComp</execPath>
      <confFile>/home/daq/daqComponents/rtc.conf</confFile>
      <startOrd>8</startOrd>
      <inPorts>
      </inPorts>
      <outPorts>
        <outPort>gatherer_out</outPort>
      </outPorts>
      <params>
        <param pid="daqId">1</param>
        <param pid="srcAddr">192.168.0.23</param>
        <param pid="srcAddr">192.168.0.24</param>
        <param pid="srcAddr">192.168.0.25</param>
      </params>
    </component>
    <component cid="Dispatcher0">
      <hostAddr>192.168.1.3</hostAddr>
      <hostPort>50000</hostPort>
      <instName>Dispatcher0.rtc</instName>
      <execPath>/home/daq/daqComponents/bin/DispatcherComp</execPath>
      <confFile>/home/daq/daqComponents/rtc.conf</confFile>
      <startOrd>7</startOrd>
      <inPorts>
        <inPort from="Gatherer0:gatherer_out">dispatcher_in</inPort>
      </inPorts>
      <outPorts>
        <outPort>dispatcher_out1</outPort>
        <outPort>dispatcher_out2</outPort>
      </outPorts>
      <params>
        <param pid="eventByteSize">8</param>
        <param pid="samplingRate">dispatcher_out2/10</param>
      </params>
    </component>
    <component cid="Logger0">
      <hostAddr>192.168.1.3</hostAddr>

```

```

<hostPort>50000</hostPort>
<instName>Logger0.rtc</instName>
<execPath>/home/daq/DAQComponents/bin/LoggerComp</execPath>
<confFile>/home/daq/DAQComponents/rtc.conf</confFile>
<startOrd>5</startOrd>
<inPorts>
  <inPort from="Dispatcher0:dispatcher_out1">logger_in</inPort>
</inPorts>
<outPorts>
</outPorts>
<params>
  <param pid="daqId">1</param>
  <param pid="srcAddr">192.168.0.23</param>
  <param pid="srcAddr">192.168.0.24</param>
  <param pid="srcAddr">192.168.0.25</param>
  <param pid="instId">ABC</param>
  <param pid="dirName">/tmp/group1</param>
  <param pid="eventByteSize">8</param>
  <param pid="isLogging">yes</param>
  <param pid="maxFileSizeInMegaByte">1024</param>
</params>
</component>
<component cid="Monitor0">
  <hostAddr>192.168.1.3</hostAddr>
  <hostPort>50000</hostPort>
  <instName>Monitor0.rtc</instName>
  <execPath>/home/daq/DAQComponents/bin/MonitorComp</execPath>
  <confFile>/home/daq/DAQComponents/rtc.conf</confFile>
  <startOrd>6</startOrd>
  <inPorts>
    <inPort from="Dispatcher0:dispatcher_out2">monitor_in</inPort>
  </inPorts>
  <outPorts>
</outPorts>
  <params>
    <param pid="daqId">1</param>
    <param pid="srcAddr">192.168.0.23</param>
    <param pid="srcAddr">192.168.0.24</param>
    <param pid="srcAddr">192.168.0.25</param>
    <param pid="samplingRate">dispatcher_out2/10</param>
    <param pid="gnuplot_path">/home/daq/gnuplot/bin/gnuplot</param>
    <param pid="png_output_dir">/home/daq/Data/png</param>
    <param pid="num_of_psd_per_module">8</param>
  </params>
</component>
</components>
</daqGroup>
<daqGroup gid="group2">
  <components>
    <component cid="Gatherer0">
      <hostAddr>192.168.1.4</hostAddr>
      <hostPort>50000</hostPort>
      <instName>Gatherer0.rtc</instName>
      <execPath>/home/daq/DAQComponents/bin/GathererComp</execPath>
      <confFile>/home/daq/DAQComponents/rtc.conf</confFile>
      <startOrd>12</startOrd>
      <inPorts>
</inPorts>
      <outPorts>
        <outPort>gatherer_out</outPort>
      </outPorts>
      <params>
        <param pid="daqId">2</param>
        <param pid="srcAddr">192.168.0.26</param>
        <param pid="srcAddr">192.168.0.29</param>
      </params>
    </component>
    <component cid="Dispatcher0">
      <hostAddr>192.168.1.4</hostAddr>
      <hostPort>50000</hostPort>
      <instName>Dispatcher0.rtc</instName>
      <execPath>/home/daq/DAQComponents/bin/DispatcherComp</execPath>
      <confFile>/home/daq/DAQComponents/rtc.conf</confFile>
      <startOrd>11</startOrd>
      <inPorts>
        <inPort from="Gatherer0:gatherer_out">dispatcher_in</inPort>
      </inPorts>
      <outPorts>
        <outPort>dispatcher_out1</outPort>
        <outPort>dispatcher_out2</outPort>
      </outPorts>
      <params>
        <param pid="eventByteSize">8</param>
        <param pid="samplingRate">dispatcher_out2/10</param>
      </params>
    </component>
    <component cid="Logger0">
      <hostAddr>192.168.1.4</hostAddr>
      <hostPort>50000</hostPort>
      <instName>Logger0.rtc</instName>
      <execPath>/home/daq/DAQComponents/bin/LoggerComp</execPath>
      <confFile>/home/daq/DAQComponents/rtc.conf</confFile>
      <startOrd>9</startOrd>
      <inPorts>
        <inPort from="Dispatcher0:dispatcher_out1">logger_in</inPort>

```

```

    </inPorts>
    <outPorts>
    </outPorts>
    <params>
      <param pid="daqId">2</param>
      <param pid="srcAddr">192.168.0.26</param>
      <param pid="srcAddr">192.168.0.29</param>
      <param pid="instId">ABC</param>
      <param pid="dirName">/tmp/group2</param>
      <param pid="eventByteSize">8</param>
      <param pid="isLogging">yes</param>
      <param pid="maxFileSizeInMegaByte">1024</param>
    </params>
  </component>
  <component cid="Monitor0">
    <hostAddr>192.168.1.4</hostAddr>
    <hostPort>50000</hostPort>
    <instName>Monitor0.rtc</instName>
    <execPath>/home/daq/daqComponents/bin/MonitorComp</execPath>
    <confFile>/home/daq/daqComponents/rtc.conf</confFile>
    <startOrd>10</startOrd>
    <inPorts>
      <inPort from="Dispatcher0:dispatcher_out2">monitor_in</inPort>
    </inPorts>
    <outPorts>
    </outPorts>
    <params>
      <param pid="daqId">2</param>
      <param pid="srcAddr">192.168.0.26</param>
      <param pid="srcAddr">192.168.0.29</param>
      <param pid="samplingRate">dispatcher_out2/10</param>
      <param pid="gnuplot_path">/home/daq/gnuplot/bin/gnuplot</param>
      <param pid="png_output_dir">/home/daq/Data/png</param>
      <param pid="num_of_psd_per_module">8</param>
    </params>
  </component>
  <component cid="Gatenet0">
    <hostAddr>192.168.1.4</hostAddr>
    <hostPort>50000</hostPort>
    <instName>Gatenet0.rtc</instName>
    <execPath>/home/daq/daqComponents/bin/GatenetComp</execPath>
    <confFile>/home/daq/daqComponents/rtc.conf</confFile>
    <startOrd>13</startOrd>
    <inPorts>
    </inPorts>
    <outPorts>
    </outPorts>
    <params>
      <param pid="gatenetAddr">192.168.0.15</param>
    </params>
  </component>
</components>
</daqGroup>
</daqGroups>
</configInfo>

```