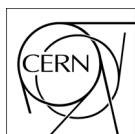




Job Control for RCMS Users Manual

Version 0.6
9/6/06

| | |
|----------------------|----------------------|
| Version: | 0.6 |
| Date: | 9/6/06 |
| Authors: | A. Oh, V. Brigljevic |
| CI identifier | xxxxxxx_D_xxxxxx |



Revision History

| Date | Version | Description | Author |
|--------------------|---------|--|---------------|
| March 15, 2004 | 0.1 | Document creation | A. Oh |
| May 25, 2004 | 0.2 | Addition of filter farm mode | V. Brigljevic |
| March 23, 2005 | 0.3 | Some changes for XDAQ3 | A. Oh |
| May 18, 2005 | 0.4 | Some changes for XDAQ3.1 | A. Oh |
| July 3, 2006 | 0.5 | Changes for XDAQ3.5 Change of installation, use xdaqd to start. | A. Oh |
| September, 6, 2006 | 0.6 | Changes for XDAQ3.7 notification mechanism for crashed jobs, ascii jobid, new web interface | |

CI Record

| Field | Description |
|----------------------|--------------|
| CI Identifier | |
| Description | |
| Submission Date | |
| Submitted By | |
| Components | |
| Dependencies/Related | RCMS |
| External Identifier | |
| Point of Contact | Alexander Oh |
| Comments | None |
| Physical Location | |

Table of Contents

| | | |
|----------|--|------------------------------|
| 1 | Overview | 5 |
| 1.1 | Introduction | 5 |
| 2 | Getting Started | 5 |
| 2.1 | System Requirements | 5 |
| 2.2 | Obtaining the Software | 5 |
| 2.3 | Installing JobControl | 6 |
| 3 | Configuration | 6 |
| 4 | Operation in Event Builder Mode | 7 |
| 5 | Operation in filter farm mode | Error! Bookmark not defined. |
| 6 | References | 8 |

Job Control for RCMS

1 Overview

In this chapter:

- Overview
- Getting started
- Operating the software

1.1 Introduction

One component of the Run Control and Monitoring System (RCMS) [1] is the Job Control service, which is required to manage processes running on its resources. The purpose of the Job Control Application (JC) described in this document is to launch and terminate XDAQ executables [2]. It is designed to work with the RCMS.

The JC is a small XDAQ application running on a XDAQ executive, which is launched at boot time. This XDAQ executive is reserved to run only the JC application. The RCMS is responsible for the configuration and control of the application.

The JC application can launch another XDAQ executive with its own set of environment variables and terminate it.

2 Getting Started

2.1 System Requirements

The JC software is an application in the XDAQ framework under the Linux operating system. It requires a working installation of XDAQ version 2 and of the RCMS. In order to communicate with RCMS a network connection is needed. Furthermore no connectivity is required.

The software has been tested on the following system:

| | |
|-------------------------|---------------------------------------|
| Linux/Intel PC platform | Linux 2.4.21-27.0.4.EL.cernsmp #1 SMP |
| XDAQ | Version 3.7.2 |
| RCMS | Version 2.2.1 |

2.2 Obtaining the Software

The JC software can be obtained from the CVS repository at `TriDAS/daq/jobcontrol` or downloaded from the XDAQ web site at <http://cern.ch/xdaq>.

After downloading install the software under your local TriDAS distribution trees as follows:

```
% cd $XDAQ_ROOT/..  
% tar -zxvf jobcontrol.tgz
```

This will unpack the software in the path `TriDAS/daq/jobcontrol`. The following directories are created:

| Folder | Description |
|--------|-------------|
|--------|-------------|

| | |
|---|--|
| doc, include, interface, lib, src, test | Standard XDAQ application directories |
| Install | Installation script, configuration files |

2.3 Installing JobControl

JobControl is part of the worksuite package of xdaq. For installation type `make Set=worksuite` in the `./TriDAS/daq` directory. This will produce the library `./TriDAS/daq/jobcontrol/lib/linux/x86/libjobcontrol.so`.

Next you will have to modify the configuration file `jc.conf` in `./TriDAS/daq/jobcontrol/install` to set the correct values for `XDAQ_ROOT`, the user, the logfile location and the port under which jobcontrol runs. The default for the port is 39999.

As of Xdaq3.5.2 the xdaq distribution contains a script to start and stop xdaq based services (similar to the originally used `xdajc` script).

To install `jc` as a service do the following:

1. Modify `jc.conf` to fit your environment
2. Copy `jc.conf` to `/etc`
3. Install the daemon script `xdaqd` (located in `TriDAS/xdag/scripts`)

The service can be started and stopped by

Starting:

```
/sbin/service xdaqd jc start
```

Stopping:

```
/sbin/service xdaqd jc stop
```

Re-starting:

```
/sbin/service xdaqd jc restart
```

Status info:

```
/sbin/service xdaqd jc status
```

3 Configuration

The JC application is stateless and has no configurable parameters. Environment variables to control the access to JobControl can be set.

Root access: `JC_ROOT_ALLOWED=YES`

If set to "YES" processes with root privileges can be started with JobControl, if set to "NO"

no processes with root privileges can be started.

User access: JC_USERS_ALLOWED=USER1:USER2:USER3

Specifies a “:” separated list of UNIX users, under which accounts processes can be started.

Other accounts are forbidden.

If not specified, all valid UNIX accounts are allowed.

4 Operation

The JC application uses a minimal set of SOAP commands to start a new XDAQ executive.

The commands and their actions are summarized in the following table.

| Command | Action |
|------------------|---|
| executeCommand() | Start an executable |
| killAll() | Kill all executables started by jobcontrol |
| killExec() | Kill specific executables started by jobcontrol |

Table 1 Summary of Commands and Action of JC

The `Configure()` command is empty. The `executeCommand()` command starts an executable. The format of the input SOAP message is:

```
<executeCommand execPath="" uid="" jid="" user="" argv="" notifierURL="">
  <EnvironmentVariable VAR1="" VAR2="" ..>
</executeCommand>
```

The username precedents the user ID in case both is given. An arbitrary environment with multiple environment variables for the executable can be defined. The process ID is stored internally for later destruction. The returning SOAP message contains a Job ID which can be used for selective kills of executives.

For security reasons JobControl can not start processes as user root unless the environment `JC_ROOT_ALLOWED=YES` is set. The access can be further restricted by setting the `JC_USERS_ALLOWED=USER1:USER2:USER3` environment variable, specifying UNIX user names, separated by “:”.

When the `killAll()` command is called all processes started with the JC application are killed.

A selective killing of processes can be done with the command `killExec()`. The format of the input SOAP message is:

```
<killExec uid="" user="" jid=""/>
```

The attributes `uid`, `user` and `jid` are optional. If only `uid` or `user` is given, all jobs owned by the user are killed. If a `jid` (Job ID) is given, only the executives with that `jid` is killed. The `jid` can be any string.

For notification of crashed jobs a `notifierURL` can be specified, to which job control will attempt to send a sentinel message containing the JID of the crashed job.

The handling of JC is implemented in the Function Manager part of RCMS. The correct starting and stopping of the XDAQ executive is coded inside the State Machine of RCMS.

The user has only to provide the correct parameters for the JC application via the Resource Service of RCMS.

5 *References*

[1] The CMS collaboration, The Trigger and Data Acquisition project, Volume II, Data Acquisition & High-Level Trigger. CERN/LHCC 2002-26, ISBN 92-9083-111-43

[2] XDAQ Companion, "Getting Started", see <http://cern.ch/xdaq>