EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH

ATLAS DAQ – DCS Communication Software

Test Plan

Authors: R. Hart (NIKHEF, Amsterdam)
V. Khomoutnikov (PNPI, St.-Petersburg)
Note Number: 166
Version: 1.0
Date: February 21 2001
Reference: http://atddoc.cern.ch/Atlas/Notes/166/Note166-1.html

1 Introduction

This note presents the **Test-Plan** for the DAQ–DCS Communication (DDC) software of the ATLAS HLT/DAQ/DCS system [1]. The DDC software is intended to interface the DAQ and DCS and to support the needs of run time information exchange. Those needs have been formally specified in the User Requirements Document [2] and were the base for the High Level Design [3]. The URD defined 3 subsystems the DDC shall provide:

1) Bi-directional exchange of data like parameters and status values (DDC-DT);
2) Transmission of DCS messages, like alarms, to DAQ (DDC-MT);
3) Ability for DAQ to issue commands on DCS (DDC-CT).

The subsystems listed above are independent and will be implemented separately. Therefore they can be tested individually. The Online software [4] (formerly known as the Back-end) is the interface point for the DDC on the DAQ side. The DCS is implemented by a SCADA system. The PVSS II system of ETM [5] will be utilized for that purpose and its API will be used as interface point.

Any manipulation with the physics data is beyond the scope of the DDC software.

2 Features to be tested

The following features have to be tested for each subsystem:

1. **Functionality**: each subsystem has to comply with the requirements of the URD. For each subsystem it means:
   - DDC-DT:
     - Setting a variable of the SCADA Database by a DAQ application. Check whether value is taken over in SCADA database.
     - Update a DCS variable of the SCADA Database. Check whether corresponding IS_Server is updated as well.
   - DDC-MT:
     - Subscribe on an alarm of the SCADA system and trigger it. Based on the configuration a message should be generated and send to the MRS. A DAQ application, which performed the proper subscription criteria, should receive it. The message itself has to comply on the MRS message convention. Check if contents of message reflects the alarm.
     - The same as previous, but instead of an alarm a variable.
   - DDC-CT:
     - Issue a command, by a DAQ application, and wait for the result. Check the result.
     - Issue an invalid command and check the result.
     - Check result if command does not reply within certain timeout.

2. **Reliability**: the subsystems shall be tested under different circumstances, like crashing of PC's, network problems, availability of servers (both DAQ and DCS), ability to update log files, etc. In all cases check how subsystem recovers and examine what goes wrong. There are a lot of scenario's possible, but at least the following subjects have to be carried out:
   - Kill the relevant subsystem application and restart it.

- Disconnect the network for 5 minutes, from PC on which subsystem server is running, and re-establish connection.
- Kill one or more PVSS servers.
- Kill IS_Server, MRS_Server or Run_Controller in case of testing DDC_DT, DDC_MT or DDC_CT respectively.

3. **Scalability**: how much DDC subsystems could run in parallel on a single Linux system. It is foreseen that for each DCS partition a separate subsystem has to run.
4. **Performance:** although the data rate between DAQ and DCS is expected to be low (compared to other systems), it is nevertheless important that the response remains between acceptable limits. For each subsystem the average time of a basic action has to be measured. An *endurance* test, running tests in different order on multiple platforms over an extended period (24 hours), has to be part of the performance test.

3 Features not to be tested

The DDC graphical user interface and the configuration database of each subsystem are not part of the test plan.

4 Approach

The tests have to be developed and made part of the DDC component. They will either be simulated or performed in a real setup. It is foreseen that on the DCS side the SCADA database is updated by an emulator, what means that no real hardware is attached. For each subsystem a dedicated configuration setup has to be developed, which has to be loaded inside its configuration database.

5 Pass/Fail, Suspension/Resumption Criteria

For each feature a list of criteria has to be made, together with an associated test. All tests must run from start to completion. For the functionality tests the pass/fail criteria are simple; it fails or does not. There are no suspension/resumption criteria.

6 Test Deliverables

A test report is delivered. It contains a detailed description how the subsystems match the requirements and how they behave under external conditions. Furthermore, a set of tests will be delivered, including the scripts how to start them. The test software and scripts are known as *testware*.

7 Environmental Needs

In order to accomplish the test plan the following hard/software is needed:

1. A Linux PC including PVSS II and Online software. This PC will behave as bridge between DAQ and DCS. The subsystem applications will run on it.
2. Two or more PC’s running WNT, W2000 or Linux on which PVSS II is installed. At least the majority of it has to run WNT or W2000, in order to reflect the expected setup of the ATLAS DCS.
3. A Solaris and a LynxOS system including the OnLine software. These systems run DAQ applications, which act as clients of the DDC subsystem.

8 Responsibilities

The main responsibility of the testware should be someone outside the field of DAQ and DCS. He or she should have the responsibility on both the development of the tests, as well as the actual performing of them. A student, with interest in software engineering, would be an ideal candidate. With support of the authors, she/he could create the testware and publish a test report.

9 Schedule

Although the DDC prototype is not finished yet (it is expected around May 2001), the work on the test plan can already start now. The complete coverage of it may not take more than 3 months.

10 References

[1] ATLAS high level Triggers, DAQ and DCS Technical Proposal, CERN/LHCC/2000-17, http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/SG/TP/draft_tp.html.

[2] R.Hart, V.Khomoutnikov “ATLAS DAQ–DCS Communication Software: User Requirements Document”, Nov. 2000, http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/DDC/ddc_urd.pdf.

[3] R.Hart, V.Khomoutnikov “ATLAS DAQ–DCS Communication Software: High Level Design”, Feb. 2001, http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/DDC/ddc_hld.pdf.

[4] I.Alexandrov et al, “Performance and Scalability of the Back-End in the ATLAS/DAQ EF Prototype –1”, RT99 Santa Fe, http:.//atddoc.cern.ch/Atlas/Conferences/RT99/Rt175.ps.

[5] PVSS II von ETM, http://www.pvss.com