Generation of Test Reports with CGS

CGS Application Note #1

Issue 1, Date: 31. August 2001

J. Rüting – Astrium GmbH, IO63

Abstract:
This short application note for CGS outlines possibilities, how test reports of CGS sessions could be generated. This paper is based on ideas how test reports could look like and describes different variants how these reports could be generated. For a specific project there might exist dedicated requirements on the format of the test reports or the limitations of the solutions outlined in this paper might not be acceptable. For those cases this note might serve as a discussion basis to find the project specific solution.

Not all of the variants described in this paper could be realised with the standard CGS functionality – if specific functions are needed, then this is described in the relevant section.

If a project finds other or modified solutions for test reports, which could be of interest for other CGS users, then a short descriptions of their solution would be highly appreciated for the benefit of other CGS user.

Table of contents

1. Definition of Test Reports

Test reports in this paper mean the more or less automated process to generate a printable report of the major events and test results of an automated test performed with CGS. The typical CGS test scenario involves the execution of several automated procedures and User Libraries (either in sequence, or in parallel, or as sub-sequences). This leads to the execution of many automated test steps. After completion of a test, it will be the task of the responsible test engineer to generate a report of the test execution to conclude on the results of the test and to document the proper execution. This test report could typically contain the major events of the test and specific test results (like selected measurements, results of statistical analysis or limit checks). An example of such a scenario and the correlated test report is sketched in figure 1-1.

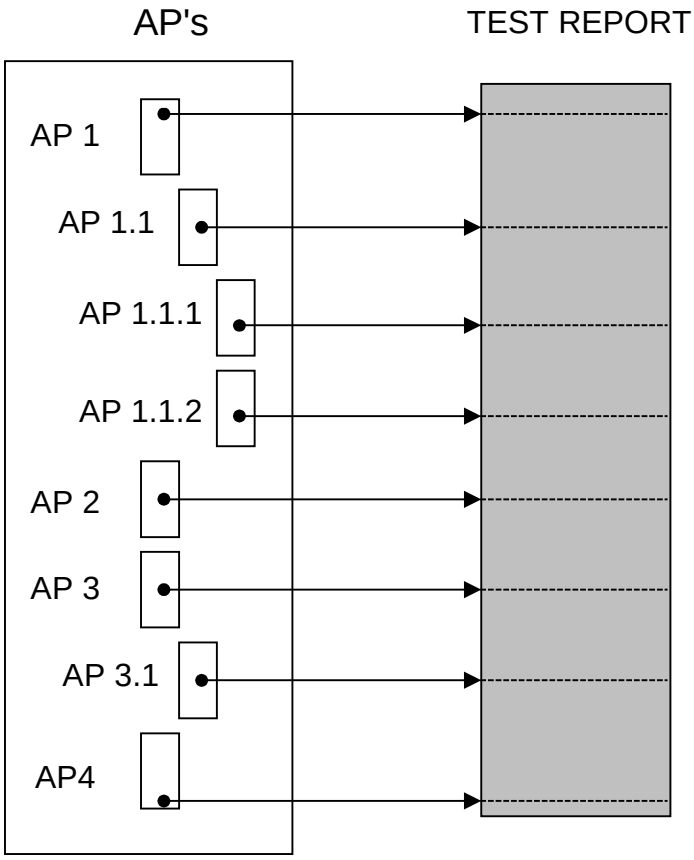

Figure 1-1 : Automated Procedures (AP's) and Test Reporting

A typical test report could contain the elements as listed below:

A. Test Execution Summary Protocol

29 Aug 2001 14:03:59	Telemetry Test for Subsystem xyz started
29 Aug 2001 14:06:23	Activation of EGSE completed – Mode = abc
29 Aug 2001 14:08:13	Standby power for Subsystem xyz activated
29 Aug 2001 14:08:23	Subsystem xyz switched on and started in nominal mode
29 Aug 2001 14:09:12	Results of statistical analysis of telemetry data, see attached table: "Statistical Analysis Telemetry Test for Subsystem XYZ on 29. August 2001"
29 Aug 2001 14:09:13	Telemetry data analysis ok – no exceptions
29 Aug 2001 14:10:15	De-activate subsystem xyz
29 Aug 2001 14:12:01	Standby power off
29 Aug 2001 14:13:00	Telemetry Test Subsystem xyz completed successfully – EGSE in standby

B. Example of a Test Result Sheet

Below an example of a result table is attached, which could be generated on-line during the test execution. This table includes the physical measurements and further processing of the data (like statistical analysis and check if the results are within the given limits). The continuation of the on-line test will depend on the result of this intermediate test data analysis.

Statistical Analysis Telemetry Test for Subsystem XYZ on 29. August 2001

Parameter	Description	Unit	Measured Min/Max	Mean value	Tolerance Min. val. Max. val.	Sigma Phys.	Sigma Tolerance	Check Val/All
F2H003	AC Supply for Pos Det	VOLT	27.8445 28.3555	28.0706	25.960 30.040	0.1805	3	OK OK
F2H005	Supply Voltage BF1	VOLT	28.5324 28.9420	28.7522	26.320 29.680	0.0887	1	OK OK
F2H007	Cur (SM Z) Ser Val	MAMP	0.7206 0.8013	0.7617	-11.000 11.000	0.0405	1	OK OK
F2H008	Cur (SM YN) Ser Val	MAMP	5.7270 5.7270	5.7270	-11.000 11.000	0.0000	1	OK OK
F2H011	Supply Current BF1	AMP	0.3738 0.3849	0.3758	0.036 0.732	0.0043	2	OK OK
H0A004	Environmental Pres	BAR	1.0200 1.0268	1.0268	0.813 1.213	0.0007	1	OK OK
H28001	Turbine Speed NO ME	TR/M	5.8896 5.8896	5.8896	-170.000 170.000	0.0000	1	OK OK

2. Test Reports based on CGS TRDB events

The standard means in CGS to generate automated test reports is using the LOG-Statement in the CGS command language system (UCL = User Control Language). This LOG-Statement can be executed either from any CGS console in the command window or within any automated procedure / User Library. As a result from this command, the string in the LOG-Command is written into the CGS TRDB log event table (together with all the other CGS events).
After completion of the test (or during the test session) the LOG events can be extracted from the TRDB using the CGS TEV (Test Evaluation S/W) into a raw test report file. The final test report could be generated through a post processing S/W (e.g. UNIX AWK script, standard editor means, EXCEL application) to extract the relevant parts of the LOG events into the final test report.
This principle of operation is illustrated in Figure 2-1; an example of a simple test report generated by this means is given in Figure 2-2.

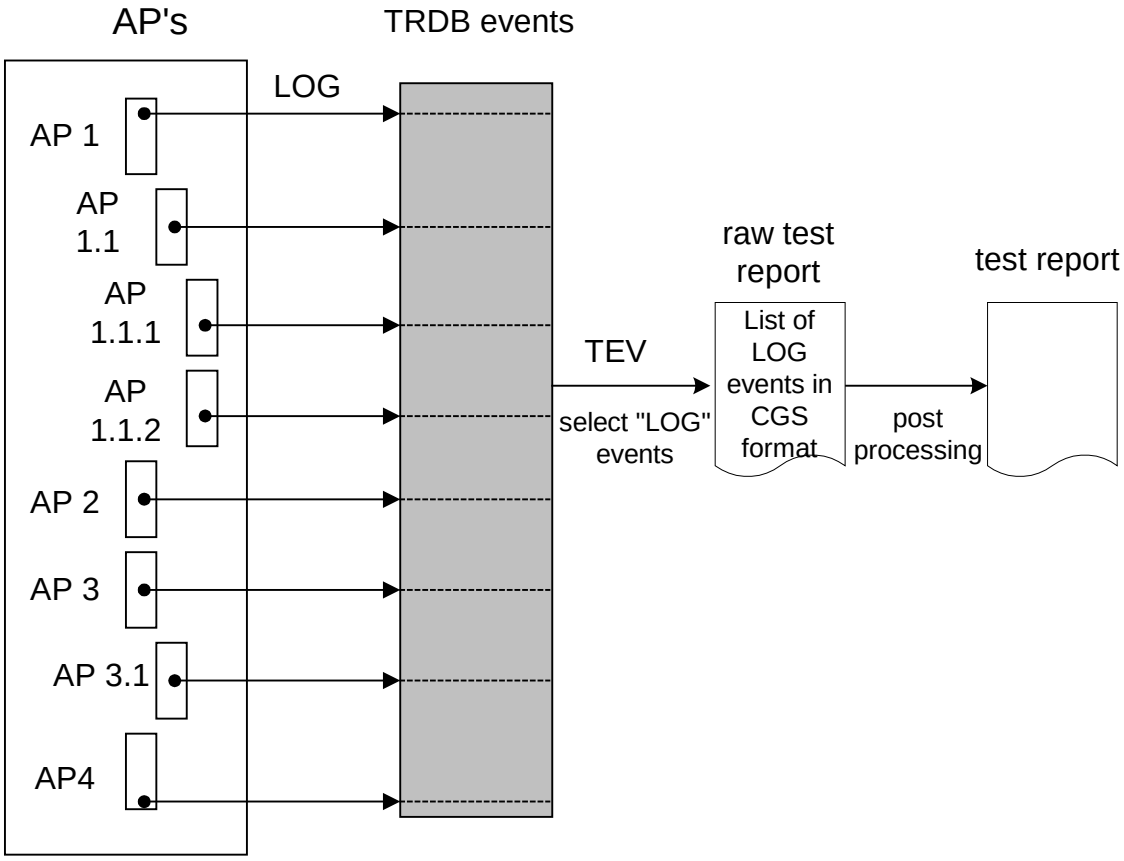

Figure 2-1 Test Reports generated from TRDB Log events

```
29 Aug 2001  11:47:03 Sample_Report_1: ----------------------------------------
29 Aug 2001  11:47:03 Sample_Report_1: Result of Power Test
29 Aug 2001  11:47:03 Sample_Report_1: ----------------------------------------
29 Aug 2001  11:52:15 Sample_Report_1: Power Supply Switched ON
29 Aug 2001  11:52:23 Sample_Report_1: Power Distribution 1.0 Switched ON
29 Aug 2001  11:53:01 Sample_Report_1: Power Distribution 1.1 Switched ON
29 Aug 2001  11:53:01 Sample_Report_1: ------------------------------------------------
29 Aug 2001  11:55:45 Sample_Report_1: Verified: \SAMPLE\PWR\VOLTAGE_1.1 = 27.999 V
29 Aug 2001  11:55:52 Sample_Report_1: Verified: \SAMPLE\PWR\STATUS_1.1 = ON
29 Aug 2001  11:58:17 Sample_Report_1: Verified: \SAMPLE\PWR\VOLTAGE_1.0 = 4.999 V
29 Aug 2001  11:58:59 Sample_Report_1: Verified: \SAMPLE\PWR\STATUS_1.0 = ON
29 Aug 2001  11:58:59 Sample_Report_1: ------------------------------------------------
29 Aug 2001  11:58:59 Sample_Report_1: End of Report
29 Aug 2001  11:58:59 Sample_Report_1: ------------------------------------------------
```

Figure 2-2 Example of a Test Report generated from CGS LOG events

3. Test Reports using the FILE_IO SAS and OPP S/W

Another alternative for test report generation could be the usage of the FILE_IO SAS together with OPP (Online Post Processor) SW. This allows the generation of flexible test reports in a comprehensive and test engineer oriented way.

The concept of the test report generation using the OPP is illustrated in figure 3-1. The automated procedures (APs) provide information related to the test reports to a central AP, the OPP AP, using the command “WRITE_MESSAGE_TO_AP”. The OPP AP generates and maintains the test related report files by use of the File I/O SAS; these files are called “Raw Test Reports” in Figure 3-1. Then the OPP S/W, a self-standing Unix process reads the “Raw Test Reports”, interprets the included formatting information and displays the result of that process either in a dedicated window on a workstation or generates a printable test report. The generated output contains information about the steps executed and formatted test results (e.g. tables of measurements). The master AP, which is in charge of controlling the entire test, generates a dedicated handle for each test and passes the handle to called APs and library procedures. The test report related messages generated by the APs and library procedures include the handle from the master AP so that the OPP AP can allocate the report messages to the individual test reports. Figure 3-2 shows an example of a final formatted test report from the METOP project.

This way of generating visible and printable test reports quasi online is fully under user control (control of the AP development). The test reports are stored in files and are therefore not under configuration management of the MDB. The test reports do not include information from the TRDB, like CGS warning and error messages.

The OPP SW is a CGS application which is not part of the basic SW system. The FILE_IO SAS is part of the CGS S/W system.

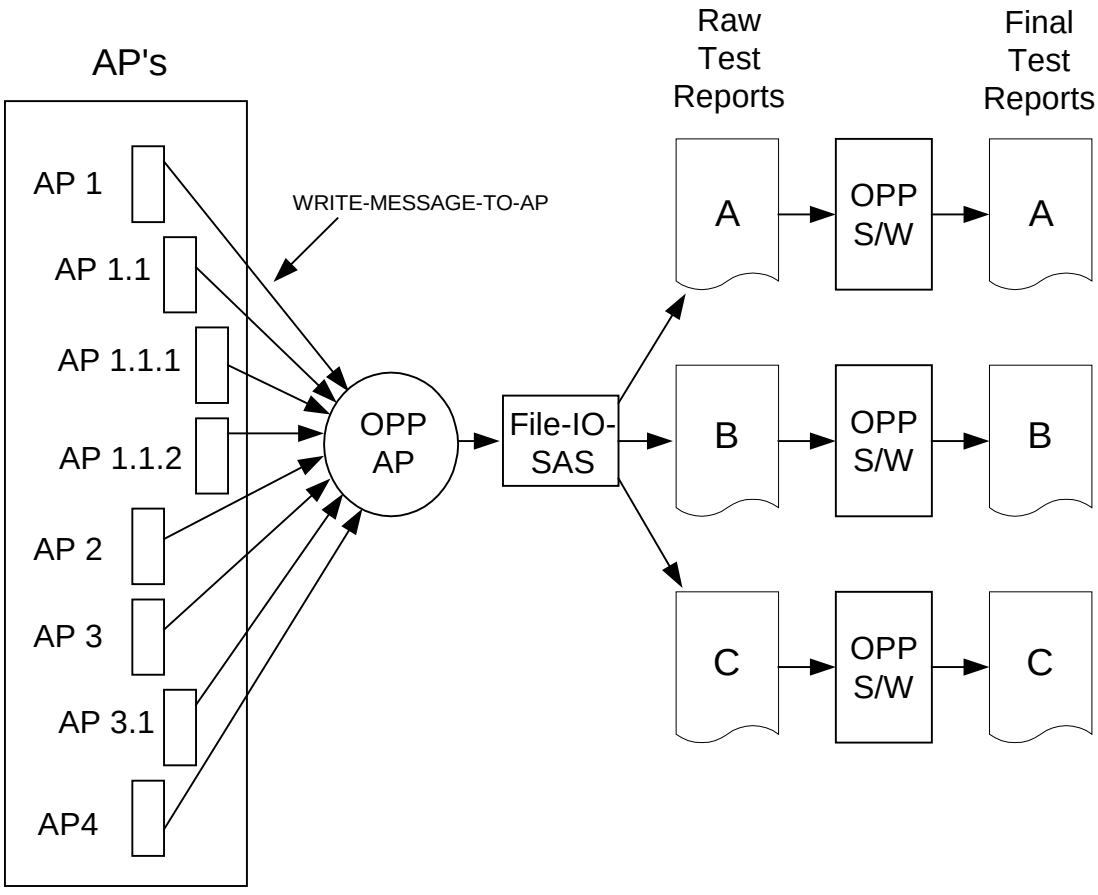

Figure 3-1: Test Reports using the FILE_IO SAS

```
<**OPP**>: AP <OAA_GOMTV_ADD1> with ID 537 registered! (12.6.2001 12:23:30.558)
<OAA_GOMTV_ADD1>:
<OAA_GOMTV_ADD1>:
<OAA_GOMTV_ADD1>: *************************************************************************************************************************
<OAA_GOMTV_ADD1>: *************************************************************************************************************************
<OAA_GOMTV_ADD1>: ****                                                                                                                 ****
<OAA_GOMTV_ADD1>: ****                                                                                                                 ****
<OAA_GOMTV_ADD1>:                                       Loop for Check of Critical Monitoring Limits
<OAA_GOMTV_ADD1>:                                       ============================================
<OAA_GOMTV_ADD1>: ****                                                                                                                 ****
<OAA_GOMTV_ADD1>: ****                                                                                                                 ****
<OAA_GOMTV_ADD1>: *************************************************************************************************************************
<OAA_GOMTV_ADD1>: *************************************************************************************************************************
<OAA_GOMTV_ADD1>:
<OAA_GOMTV_ADD1>:
<OAA_GOMTV_ADD1>: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
<OAA_GOMTV_ADD1>: ****** Begin of AP  OAA_GOMTV_ADD1  at 12:23:33 ******
<OAA_GOMTV_ADD1>: vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
<OAA_GOMTV_ADD1>:
<OAA_GOMTV_ADD1>:  -====
<OAA_GOMTV_ADD1>:  -====   Step 1 Display and verify the GOME TM
<OAA_GOMTV_ADD1>:  -====
<OAA_GOMTV_ADD1>: -+----------+--------------------------------------+------------------+------+-----------+-----------+-----------+-------
<OAA_GOMTV_ADD1>:  I     Time I                      Parameter Desc. I         End Item I Unit I    actual I       expected        I Result
<OAA_GOMTV_ADD1>: -+----------+--------------------------------------+------------------+------+-----------+-----------+-----------+-------
<OAA_GOMTV_ADD1>:  I 12:23:34 I            FPA 1 Temperature (first) I          ONA0014 I    K I  2.52E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:35 I            FPA 2 Temperature (first) I          ONA0015 I    K I  2.53E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:35 I            FPA 3 Temperature (first) I          ONA0016 I    K I  2.52E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:35 I            FPA 4 Temperature (first) I          ONA0017 I    K I  2.53E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:35 I            FPA 1 array Temperature-2 I          ONA0051 I      I  2.52E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:35 I            FPA 2 array Temperature-2 I          ONA0052 I      I  2.53E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:36 I            FPA 3 array Temperature-2 I          ONA0053 I      I  2.53E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:36 I            FPA 4 array Temperature-2 I          ONA0054 I      I  2.53E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:36 I                     PMD-s array temp I          ONA0027 I      I  2.80E+02 I  2.53E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:23:36 I                     PMD-s array temp I          ONA0120 I      I  2.80E+02 I  2.53E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>: -+----------+--------------------------------------+------------------+------+-----------+-----------+-----------+-------
<OAA_GOMTV_ADD1>: Abort of this AP only via HCI command possible !!!!!!!!
<OAA_GOMTV_ADD1>: -+----------+--------------------------------------+------------------+------+-----------+-----------+-----------+-------
<OAA_GOMTV_ADD1>:  I     Time I                      Parameter Desc. I         End Item I Unit I    actual I       expected        I Result
<OAA_GOMTV_ADD1>: -+----------+--------------------------------------+------------------+------+-----------+-----------+-----------+-------
<OAA_GOMTV_ADD1>:  I 12:28:37 I            FPA 1 Temperature (first) I          ONA0014 I    K I  2.52E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:37 I            FPA 2 Temperature (first) I          ONA0015 I    K I  2.52E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:37 I            FPA 3 Temperature (first) I          ONA0016 I    K I  2.52E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:37 I            FPA 4 Temperature (first) I          ONA0017 I    K I  2.53E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:37 I            FPA 1 array Temperature-2 I          ONA0051 I      I  2.52E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:37 I            FPA 2 array Temperature-2 I          ONA0052 I      I  2.52E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:37 I            FPA 3 array Temperature-2 I          ONA0053 I      I  2.53E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:38 I            FPA 4 array Temperature-2 I          ONA0054 I      I  2.53E+02 I  2.40E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:38 I                     PMD-s array temp I          ONA0027 I      I  2.80E+02 I  2.53E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>:  I 12:28:38 I                     PMD-s array temp I          ONA0120 I      I  2.79E+02 I  2.53E+02 -  3.23E+02 I
<OAA_GOMTV_ADD1>: -+----------+--------------------------------------+------------------+------+-----------+-----------+-----------+-------
<OAA_GOMTV_ADD1>: Abort of this AP only via HCI command possible !!!!!!!!
```

Figure 3-2 Example Test Report using FILE_IO SAS and OPP S/W (from the METOP Project)

4. Future Test Reports using CGS Extensions

In the following two alternative solutions for generating test reports are briefly discussed. These possible concepts have been discussed various times, but not yet been implemented in CGS. If a project has specific needs on test reports, then the ideas presented here could be of interest in formulating the needs for CGS extensions.

4.1 Test Reports using advanced FILE_IO features

This solution would be similar to the solution with the FILE_IO SAS (described in section 3), but instead of using a separate ADA process (the FILE_IO SAS), the output file could be written directly from the AP or User Library through specific UCL statements. Depending on the formatting possible by these new UCL commands, the output files could represent immediately the test reports.
This simplified approach compared with the solution using the FILE_IO SAS could be taken into account, when the migration of the CGS S/W system to ADA95 has been completed (Version 5 upwards). In the old implementation with ADA83 the impact on the run-time environment was not acceptable.

As mentioned before, this solution is not available yet, but ideas for the implementation exist. On request on a specific project, this CGS extension could be implemented.

4.2 Test Reports using data in the Test Result Database (TRDB)

The most flexible solution for test reports could be the following concept: Data to be written into a test report would be written into a dedicated area in the Test Result Database (TRDB). The area could be called “User Result Table”. The CGS Language System (UCL) would be extended by commands related to read and write data in these tables.
New commands to be implemented would be:

- Open Table in “User Result Table”
- Write data into “User Result Table”
- Append Data to “User Result Table”
- Read data from “User Result Table”
- Close Table in “User Result Table”

This concept allows to store any kind of information related to test reports in the database and also allows processing of the data from AP’s (e.g. calculation of mean values, maintenance of counter for valves).
The test report itself could be generated from the “User Result Tables” by standard database tools like SQL_REPORT or advanced techniques like XML formatted reports.

As mentioned before, this solution is not available yet, but ideas for the implementation exist. On request on a specific project, this CGS extension could be implemented.

5. Abbreviations

AP	Automated Procedure
CGS	Columbus Ground Software
OPP	Online Post Processing
SAS	Special Application Software
TEV	Test Evaluation
TRDB	Test Result Data Base
UCL	User Control Language