

# *dBm069 a new TR-069 testing concept*

## 1 GENERAL

- dBm069 is a tool that validates the TR-069 protocol behaviour of a CPE (a router or a LAN device) in a fast, simple, and reliable way. It includes auxiliary tools to provide a full testing scenario.
- dBm069 is a software tool, no proprietary hardware is needed. It runs on a PC with Windows OS.
- User Friendly Tool. The tool helps tester during all testing process.
- 7 to 1 testing time reduction.
- Allows both automatic or manual tests execution.
- Automatic CPE configuration during automatic testing by means of scripts.
- Allows automatic functionality verifications.
- Includes Pass/Fail module that automates the decision of results validation without the need of checking ACS logs manually.
- Execution logs always available to help debuggers to understand CPE behavior.
- Automatic WireShark captures to ease debugging.
- Node-locked license based on an encrypted "key" locked to a specific dongle. One simultaneous run per license.

## 2 TOOLS INCLUDED

### 2.1 dBm-ACS

- Proprietary ACS. Developed following BBF standards to perform TR-069 tests against a CPE. Able to connect one CPE at a time for testing purposes.
- Allows CPE registration in dBm-ACS even if TR-069 authentication protocol is not correctly implemented in the CPE, detecting and reporting those faults but allowing CPE testing.
- Automatic registration of new CPEs at dBm-ACS by means of detecting an Inform message.

### 2.2 Connection to a comercial ACS

- dBm069-ACS-MOT allows to connect dBm069-Engine to NOKIA ACS (Motive) and run all tests in the same way than against dBm-ACS. Test Cases are the same (no need to be adapted when running them against a different ACS).
- Ready to automatically test a CPE against any ACS brand (under special request).

### 2.3 Servers

- dBm-STUN server.
- HTTP and FTP servers to upload and download files.

### 2.4 Traffic Generator

- Traffic generator/analyzer allows testing related functionalities.

### 2.5 Other Tools

- "Virtual switch" opens and closes PC Ethernet ports simulating connection and disconnection of the CPE under test to avoid manual intervention during automatic test.
- "Comparative Analysis" allows comparing the results of two or more Test Plans.
- "Connectivity Wizard" helps users to validate all scenario connections saving time and helping to locate errors.
- Wireshark captures embedded.
- Test Report generator.

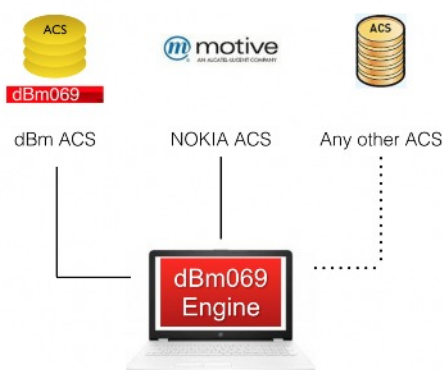


Figure 1: dBm069 can work with its own ACS or with any other ACS brand.



### 3 SETTING THE TESTING SCENARIO

- dBm069 allows testing in different scenarios.
- Connectivity Wizard to help users to solve connectivity problems. Includes instructions to solve connectivity problems:
  - ▶ Manually when configuring the testing scenario.
  - ▶ Automatically when starting test execution.

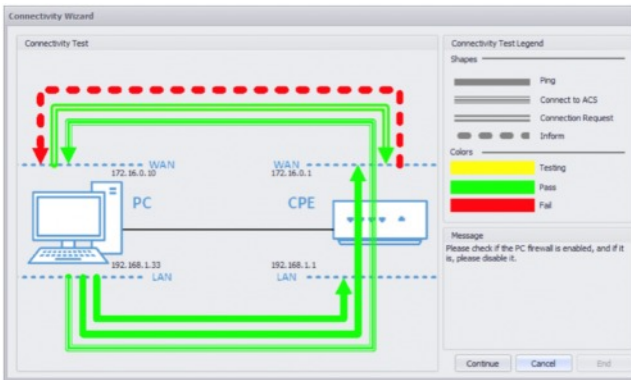


Figure 2: Connection Wizard to help user scenario configuration

### 4 USERS

- Two types of users:
  - ▶ Administrator: Full access and visibility.
  - ▶ Tester: Limited access to information. Can only access their own results or those set as public.
- Allows non-simultaneous multi user. Different users can use the same license using their own passwords. Test execution and results can only be accessed via user password.
- Allows creating, editing, deleting or modifying users.

### 5 MANUAL TESTS

- Manual tests can be performed to:
  - ▶ Check some incorrect behaviour detected during automatic tests and get more information for debugging.
  - ▶ Check the connection between CPE and ACS.
- Manual tests are driven by a friendly and easy to use interface with a powerful “Click-and-Add Methods” selection GUI, with real-time visibility of test execution.

- Several RPC methods can be executed within the same session.
- Parameters to be used in RPC methods can be selected from the CPE data model. Vendor-specific parameters can be shown or hidden from CPE parameters tree.
- “Capture Traces” button records the activity between the CPE and ACS in .pcap format for further Wireshark analysis in manual test mode.
- Results of manual test performed with dBm069 interface are shown at the ACS tab. Logs are shown sequentially as they are displayed by the ACS in a structured way and displayed in different colors depending on whether it is the ACS or the CPE who has sent the message. A second window shows only the name of the RPC methods and their responses, specifying if they have succeeded or failed. By clicking on any method at this window, dBm069 points out the corresponding part of the log at the first window. This allows user to:

- ▶ see at a glance all the RPC methods that failed.
- ▶ easily locate the detail of the ACS log where a RPC method fails.

### 6 AUTOMATIC TESTS

- Allows automatic execution of test cases.
- 24/7 unattended automatic execution on fully automated tests. No need of user interaction in 95% of cases.

#### 6.1 Automatic Test Set Up

##### 6.1.1 CPEs

*CPEs are the devices to be tested with dBm069. A CPE is identified by its serial number.*

- CPEs can be routers (modems, cablemodems, HGUs...) or LAN devices.
- CPEs to be tested automatically need to be created in dBm069 so their testing results can be stored.
- Test Plans and test results are linked to the existing CPEs.
- CPEs can be deleted or modified.

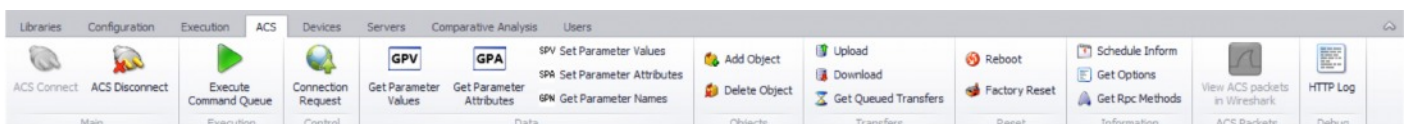


Figure 3: “Click-and-Add Methods” selection GUI Tool bar for easy manual test

6.1.2 Libraries

*Libraries are XML files that contain definition of a list of test cases.*

*A Test Case is a set of RPC methods, actions, and conditions that are executed automatically to check a particular requirement of a CPE.*

*An Action is a proprietary command that dBm069 uses to allow performing automatically some specific tasks such as detecting an Inform with a particular EventCode, opening the http server, etc.*

- Different libraries corresponding to different data models are available.
- Users can create their own libraries or modify the existing ones. Test Cases can be created from scratch or by copy-paste of existing ones. Tree View of libraries allow easy modification of Test Cases using an external XML editor.
- dBm069 works with encrypted libraries. Libraries are encrypted automatically when .xml files are imported.
- Only administrators with dBm069-EDIT module can import .xml libraries and encrypt libraries.
- Administrators with any kind of license can import encrypted libraries that can share with dBm069 licenses that do not have dBm069-EDIT module.
- Libraries can be imported and also deleted if they become obsolete.

6.1.3 Scripts

- Allows to create, edit, delete or modify scripts.  
*Scripts are sets of instructions based on CPE's commands that allow dBm069 to communicate directly with the CPE via telnet, ssh or RS-232. This is useful for example to automatically restore the Configuration file after a FactoryReset.*
- Powerful and friendly Script Editor that helps users creating them.

6.1.4 Test Plans

- A Test Plan is an association of a CPE, a scenario and a Library. A Test Plan contains the set of test cases to be executed automatically to the CPE for which the Test Plan has been created.
- Easy creation of Test Plans.
- The Test Plan designer can define the Test Plan visibility. A Test Plan can be private (only visible for the designer and administrators) or public (visible for all users).
- Possibility of editing the Test Plan properties: its name, visibility, and comments associated to it.

6.2 EXECUTION OF AUTOMATIC TESTS

- Test Plans can be totally or partially executed.
- Intuitive diagnostic display: User can see if a test case has passed or failed both at dBm069 Execution tab and in test reports.
- Allows to pause, re-schedule, abort, and delete Test Cases execution.
- Traffic visualization of all sent and received ACS's and CPE's packets during automatic execution at the Execution tab.

```

13:10:02.318 [CPE <----- ACS] ConnectionRequest
13:10:03.333 Session Start
13:10:03.333 [CPE -----> ACS] Inform [6]
13:10:03.364 [CPE <----- ACS] InformResponse
13:10:03.427 [CPE -----> ACS] End of CPE requests.
13:10:03.458 [CPE <----- ACS] GetParameterValues
13:10:21.928 [CPE -----> ACS] GetParameterValuesResponse
13:10:22.006 [CPE <----- ACS] GetParameterAttributes
13:10:23.522 [CPE -----> ACS] GetParameterAttributesResponse
13:10:24.694 [CPE <----- ACS] GetParameterNames
13:10:25.850 [CPE -----> ACS] GetParameterNamesResponse
13:10:25.959 [CPE <----- ACS] End of ACS requests.
13:10:25.959 Session End
13:11:04.743 [CPE <----- ACS] ConnectionRequest
13:11:07.508 Session Start
13:11:07.508 [CPE -----> ACS] Inform [6]
13:11:07.540 [CPE <----- ACS] InformResponse
13:11:07.618 [CPE -----> ACS] End of CPE requests.
    
```

Figure 4: Detail of the "Execution Window". It shows a simplified list of the transaction between the ACS and CPE.

Table 1: Test Plan Information at the Configuration tab

<b>Program Variables</b>	List of program variables according to the testing scenario and their values (IPs, host name, current date, etc.)
<b>Library Variables</b>	List of library variables defined specifically in the library associated to the Test Plan. Allows library variables edition
<b>Scripts</b>	List of scripts that will be executed during the Test Plan Execution (if there are test cases that use them)
<b>ACS</b>	ACS and CPE data and status (first connection time, if it active or not, SN, CR URL, etc)
<b>Test Plan Data</b>	Test Plan general information: name, owner, CPE type and SN, library to which it is associated, etc

- "Test Steps" window shows test cases syntax in Tree View allowing the user to see the test case definition.

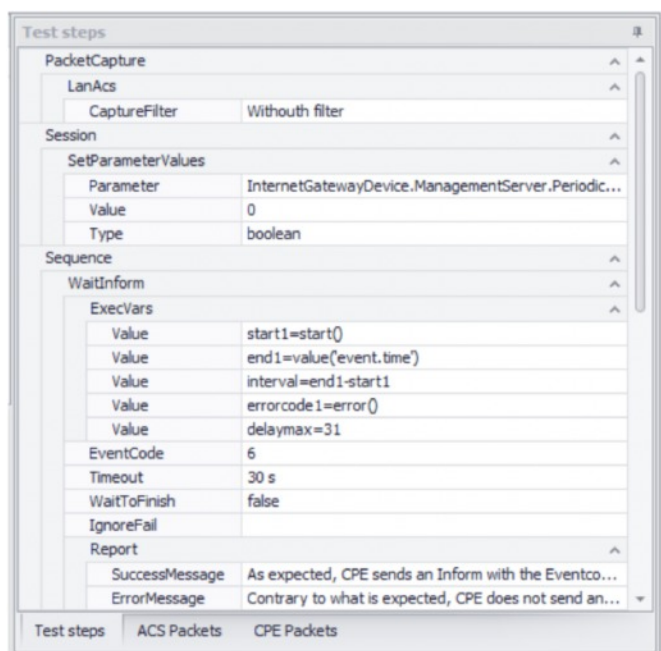


Figure 5: "Test Steps" window shows the Test Case syntax and allows tester to see all the details of the test to be performed.

- Full visibility of functionality check logs (Actions logs) at the Execution tab.
- WireShark Synchronized connection that allows automatic real time capture of CPE-ACS transactions.
- Setting of Wireshark filters before test execution: only LAN side, only CPE side or both. Other type of filters can be used to select the information to be captured.
- Use of variables to allow full automatic execution of Test Cases. Variables allow saving data read from CPE, using data related to the scenario configuration or data that can be different for any CPE, e.g. FW version, WAN IP, etc. :
- dBm069 uses 3 different kind of variables:
  - ▶ Program variables: Non-editable variables related to the testing scenario.
  - ▶ Library variables: Variables that need / can be edited before starting execution to avoid the need of human intervention. Possibility of importing library variables from previous Test Plans.
  - ▶ Execution variables: used to keep temporary values obtained during test execution to be used at another part of the test case.

### 6.2.1 Resources used to avoid human intervention

- Able to find out the indexes of an existing multi-instance object to execute the correct instance of the objects during automatic execution.
- Allows defining "Preconditions" in Test Cases, i.e. a list of the necessary conditions that must be accomplished in order to allow proceeding to the execution of a test case. This allows to avoid execution of test cases that do not apply to the testing scenario, such as test cases that test configurations / connectors not supported/existing in the CPE.
- Use of control instructions (If, For, Do-while, While-do) allow the execution of different blocks of a test case under different circumstances, allowing executing only the necessary steps in each case.
- Allows the execution of different blocks of a test case depending on whether the CPE is FTTH or XDSL. Note that configuration and some parameter names are different for both kinds of CPEs.
- Functionality tests can be designed to test CPE's configuration and real performance after TR-069 configuration. Available Libraries contain tests cases such as:
  - ▶ WAN browsing: surfing on internet using an specific WAN.
  - ▶ WiFi functionalities: checking available SSID, connecting to a specific WIFI network or surfing on internet using a WIFI network.
  - ▶ LANs accessing the CPE web-GUI and selecting a specific LAN interface.
  - ▶ Port mapping using the embedded traffic generator/analyzer. Different protocols and ports can be checked.
  - ▶ Downloading and uploading files and measuring their real size.
  - ▶ USB functionality: checking the access to a USB stick through the CPE LAN.
  - ▶ PC's Network connection: allows checks on the PC where dBm069 is installed similar to the console IPconfig, IPconfig renew and IPconfigrelease commands.
- Includes Pass/Fail module that automates the decision of results validation without the need of checking ACS logs manually.

### 6.2.2 Automatic execution results in real time

- Real-time visualization of automatic test execution results in a structured way and showing the results of every test step.

- Exchange of messages between CPE and ACS are shown in real time during Test Case execution allowing the Tester to identify at a glance:
  - ▶ Messages sent from the ACS or the CPE.
  - ▶ RPC methods and their responses.
  - ▶ Beginning and end of each TR-069 session.
- dBm069 shows different kind of logs:
  - ▶ ACS logs. Available at the ACS tab; the Execution tab (when testing automatically); and at the HTML test reports.
  - ▶ Actions logs. Available at the Execution tab (when testing automatically).
  - ▶ HTTP logs show communication between the ACS and the CPE at HTTP protocol level. Available at the ACS tab. They are useful when the connection between ACS and CPE has been established correctly but ACS logs are not shown, as it happens when there are authentication problems.
  - ▶ Script logs. If a test case contains a script execution i.e. direct communication of dBm069 with the CPE through CPE's commands.
- Visualization of all sent and received messages at the FTP/HTTP servers, as well as the progress of files upload/download. Available at the Servers tab.
- Visualization of all sent and received messages by the traffic generator, as well as the progress of the exchanged traffic through a port or a range of ports. Available at the Servers tab.
- Visualization of all sent and received messages by dBm-STUN server when testing a LAN device against dBm-ACS.

6.2.3 **End of automatic execution**

- Test Plans are automatically saved after execution and can also be saved although execution is not still finished.
- Test Plans and results are stored in a database to allow test cases re-execution.
- "Reduce Test Plan" button allows to reduce the Test Plan size deleting the Wireshark captures (all of them or only those associated to test cases that have passed).

6.3 **Analysis of automatic execution results**

- Tester can check Test Cases execution results:
  - ▶ From the Execution tab: info displayed in a structured way allowing to easily see the part of the ACS log corresponding to each part of a test case. By selecting a specific test case step, user can see detailed logs of the selected part.
  - ▶ From the ACS tab: info is shown sequentially as displayed by the ACS. Available during execution. Available after execution only if dBm069 has not been closed after finishing the Test Plan execution.
  - ▶ Generating an HTML test report. Available only after execution.
- Displays specific error messages both at the Execution Tab and in HTML reports to help the tester to identify the exact origin of the issue.
- Easy identification of the Informs sent by the CPE to the ACS and the reason of sending them by means of showing their EventCodes. Also in Grid format.
- "Execution Variables" window shows the values of variables used in the test case that is selected.
- "Conditions" window shows the list of conditions defined in the selected test to pass and the values compared for the result evaluation.
- If a test case contains several steps, the test case will only pass if all its steps have passed.

7 **IMPORT/EXPORT TEST PLANS**

- Executed Test Plans can be exported/imported allowing to open them in different PCs from where they were executed in order to share them with other users in other locations.
- When exporting/importing a Test Plan the user may discard WireShark captures associated to it if a smaller size file is wished.
- When importing a Test Plan its visibility can be defined, i.e. it can be imported as Private or Public defining in this way which users in that PC will have access to it.

Table 2: Different kind of Logs available at dBm069

<b>Script Log</b>	Available at Execution tab. Shows logs of scripts of those automatic test cases that includes them (set of commands to communicate directly with the CPE via telnet, SSH or RS232)
<b>Traffic Log</b>	Available at Servers tab. Shows Traffic logs related to all traffic sent/received by the http/ftp servers and the traffic generator/analyzer during Test Plan execution
<b>dBm STUN Log</b>	Available at Serves tab. Shows Traffic logs related to the traffic sent/received by the STUN server during Test Plan execution
<b>ACS Log</b>	dBm069 shows ACS logs corresponding to the whole automatic Test Plan (at the ACS tab) or to each test case (at the Execution tab). Logs are shown in a friendly and structured way
<b>Action Log</b>	dBm069 shows logs corresponding to the execution of Actions (proprietary commands that allow functionality checks).
<b>HTTP Log</b>	Available at ACS tab. Shows ACS sent/received HTTP messages during Test Plan execution

8 **TEST REPORTS GENERATION**

- Allows generation of HTML Test Reports of Test Plans executed automatically.
- Information in Test Reports is organized in different blocks:
  - ▶ Execution Data: Contains the information related to the Testing Scenario which was used to execute the Test Plan.
  - ▶ CPE Identification Data: Contains all the information related to the CPE tested.
  - ▶ General Defects: List of general defects detected in the Test Plan.
  - ▶ Test Plan Summary: List of test cases with their results and error messages.
  - ▶ Methods: Includes description and summary of test steps performed in each Test Case (all RPC methods and actions sent).
  - ▶ Logs: Detailed ACS logs of each Test Case.
- *A General Defect is a defect related to the wrong implementation of the protocol in the authentication, the format of the SPVResponse, a problem related to the Inform time, etc. General defects are detected and reported but dBm069 can still go on with the execution of the Test Plan.*
- HTML reports include hyperlinks to easily navigate through the Test Report both down to the desired part of the log; and up from the log to the list of Test Cases.
- Testers can generate different kind of Test Reports.
  - ▶ Test Report of the whole Test Plan.
  - ▶ Individual Test Reports of the whole Test Plan.
  - ▶ Test Report including only a selection of Test Cases.

- Test reports include ACS logs in a structured way to easy the analysis of Test Cases results. Starting and end of RPC methods exchanged, their responses, and actions are clearly identified in HTML test reports.

9 **WIRESHARK CONNECTION**

- WireShark analysis available of all sent and received ACS's and CPE's packets of every test case.
- Wireshark analysis available at the ACS tab of all sent and received ACS's packets since dBm069 last connection with the ACS. Available for both manual and automatic test cases.
- Possibility of reduce the Test Plan size by eliminating either all WireShark captures associated to the Test Plan, or only those associated to Test Cases that have passed.

10 **COMPARATIVE ANALYSIS TOOL**

- Comparative Analysis tool allows to compare different Test Plans results by creation of **Comparatives** (group of Test Plans to be compared) allowing users to easily compare if a test case result has improved or worsened in the different Test Plans performed.
- Some data can be changed in Comparatives: Comparative's name; visibility (private/public); test plans order; and test plans can also be added or deleted.
- A comparative shows:
  - ▶ All general data of the Test Plans included in the comparative: FW version, model, etc.
  - ▶ Test Plans results, showing each test case result.
  - ▶ Comparative conclusions for each Test Plan.

Table 3: List of dBm069 windows that provide real time information about automatic execution at the Execution tab

<b>Test Plan</b>	List of test cases to run in a CPE. Full Test Plan and status of Test Cases
<b>Test Log</b>	Full test logs captured in a Test Case execution. Includes RPC methods and Actions
<b>Conditions</b>	List of conditions that the CPE must pass to mark an executed Test Case as PASS specifying the values obtained
<b>Execution Variables</b>	List of execution variables of the Test Case and their values
<b>Test Report Variables</b>	List of the Test Report variables of a Test Case and their values
<b>Script Log</b>	Script logs corresponding to all Test Cases executed in a Test Plan
<b>Text/Grid</b>	ACS logs in text/grid view corresponding to the RPC method/Action selected at the Test Log window
<b>Test Steps</b>	Test steps as they are defined in the .xml library
<b>ACS Packets</b>	Sent and received messages by the ACS related to a Test Case (not only SOAP)
<b>CPE Packets</b>	Sent and received messages by the CPE related to a Test Case (not only SOAP)
<b>Execution Details</b>	Real time view of all RPC methods and actions executed in a Test Case, their results, and specific messages about their execution
<b>Test description</b>	Test Case description, aim of the test
<b>Error Message</b>	Detailed error messages associated to a Test Case in case it has failed
<b>Notes</b>	Field for the tester to write any comment/note associated to the Test Case
<b>Events</b>	General information about the Test Plan execution in real time: errors, warnings, and other messages associated to it

- Comparative conclusions show the following statistics:
  - ▶ Total number of Test Cases.
  - ▶ Number of passed Tests Cases.
  - ▶ Number of failed Test Cases.
  - ▶ Number of Non-executed or Aborted Test Cases.
  - ▶ Number of tests with Improved Results vs. previous result.
  - ▶ Number of tests with Worsened Results vs. previous result.
  - ▶ Number of tests with Same Results vs. previous result.
  - ▶ Number of test cases excluded from the Comparative.
- Comparative Analysis tool shows detailed information of every test case that fails:
  - ▶ Logs of the test case (communication messages sent and received between the ACS and the CPE).
  - ▶ Description and Fail message of the test case.
- Comparative results can be displayed in an integrated spreadsheet, where the user can add notes, change colors, etc. From there they can be saved in XSLX, XSL, CSV or text format.
- Allows sharing Comparatives with other dBm069 users in other locations by exporting/importing with a different dBm069 license. Comparatives can also be exported to .csv format to be opened with any spreadsheet editor.
- Allows to generate Comparative reports in .html format.

11 **STANDARDS AND TEST PLANS**

- Supports all TR-069 data models (TR-181, TR-111, TR-098, TR-106, TR-104, TR-157, TR-135, TR-140, TR-142, TR-143, TR-196)
  - Allows working with different libraries (different data models).
- Supports Conformance, Performance and Functionality oriented Test Cases.
- Creation of new TEST CASES including Pass/Fail criteria without requiring the user to be a SW programmer.
- Allows defining test cases according to Vendor specific requirements and parameters.

12 **AVAILABLE LIBRARIES**

- Predefined TR-098 library: includes more than 130 test cases including functional checks of configurations performed.
- Predefined TR-098 Parameters Check Library: includes more than 1100 test cases designed to check the correct implementation of all TR-098 parameters.
- Predefined TR-104 library: Customizable to customer requirements. Customer must provide the gateway data model since it involves vendor-specific parameters.
- Predefined TR-111 library: Includes Test Cases both on the Gateway and LAN device to check the correct implementation of TR-111
- Predefined TR-135 library: Customizable to customer requirements. Customer must provide the STB data model.
- Predefined TR098 IPV6 Library: Customizable to customer requirements. Customer must provide the gateway data model, since it involves vendor-specific parameters.

114	GTW.TR069.098.f.wan.5	WAN - Different configurations using DHCP for the dynamic WAN	✓ Pass	✓ Pass	✓ Pass	✓ Pass
115	GTW.TR069.098.f.wan.6	WAN - Incorrect values	✓ Pass	✓ Pass	✓ Pass	✓ Pass
116	GTW.TR069.098.f.wan.7	WAN - WAN configuration (Direct Connection)	✓ Pass	✓ Pass	✓ Pass	✓ Pass
117	GTW.TR069.098.f.wless.1	Wireless - WLAN disabled	✓ Pass	✓ Pass	✓ Pass	✓ Pass
118	GTW.TR069.098.f.wless.2	Wireless - WLAN without encryption	✓ Pass	✓ Pass	✓ Pass	✓ Pass
119	GTW.TR069.098.f.wless.3	Wireless - WLAN with WEP encryption	✓ Pass	✓ Pass	✓ Pass	✓ Pass
120	GTW.TR069.098.f.wless.4	Wireless - WLAN with WPA encryption	✗ Fail	✗ Fail	✗ Fail	✗ Fail
121	GTW.TR069.098.f.wless.5	Wireless - WLAN with MAC filter	✗ Fail	✗ Fail	✗ Fail	✗ Fail
122	GTW.TR069.098.f.wless.6	Wireless - Different SSID configurations	✓ Pass	✗ Fail	✗ Fail	✗ Fail
123	GTW.TR069.098.f.wless.7	Wireless - Different WPS configurations	✗ Fail	✗ Fail	✗ Fail	✗ Fail
124	GTW.TR069.098.f.wless.8	Wireless - Incorrect values	✗ Fail	✗ Fail	✗ Fail	✗ Fail
125	GTW.TR069.098.f.wless.9	Wireless - Different channel values	✗ Fail	✗ Fail	✗ Fail	✗ Fail
126	GTW.TR069.cp.spv.5	SPV - Check ParameterKey parameter	✗ Fail	✗ Fail	✗ Fail	✗ Fail
127	GTW.TR069.cp.spa.2	SPA - Operation	✗ Fail	✗ Fail	✗ Fail	✗ Fail
128	GTW.TR069.cp.ao.2	AO - Check ParameterKey parameter	✗ Fail	✗ Fail	✗ Fail	✗ Fail
129	GTW.TR069.cp.do.2	DO - Check ParameterKey parameter	✗ Fail	✗ Fail	✗ Fail	✗ Fail
130	GTW.TR069.cp.rbt.1	Reboot - Operation	✓ Pass	✓ Pass	✓ Pass	✓ Pass
131	GTW.TR069.cp.fr.1	FactoryReset - Operation	✓ Pass	✓ Pass	✓ Pass	✓ Pass
132	GTW.TR069.098.f.id.1	Identification and Registration - Inform and Inform Response	✓ Pass	✓ Pass	✓ Pass	✓ Pass
133	GTW.TR069.098.f.id.2	Identification and Registration - Parameters needed at the registration process	✓ Pass	✓ Pass	✓ Pass	✓ Pass
Passed Tests			83	79	83	82
Failed Tests			36	40	37	38
Non-executed or Aborted Tests			14	14	13	13
Total Tests			133	133	133	133
Improved Results vs. previous column				1	4	0
Worsened Results vs. previous column				4	1	0
Same Results vs. previous column				113	115	119
Excluded Tests from the Comparative				15	13	14

Figure 5: Comparative of results of Test Plans performed in a router.

13 **PC REQUIREMENTS** (PC not included)

- List of PC Requirements:
  - ▶ Windows 7 or higher.
  - ▶ 4 GB RAM.
  - ▶ 5 GB available hard drive space.
  - ▶ A WiFi port, necessary to allow execution of Test Cases that require WIFI connection (Wireless and HostLan test cases in TR-098 Library). Execution of other Test Cases do not require the existence of this port.
  - ▶ 2 Ethernet ports necessary to allow the execution of Test Cases related to the CPE LAN (access the CPE LAN, traffic Test Cases, Test Cases that require the use of scripts, etc.). Execution of other Test Cases require only one Ethernet port.
- Instalation Requirements:
  - ▶ Deactivate PC's antivirus before starting dBm069 installation. It can be activated again once installation is over.
  - ▶ Start dBm069 as a Windows' administrator.
  - ▶ Connect the dongle key with the license in one of the PC's USB connectors.

14 **TESTING SCENARIOS**

- dBm069 allows two different ways of working with different requirements:
  - ▶ Connection between the CPE and the ACS through the internet. This scenario requires two internet lines, one to connect the CPE and the second one to connect the PC (through a support router).
  - ▶ Direct Connection between the ACS and the CPE: Available for test cases where the CPE WAN side is an Ethernet interface. Communication between ACS and CPE is faster (it does not depend on Support Router connection speed). It does not support PPPoE. There is not internet access, WEB surfing test cases are performed using dBm069 http server.

- Scenario facilities needed to use dBm069:
  1. Connection between the CPE and the ACS through the internet:
    - a. One line with access to the internet to allow connecting the PC where dBm069 is installed to the internet (a support router which is not the CPE to be tested is needed).
    - b. A second line with access to the internet to allow connecting the CPE to be tested to the internet:
      - i When the CPE to be tested is a gateway: it needs to be connected to the internet.
      - ii When the CPE to be tested is a LAN device: a gateway needs to be connected to the internet, and the LAN device to be tested needs to be connected to the gateway's LAN.
  2. Direct Connection: CPE to be tested must be either:
    - a. A gateway with an Ethernet port on the WAN side.
    - b. or a LAN device connected to the LAN port of a gateway with an Ethernet port on the WAN side.
- Any kind of CPE can be tested with dBm069. Test cases requiring to restore the CPE Configuration file can be fully automatically executed if the CPE supports at least one of the following ways of restoring the Configuration file:
  - ▶ ftp: by means of the put command.
  - ▶ sftp: by means of the sput comand.
- Otherwise, it will be still be possible to test that CPE, but it will be necessary to execute those tests as semi-automatic in order to allow manual update of the Configuration file. An example of test cases that require a Configuration file upload is the case of those where a FactoryReset is executed.

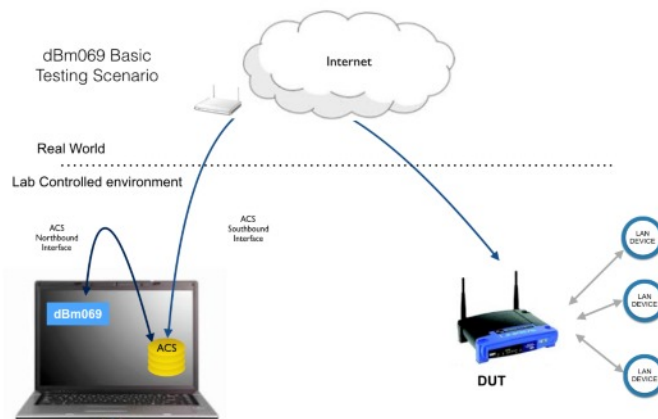


Figure 6: several scenarios are available to perform TR-069 tests.

