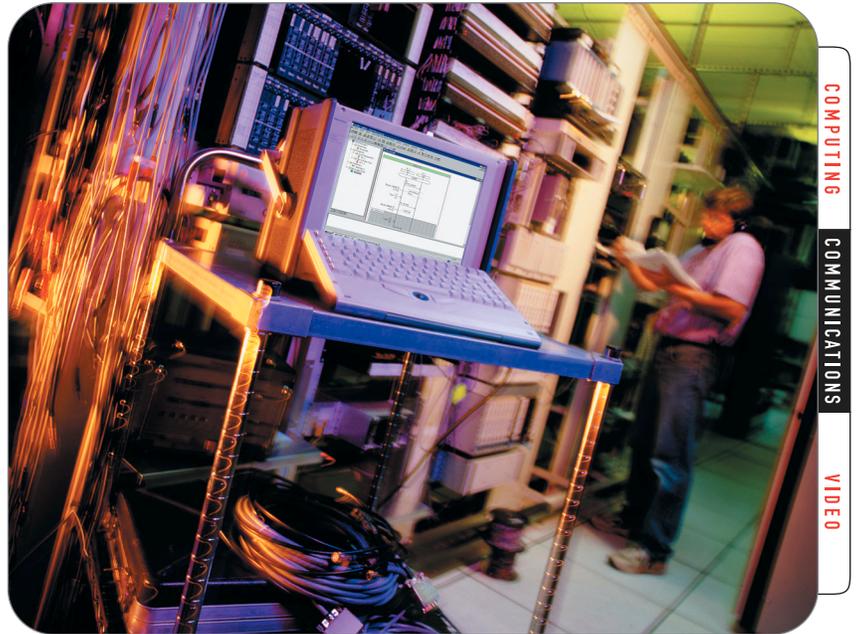


Test Automation – Using the Example of Testing Location Based Services



▶ Introduction

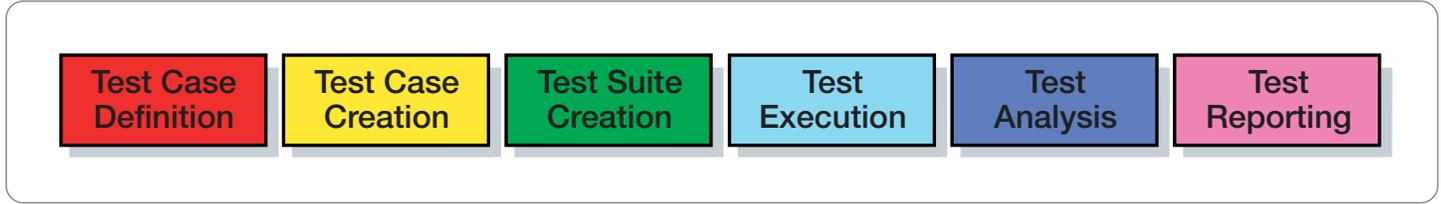
The mobile communications industry is facing an era of uncertainty. Revenue expectations of some years ago did not materialize and analysts around the world are very careful with their predictions about the end of this downturn. Investments are at a minimum and several operators have postponed the introduction of next generation networks. The market consolidates, the number of subscribers stagnates and the average return per user (ARPU) decreases. This situation leads to the need to increase revenue from existing subscribers so that value added services take a prime position in the mobile portfolio. It also drives an even stronger focus on cost containment and time to market.

Key to the success of any new service is acceptance by the target market. Besides the value itself and the time to market, a new service must be easy to use, reliable and of good quality. This requires intensive testing and trials before introduction. Intensive testing and trials, however, are contrary to lowering expenses and fast introduction.

Automated testing and the use of automated test tools is one way to overcome this challenge and to assure the quality of services while meeting today's time to market requirements.

Automated Test

▶ Application Note



▶ **Figure 1.** Phases of testing

The purpose of this application note is to introduce automated testing. The discussion shall be focused around protocol testing of mobile telecommunication networks with the Tektronix K1297-G20 protocol tester and the Tektronix Test Management System (TMS); a tool for automated testing.

You will find information about a general test process and on how automated tests can help to ease your work and increase test efficiency.

- ▶ How to easily run tests overnight, collect the results and prepare them to present to management
- ▶ How to retest new implementations without having to set up the test environment from scratch

This application note covers the following areas, any of which may be useful to the reader as a stand-alone information source:

- ▶ Testing—an introduction
- ▶ Automated testing using the K1297-G20 protocol tester and the Tektronix Test Management System (TMS)
- ▶ Test example—mobile-oriented location request
- ▶ Step-by-step walkthrough of the described test example
- ▶ Regression testing and change management support

Testing – An Introduction

- *“I need a database with a set of test-cases that I can choose from.”*
- *“I want to run my tests without any supervision (e.g. overnight and collect and check the test results the next morning.)”*
- *“Changes on the system under test are frequent. I want to easily rerun the same test I ran before the change without having to reconfigure the whole thing.”*
- *“I do not want to learn the test equipment in detail. I just want an easy-to-use system to execute the tests.”*
- *“While doing regression testing I need an easy way to generate comparative and comprehensive test results that I can present to management.”*

These and other statements indicate the need for automated testing and the need for tools that enable faster, more efficient and more accurate tests.

Tests of any kind can be split into several phases. The responsibility for the test phases might all be with one test engineer or spread over different departments of a company. Nevertheless, all phases will need to be addressed at some point.

Due to accelerated technology development in the core and mobile telecommunications sectors and because of more frequent and faster introduction of new services, the industry anticipates an increased demand for testing in general.

The industry is facing the following challenges:

- ▶ Shorter time to market
- ▶ Higher quality and warranty expectations from customers
- ▶ Heterogeneous test-network environments
- ▶ More complex measurement requirements and applications
- ▶ Huge amounts of test data

This situation calls for tools that address these challenges—tools that support the user from test case definition to test execution to test reporting—and tools that support all phases of testing with the following attributes:

- ▶ Early protocol availability
- ▶ Intrusive testing
- ▶ Functional testing
- ▶ Automated testing
- ▶ Regression testing
- ▶ Change management support
- ▶ Test management
- ▶ Scalable performance

Here's an example of the benefits of three of the above attributes: automated testing, regression testing and change management support.

The benefits of automation are significant and can save up to 80% of manual test time. What once took several days now takes only hours. Labs and test equipment that once ran only during working hours now run around the clock—without supervision, resulting in reduced cost per hour for the lab and increased capacity. Systems can be tested faster and more thoroughly, which results in faster time to market and higher quality.

Automated Testing Using the K1297-G20 Protocol Tester and the Tektronix Test Management System (TMS)

An integral part of the industry standard Communications Protocol Tester K1297-G20 is the Message Sequence Chart (MSC) graphical tool based on [ITU-T Z.120]. Users have full control over message flow and message content. Virtually any equipment behavior (correct or faulty) can be simulated with the MBS/MSC tools, making it particularly useful for detail level testing and troubleshooting. It is also ideally suited for validation and conformance test procedures.

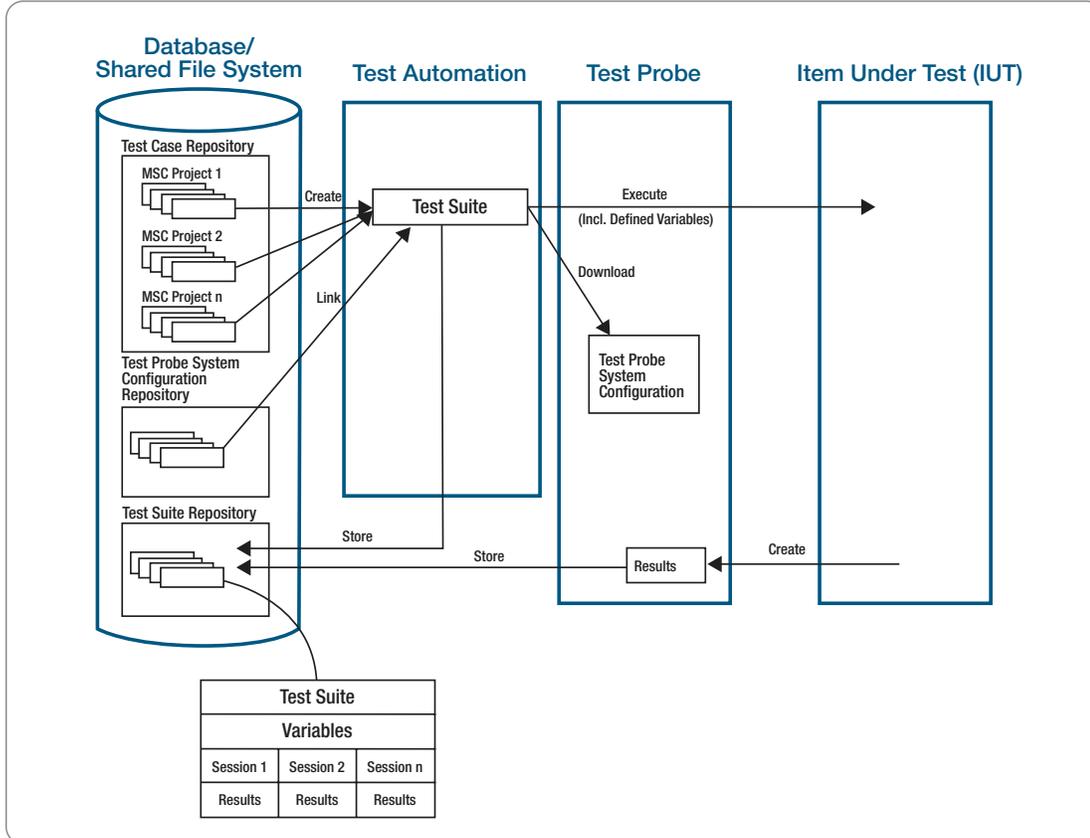
Individual simple test steps can be implemented using the MSC methodology and combined with more complex, high-level MSC (HMSC) test cases. Several test cases constitute a test project. MSC projects are a very efficient and economic alternative to the TTCN-based test suites.

The Test Management System (TMS) complements the MSC-based test projects, or test suites, and allows the user to prepare test sessions for unsupervised execution.

Before test case implementation, the user should consider the immediate and possible future use of every test case. The test case definition phase is as important as the test case implementation itself. The better the test case design, the easier the test. The design needs to reflect that every test case is a closed entity that can be reused in different test suites for different test purposes.

Automated Test

► Application Note



► **Figure 2.** The process of automated testing

Test automation refers to the automatic execution of a series of test cases. Figure 2 illustrates the general process of test automation.

- Single test cases are taken from a test case repository and combined to form a test suite. The test cases must be defined and stored in the repository beforehand.
- Additional information to automatically set up the test environment (Test Probe System Configuration) may be linked to the test suite. This ensures that the test suite is always executed with the correct test environment.
- The assembled test suite will be stored in a test suite repository. This enables use of the test suites, including the linked information, for other test runs (regression testing).
- The test cases that are assembled to a test suite might contain variables that are to be set before execution. This, of course, depends on the test purpose and is to be done on a case-by-case basis. Nevertheless, for reproducibility of results and in order to perform regression testing, these variables are stored together with the test suite.

- We call the process of executing tests a “test session.” Every session is associated with its unique results, which are created during a test run and stored with the test session.
- Change management is realized by automatically storing logically associated data at one location. Variables are always stored along with the test suite, whereas test results are stored with the test session.
- A test suite might also have linked information that applies to all sessions below the suite (in our case, the test probe system configuration).

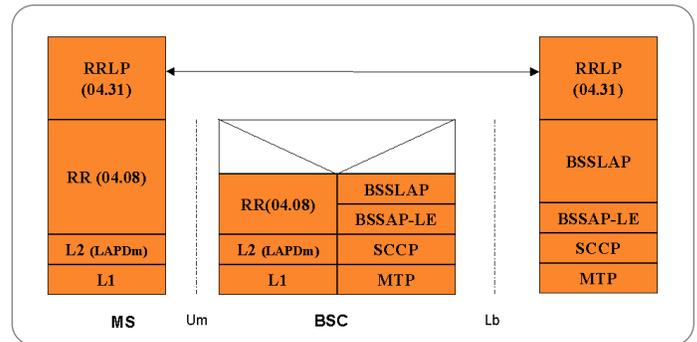
Test Purpose - Mobile-Oriented Location Request

In this section, we describe the testing of Location Based Services by using a tool for test automation (Test Management System—TMS).

- ▶ We will perform a mobile-oriented location request and concentrate on the Lb interface that connects the Base Station Subsystem (BSS) and the Serving Mobile Location Center (SMLC), where the BSS is a combination of the BSC and BTS.
- ▶ Mobile-oriented location requests are used in applications such as *WhereAmI* or *FriendFinder*. All location-based applications that are initiated by the end user are subject to this workflow.
- ▶ The positioning method taken as base for this example is TA (Timing Advance).
- ▶ The test automation will be demonstrated by creating a test suite that contains three times the location request.
- ▶ This test suite will be created and executed by using test automation software (TMS) and will run automatically.

In order to fully understand the location request, we first need to look at the complete message flow over all involved interfaces and to the BSS'-based protocol stack, shown in Figure 3.

Here we focus on the messages between the BSC and the SMLC (Lb Interface), shown in Figure 4.

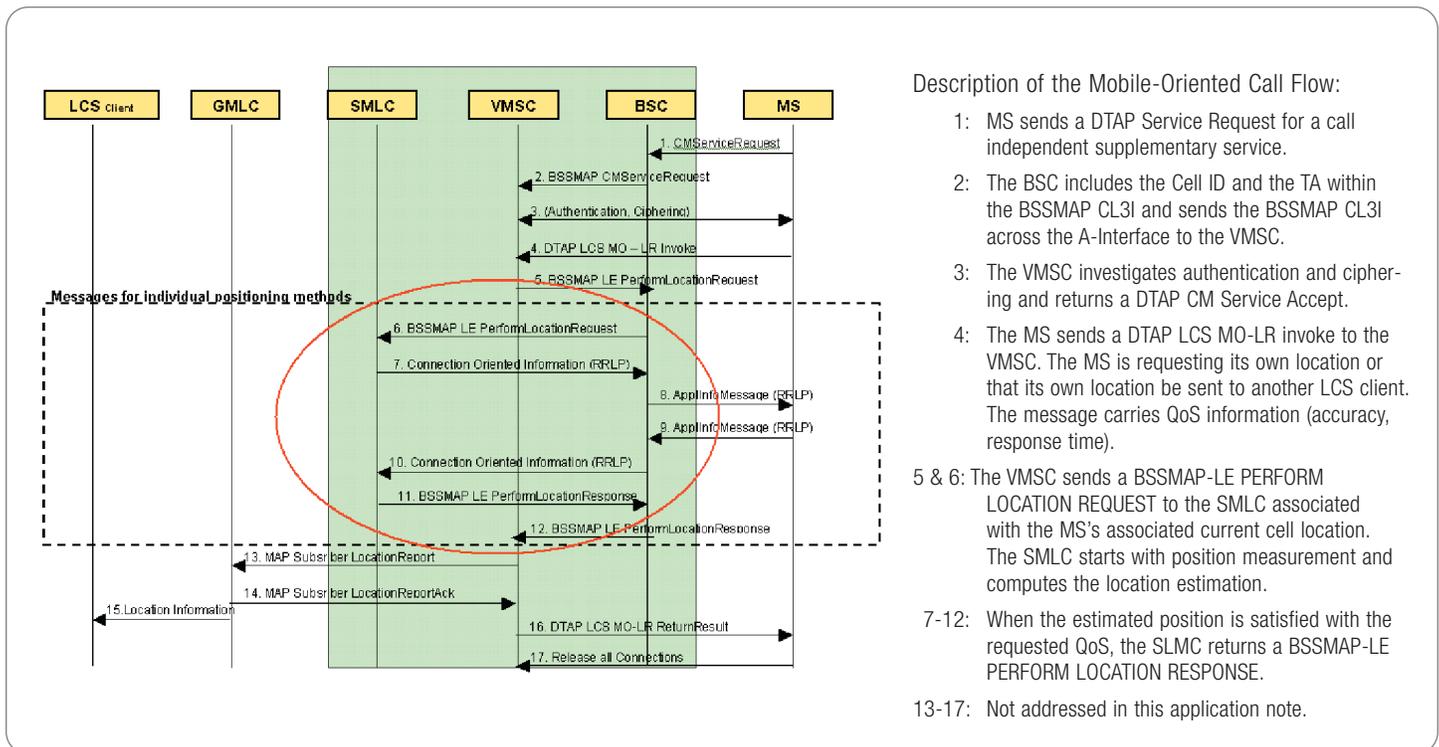


▶ **Figure 3.** BSS-based SMLC protocol stack

1. BSS-based means that the communication is performed via the Lb interface and not via the MSC and the Ls interface (NSS-based). Whether the communication is BSS- or NSS-based depends on the individual implementation of each operator.

Automated Test

► Application Note



► **Figure 4.** Call flow of a mobile-oriented location request (BSS-based SMLC)

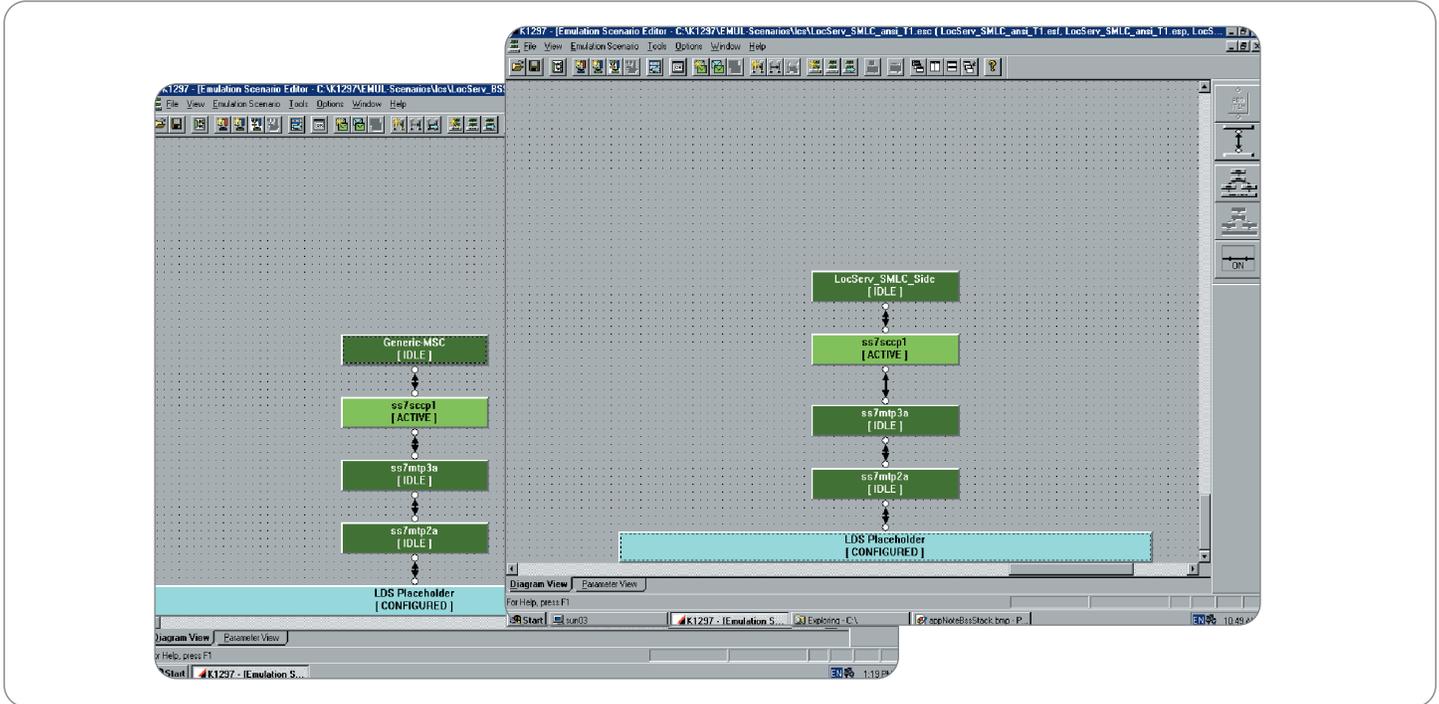
Test Preparation

- "Changes on the system under test are frequent. I want to easily rerun the same test I ran before the change without having to reconfigure the whole thing."
- "I do not want to learn the test equipment in detail. I just want an easy-to-use system to execute the tests."

Earlier we described a general applicable test process that starts with test case definition and ends with test reporting. But before applying this process to our example, we first need to prepare the test environment.

During test preparation saved configurations can automatically be loaded to the test equipment. In most cases intrusive testing must be performed on a subsystem. In order to test a complete message flow, the relevant elements of the network need to be emulated.

Network Element Emulations on the K1297-G20 are designed to allow network elements (such as HLR, SMLC, etc.) to be replaced by the Emulation for testing purposes. All parameters—phone numbers, IMSIs and more—are configured through menus. Once activated, the Network Element Emulation will automatically respond to incoming messages and/or initiate calls and procedures.



▶ **Figure 5.** Emulation stacks for the BSS and SMLC side

In this case, where a mobile-oriented location request is tested, we have a test case that is initiated by the BSS-side. The lower layers of the BSS (up to the SCCP) are emulated, whereas the Message Sequence Chart (MSC) simulates the upper layer (BSSMAP-LE, BSSLAP and RRLP). This layer generates the messages with which we're testing the SMLC. The SMLC represents the SUT (System Under Test)².

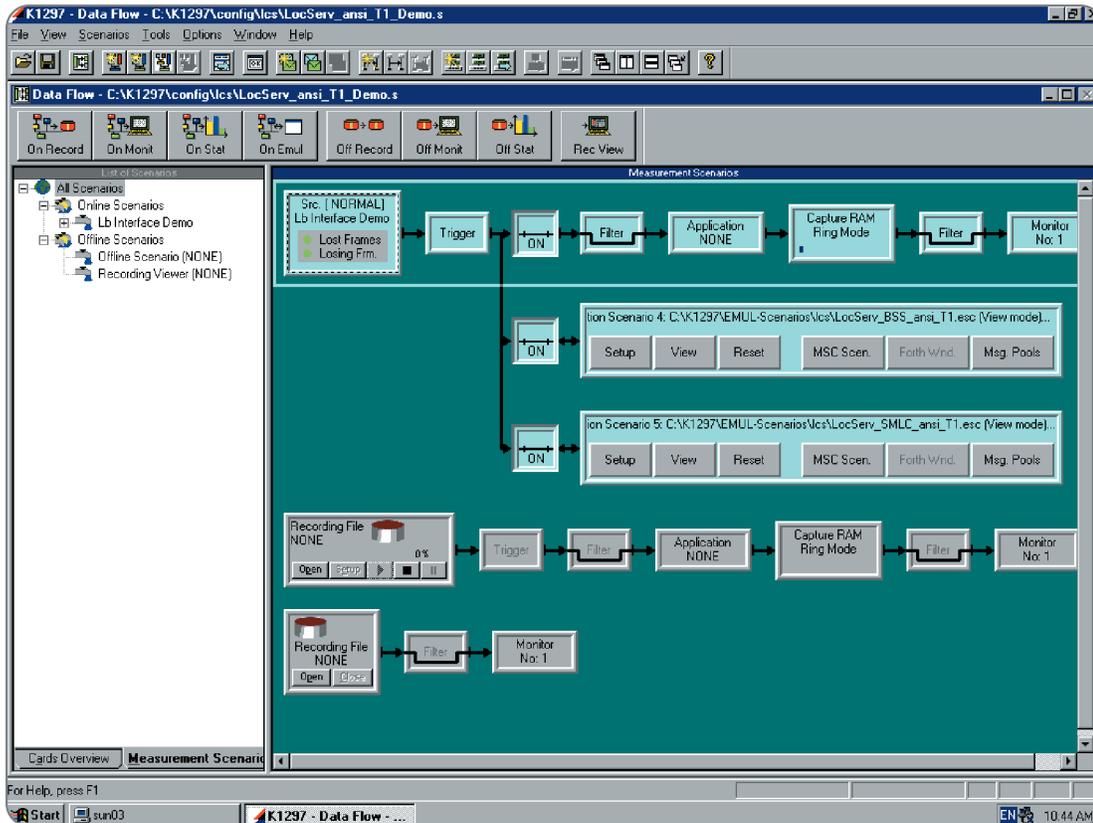
The test case is represented by the Generic-MSC layer shown in Figure 5. We use the Generic-MSC layer as a placeholder for test cases that are being uploaded while the test is being executed.

In this case we're testing only the mobile-oriented location request. For other test purposes (and in order to test the complete functionality of the SMLC) we could think of a large variety of test cases. The emulation stack remains the same and the placeholder would be uploaded sequentially with different test cases.

2. In our case, where we do not have a real SUT, we use an emulation to create that same behavior. We call this a "back-to-back scenario" (a scenario where we test against our own implementation rather than against a real switch).

Automated Test

► Application Note



► **Figure 6.** K1297-G20 pipeline concept

The K1297-G20 pipeline concept shown in Figure 6 represents a measurement scenario set up for the purpose at hand. The first branch represents the monitoring pipeline, in which we monitor the Lb interface that connects the BSS to the SMLC.

The other two branches represent emulations of the two network elements described in Figure 5. In a real test environment we would emulate only the BSS-side and connect the K1297-G20 to a real SMLC. Each branch represents one side of the measurement scenario (BSS and SMLC).³

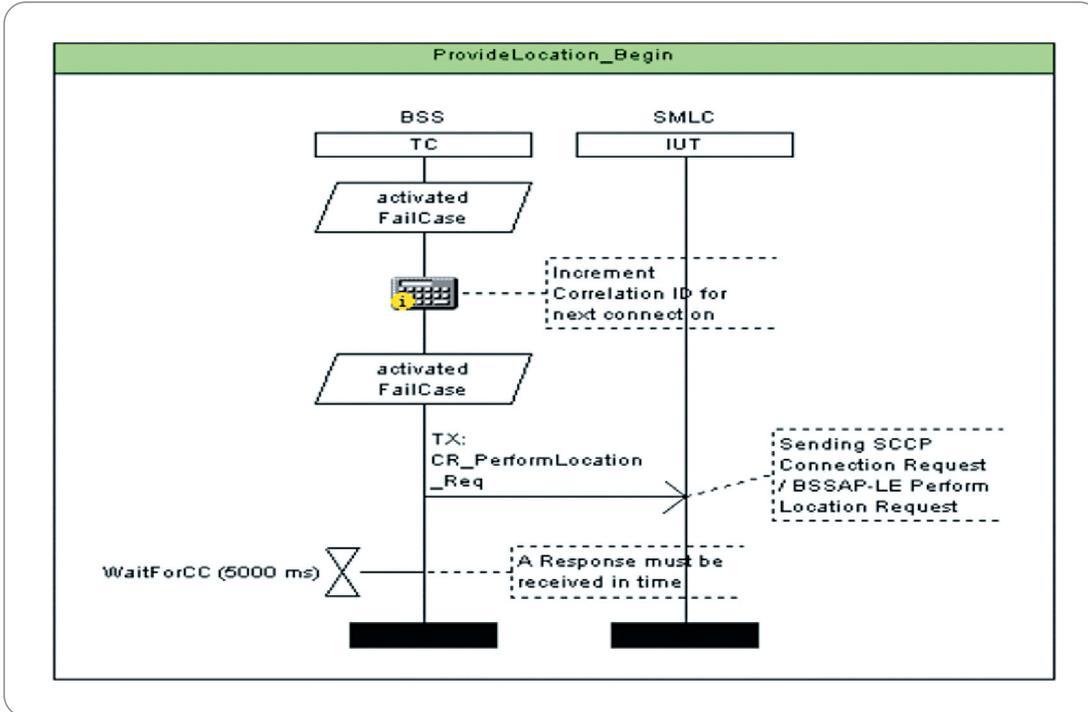
Test Case Definition

- "I need a database with a set of test cases that I can choose from."
- "I want to run my tests without any supervision, (e.g. overnight, and collect and check the test results the next morning.)"
- "Changes on the system under test are so frequent. I want to easily rerun the same test I ran before the change without having to reconfigure the whole thing."

Test case definition always depends on the test purpose and on possible re-uses of the test case. Once again, the better the test case design, the easier the testing.

In this case, the test purpose is to perform a mobile-oriented location request concentrating on the Lb interface. The message flow for this test case has been described in the next section, *Test Case Creation* on page 9.

3. In our case, where we do not have a real SUT, we use an emulation to create that same behavior. We call this a "back-to-back scenario" (a scenario where we test against our own implementation rather than against a real switch).



▶ **Figure 7.** Provide_Location_Begin

Test Case Creation

Test case mobile-oriented location request was created using Tektronix' K1297-G20 message sequence chart and its associated message building system. These allow users to graphically design and modify test cases. Users have full control over message flow and content. The first part of this chapter describes the creation of the message flow, whereas the second part concentrates on the individual messages.

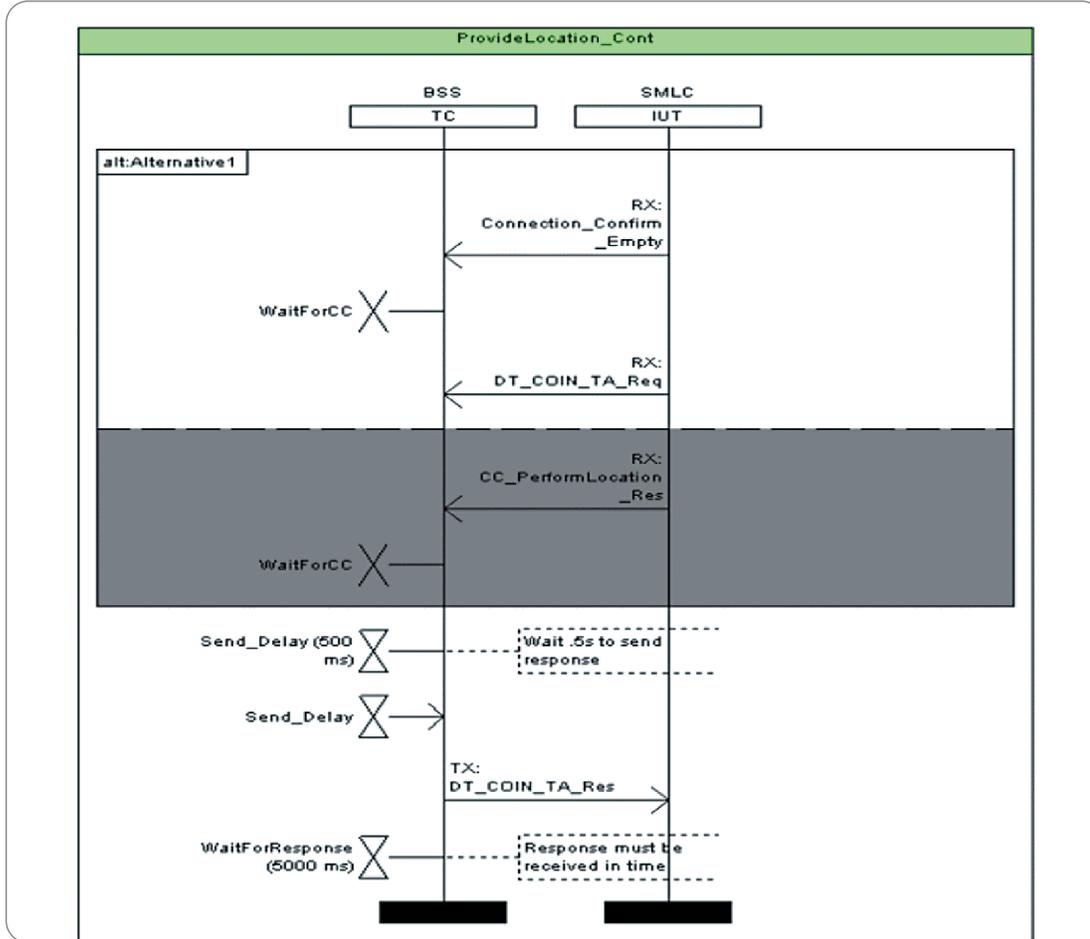
For a better overview, we have split the test case into three sections:

- ▶ Provide_Location_Begin
- ▶ Provide_Location_Continue
- ▶ Provide_Location_End

After sending the message "CR_PerformLocation_Req," a timer is set to 5 seconds. A response from the IUT (here the SMLC) must be received within these 5 seconds. If the response is not received, the sequence will branch to the "FailCase," the test case will be aborted and a Verdict (in this case "Fail") will be generated. The "FailCase" is an alternative to a condition, which in this case is the expiration of the timer (see Figure 7).

Automated Test

► Application Note

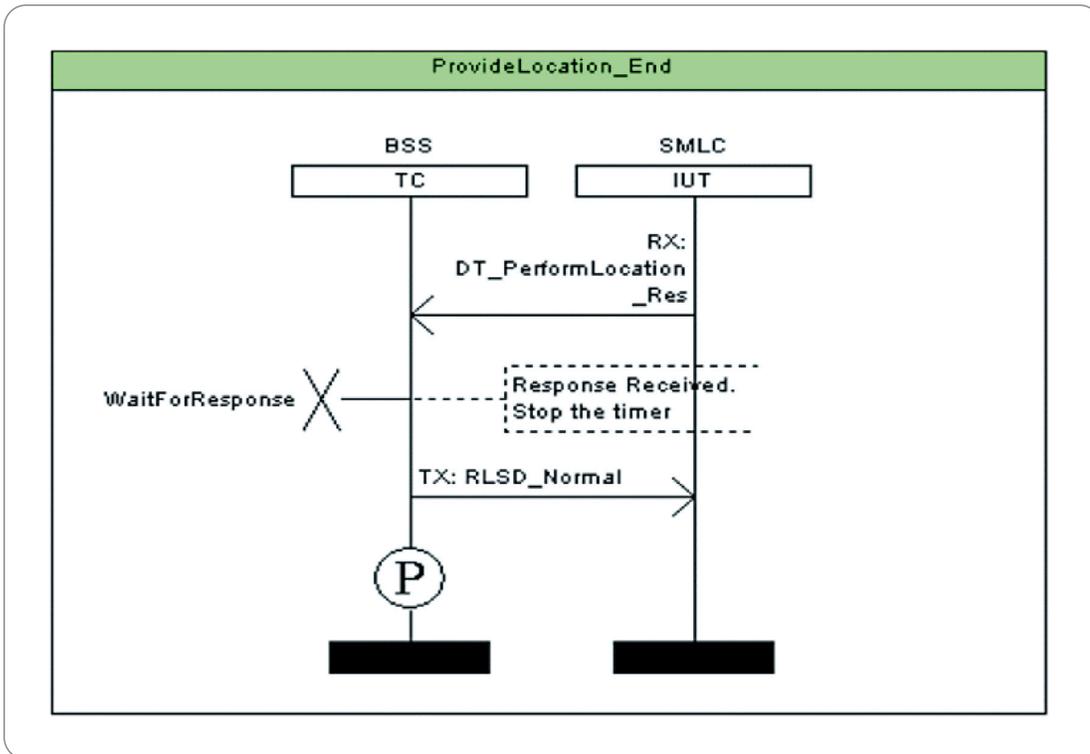


► **Figure 8.** Provide_Location_Continue

The test case continues with the section "Provide_Location_Continue" (Figure 8) in case a response from the SUT is received.

This example refers to the positioning method timing advance (TA), where the position of a mobile subscriber is calculated using the cell ID and the length of time it takes a signal to travel from the BTS to the mobile and back. For this calculation, the SMLC needs these two parameters. In our message flow, this information is represented by the messages DT_COIN_TA_Req and DT_COIN_TA_Res. The SMLC asks the BSS about these two values (request) and gets the response, with which it computes the X Y position of the mobile subscriber.

This is demonstrated by the MSC chart shown above. The timers give the SMLC time to initialize and calculate the position of the mobile subscriber and to timeout and branch to the alternative "FailCase" if no response is received.



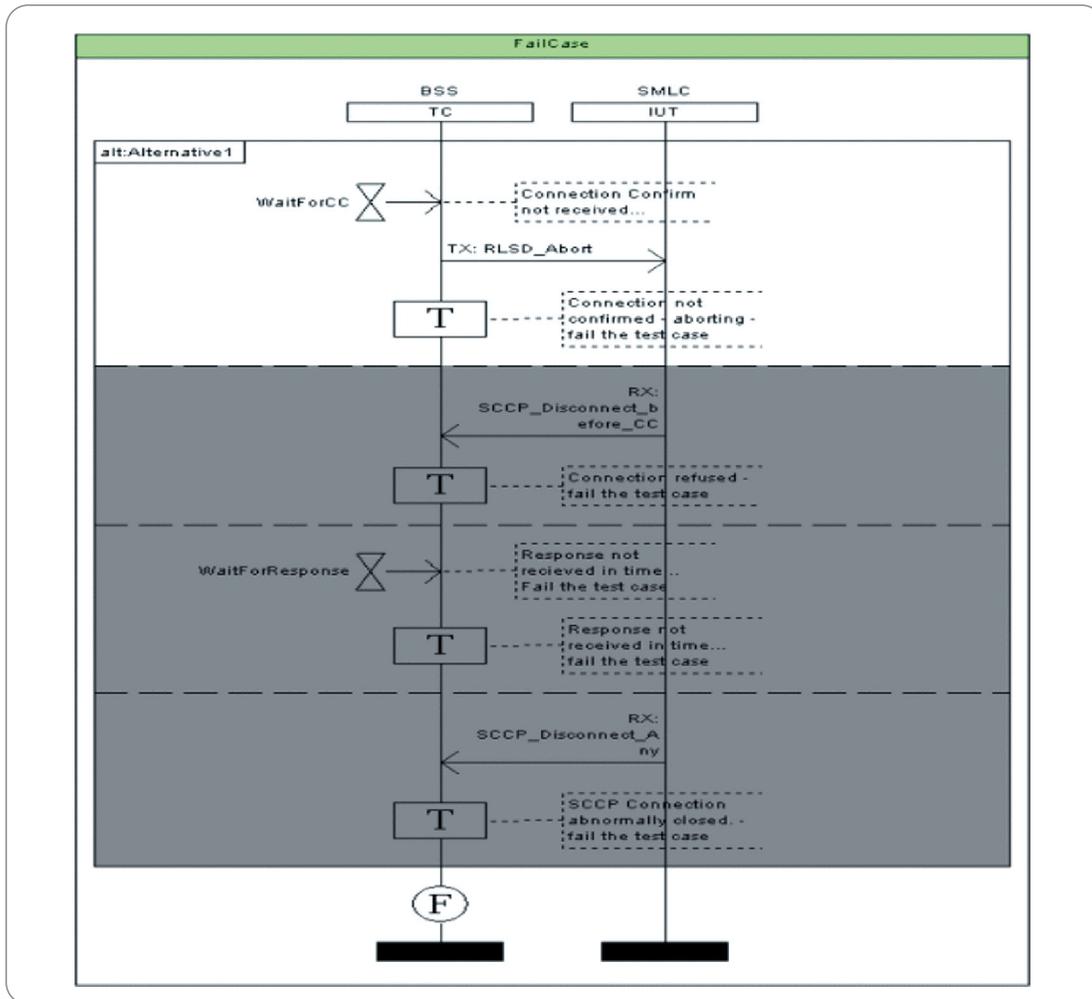
▶ **Figure 9.** Provide_Location_End

If the SMLC reacts as expected, the message flow will continue, concluding with Provide_Location_End.

In this example, the mobile receives its position and, in return, releases the connection. The generated verdict "PASS" indicates that the test case ran successfully.

Automated Test

► Application Note

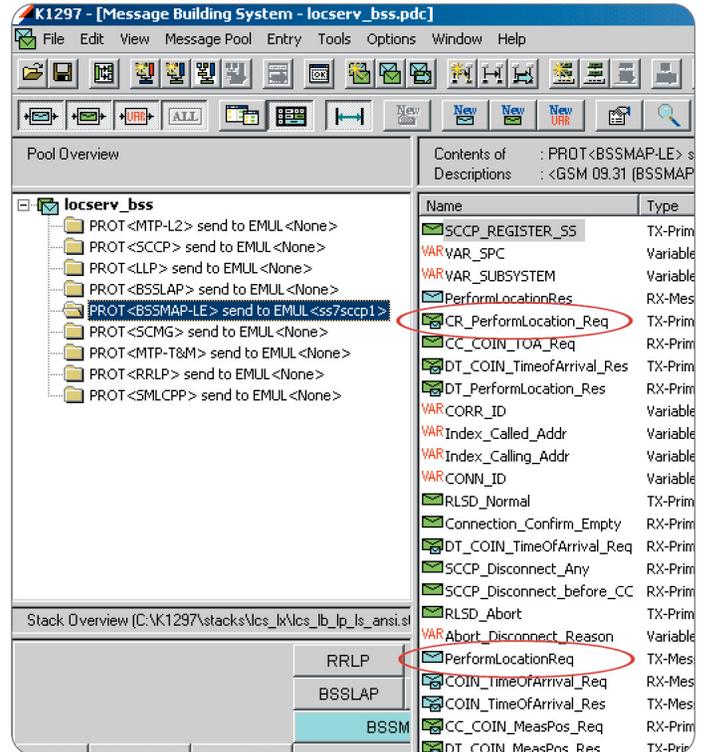


► **Figure 10.** Background Chart "FailCase"

Misbehavior of the mobile or SMLC will always cause the message sequence chart to branch into the alternative "FailCase."

FailCase will cause the SMLC to disconnect from the mobile (in our case BSS) and produce the verdict "Fail."

All messages used in an MSC will be represented in the Message Pool, as shown in Figure 11. This message pool is referenced by the MSC test case. As an example, please see the message CR_PerformLocation_Req, in which all parameters are defined.



▶ **Figure 11.** Messages represented in the Message Pool

Primitive Overview		Contents of : CR_PerformLocation_Req (TX-Primitive, SAP: ss05ap(connection to the SCCP service user log_user_id 1))	
		Description :	
Name	Size	Description	
PR length	2 Bytes	Length of the whole ASP	
PR corr_id	4 Bytes	Connection identifier for connection establishment	
PR called_addr	4 Bytes	Index of the called user from the REMOTE_USER data base	
PR calling_addr	4 Bytes	Index of the calling user from the LOCAL_USER data base	
DT user_data	16 Bytes / 1024 Bytes	Transported user data	

Automated Test

► Application Note

Configuring the TMS and Test Suite Creation

- *"I need a database with a set of test-cases that I can choose from."*
- *"I want to run my tests without any supervision, (e.g. overnight and collect and check the test results the next morning.)"*
- *"Changes on the system under test are frequent. I want to easily rerun the same test I ran before the change without having to reconfigure the whole thing."*

Every test case will represent a single test that will solve one specific issue. Single test cases are mainly used for manual functional testing, which is usually done during the early design and system integration phase. For later stages of system integration, more complex testing is used to test different IUT behavior with one test run. This leads either to very complex test cases that cover many different purposes or to a collection of independent test cases, which is called a test suite.

A test suite is a collection of single test cases that can be stored and executed as one. Test suite results always contain the results of all test cases that are part of the test suite.

Test suites are frequently used for regression tests but can also easily be rearranged or changed in order to meet a different or adapted test purpose.

For the test suite creation we are using TMS. The TMS tool allows automated and regression testing. It allows users to select MSC test cases and to build, modify, execute and analyze test suites and test campaigns either locally or remotely.

Configuring TMS

Before the test suite creation can be started, the most important information to be provided to the TMS is:

- The location of the work directory
- The IP address or name of the K1297

The work directory sets the shared file system on which all test cases, system configuration and test results are located. This shared file system can also be part of any change management system. Please refer to the section *Change Management via Shared File System*, on page 23, for further details.

The second option to set is the remote information, which defines the address of the K1297-G20 on which we are going to load the system configuration and execute the test suite.

The TMS application can be started from the "K1297-G20 Protocol Tester" program group of the PC Start Menu. The install directory for the TMS executable and configuration file is the same as for the DPE server C:\K1297\udpe. The name of the executable is msc-tcm.exe, which starts the TMS main window and calls runtms.exe, which opens a command prompt window, (Figure 12).

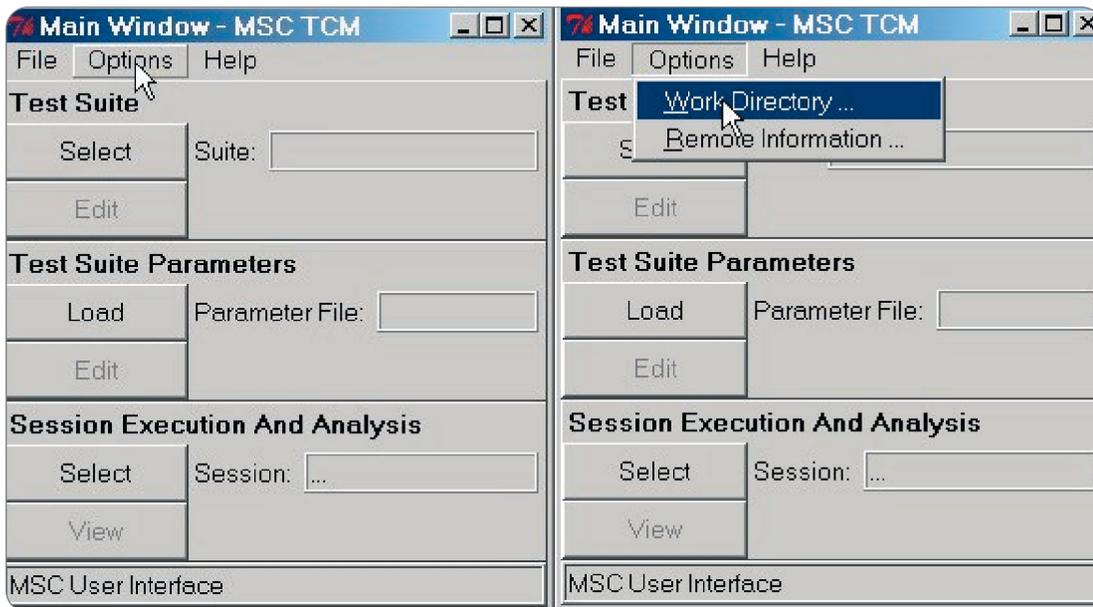
To check or modify the current settings for the location of the work directory, open or restore the main window of the MSC TCM (TMS) and click "Options", then select "Work Directory" from the drop down. This will open a new window called "Choose Work Directory," (Figure 13).

The last saved work directory path will be displayed in the windowpane and the entry field. If the TMS application is able to establish a network connection to the work directory, subdirectories will be displayed in the browser pane of the window.

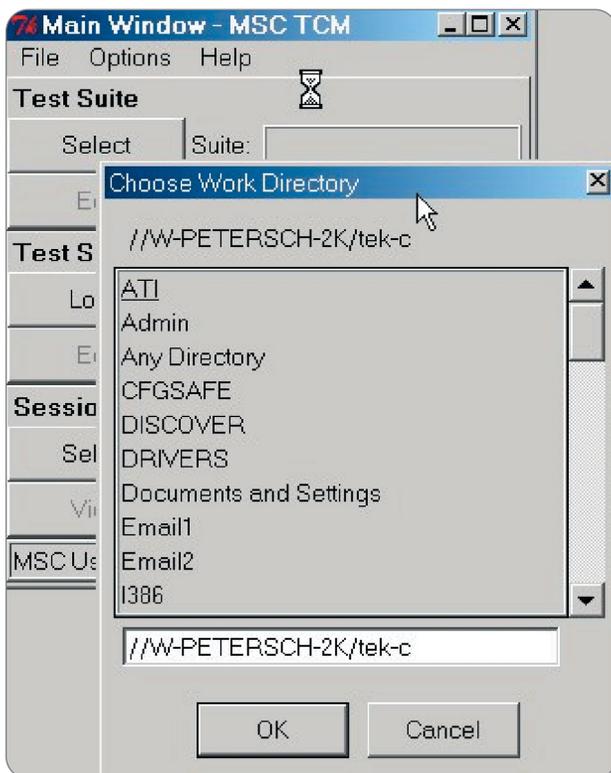
The most recently used values for the TMS options are read from the default configuration file "tboxmsc," which can be found in the install directory of TMS C:\K1297\udpe.

Settings can be saved and retrieved using the "File" drop down menu.

To enter a new work directory, type the file name in the entry field of the "Choose Work Directory" window. After depressing the "Enter" key or clicking the "OK" button, the "Choose Work Directory" will close without showing the new work directory path. Upon reopening the "Choose Work Directory" window, (assuming TMS is able to establish a network connection to the new work directory) the new path name will be displayed in the entry pane and subdirectories will be displayed in the browser pane.



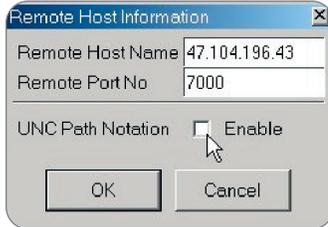
▶ **Figure 12.** TMS Main Windows



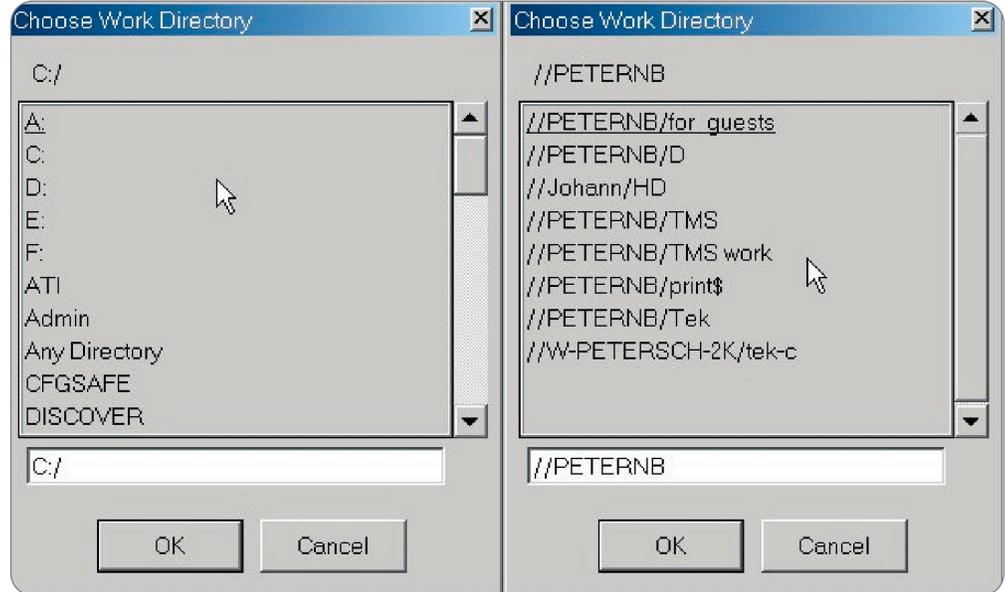
▶ **Figure 13.** Choose Work Directory browser window

Automated Test

► Application Note



► **Figure 14.** “UNC Path Notation” Checkbox in the “Remote Host Information” window



► **Figure 15.** Drive letters versus UNC names convention

The work directory path can be entered by using universal naming conventions or by using drive letters. The format (UNC or drive letters) is controlled by a checkbox in the second dropdown function of the main window “Options” dropdown, the “Remote Information...” function.

Selecting the Remote Information function opens the “Remote Host Information” window (Figure 14). If the UNC path notation checkbox is not checked, drive letters are used instead of UNC path names in the “Choose Work Directory” window (Figure 15).

- UNC checkbox unchecked means all assigned drive letters and directories in the currently selected drive letter are displayed.
- UNC checkbox checked means all locally available shared folders and network connections to shared folders on other computers are displayed (//Johann/HD and //W-PETERSCH-2K/tek-c are active network connections).

For further information about the Remote Host Information window, remote access and taking control of the K1297-G20 tester, please refer to the section *Taking Control* on page 20.

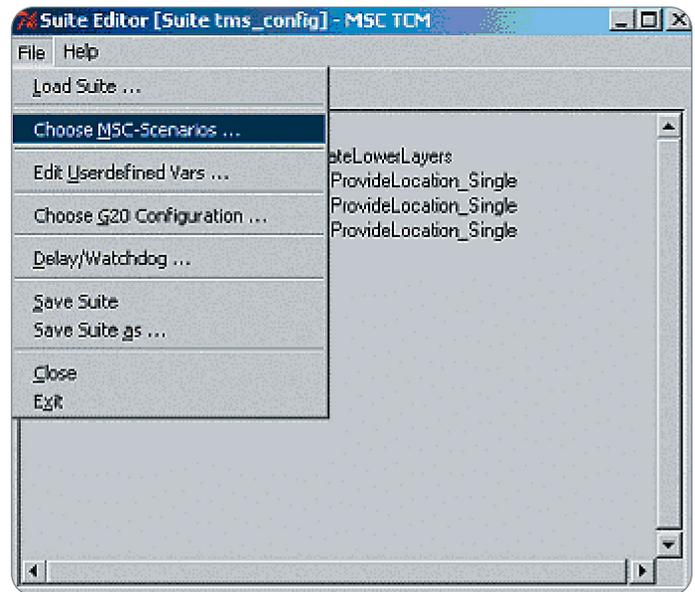
Test Suite Creation

Having set the work directory, we can now start to assemble the test suite. The basic concept is to work through the pull down menu from top to bottom.

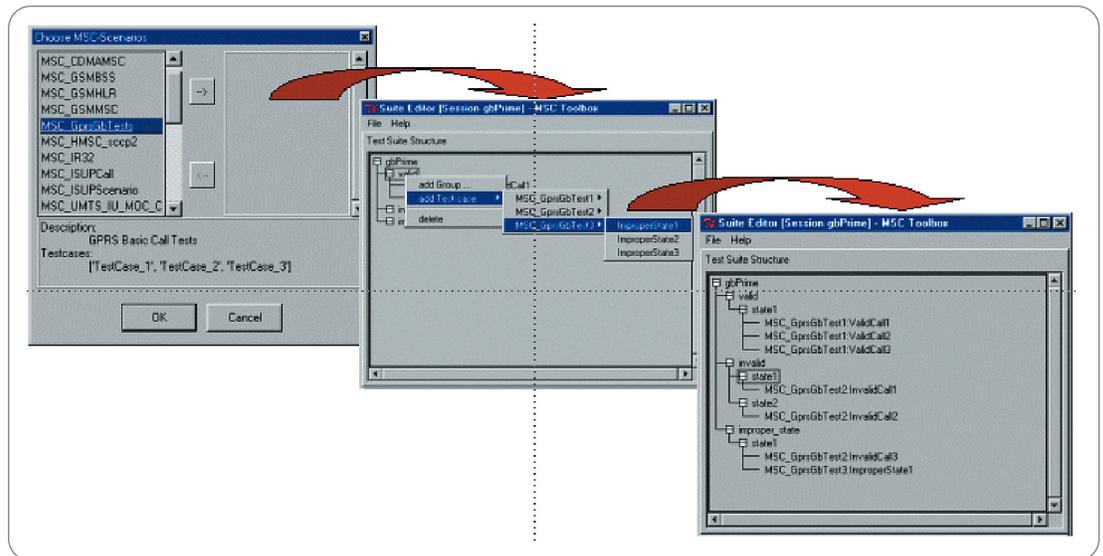
The test suite is a collection of previously defined test cases. For large test suites, it is beneficial to collect test cases as groups and subgroups. Test suites enable users to repeat test cases as many times as required without physically copying the test case itself. Test suites might contain test cases from different MSC scenarios (also known as MSC projects). See Figure 16.

The first step is to select the MSC projects that contain the test cases to be assembled in the test suite. The test cases contained in an MSC project depend on the test case design. Having selected the MSC projects, the user may begin to assemble the test suite. With a right mouse-click, users can create groups, subgroups and add individual test cases to the test suite, as shown in Figure 17.

After the test suite is built, users may choose to link a system configuration to the test suite. A test suite that is linked to a system configuration of the test probe enables reruns of the same test under the same conditions as many times as is needed. This enables proper regression testing without having to remember the details of a previous test and system configuration.



▶ **Figure 16.** Choose MSC scenarios

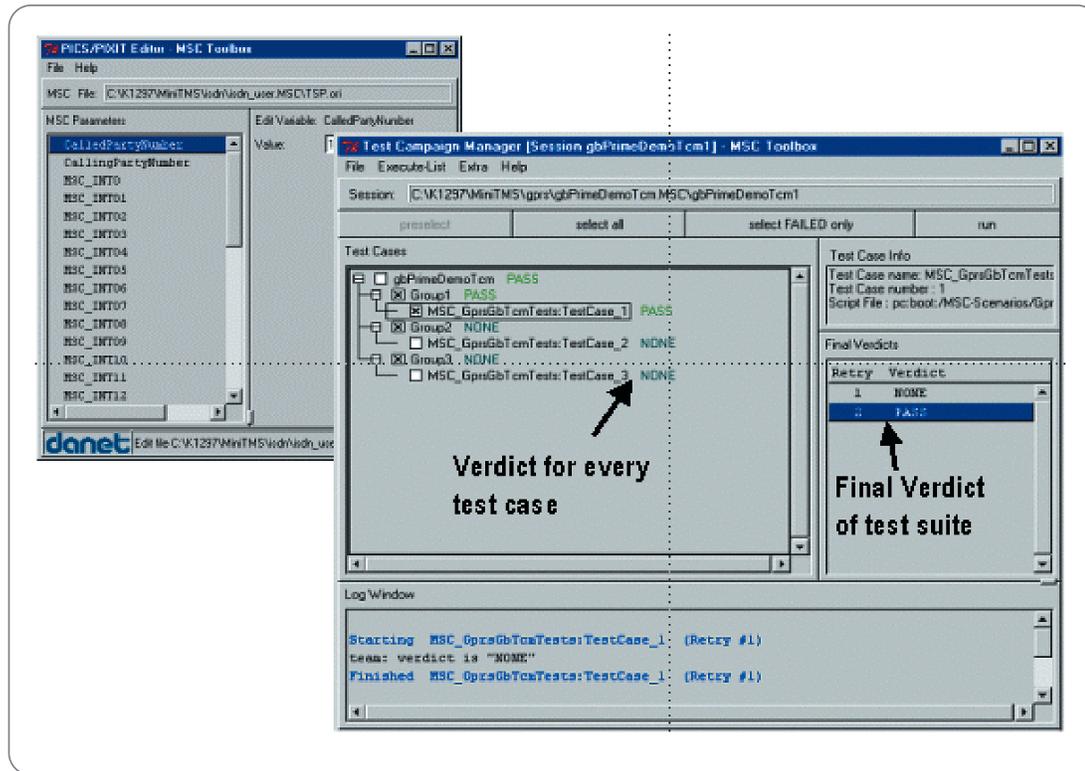


▶ **Figure 17.** Test suite assembly⁴

4. Test suite illustrates the general process. It does not correspond directly to the location request.

Automated Test

► Application Note



► **Figure 18.** Defining variables and test suite execution⁵

Test Execution

Having talked about the test suite assembly and some advantages over single test cases, we now come to the test execution phase.

The test case definition, test case creation and test suite assembly are the basis for the complete test process. The test execution and the following test analysis can only be accurate and informative if the test suite describes what has been designed. The test case designer must have the potential result already in mind while defining single test cases.

It is the test execution phase where automation becomes important. Test time is very precious and must be used very effectively.

The example shown here depicts a very small test suite that will only serve to demonstrate the general functionality. Test suites with several hundred test cases have been implemented for other projects.

The user can start the test execution right after having set the values for variables defined during test case creation, as seen in Figure 18. All variables are stored together with the test suite. This assures that in case of regression testing, the same variable values are used.

A test suite can be started completely, by group, subgroup or even by single test case. The test progress can be observed during test execution and results for single test cases can also be seen. The complete result file and all associated recordings are only stored and accessible after the completion of the entire test run.

Test Analysis

Test analysis always depends on the test purpose, test suite and on the information that is needed. Tools can only assure the generation of test results, and a test run will only be as successful as the test analysis to be performed. In automated testing, where the user is not constantly observing the test process it is important to analyze the results offline. In our case, two types of data mainly support the test analysis:

1. Test cases that will be automated by using the Test Management System (TMS) must generate a verdict at the end of the test case. This verdict indicates that the test case has been terminated and also gives a result for the user. This result file can also be converted into .csv format that can then be used for any kind of post-processing.
2. The TMS also generates a recording file (rf5) for every test case, which can be analyzed offline by using K1297-G20 monitoring capabilities.

Both types of data are stored together with the test session. This assures their relationship for later analysis.

With these two results, every test and test run can be thoroughly analyzed.

Test Reporting

When performing a regression test it is crucial to provide a report that states the difference from previous results. Regression testing is meaningless when reporting or result comparison are not performed.

The reports themselves are specific to the test purpose, test session, the user and to the target audience of the report. This complicates automatic report generation as these will only meet the requirements of some users. Every audience has a different expectation of the test report, including its parameters, the depth of data and the representation of the test results. Sometimes, the same test results must be presented in different ways for different audiences. Therefore, we apply the rule that test reports can only be supported by the generation of test data with a maximum depth.

Post-processing tools or other programs such as Microsoft® Excel may be used to generate many needed reports.

Automated Test

► Application Note

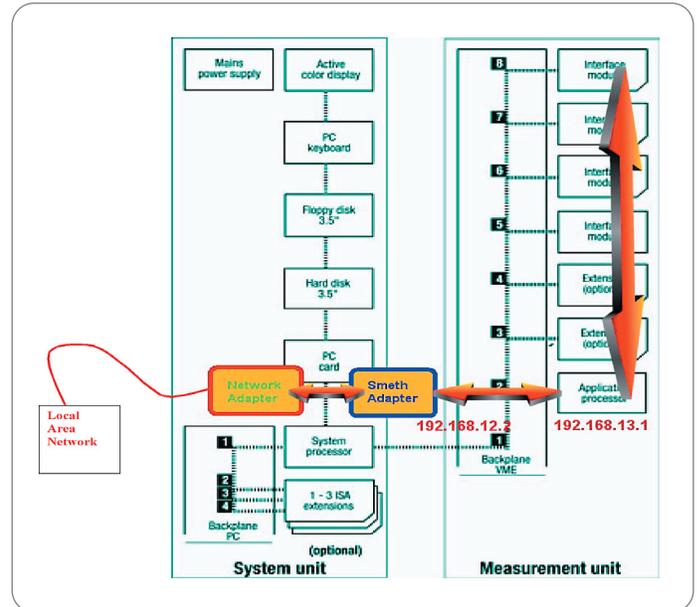
Taking Control

TMS is only one of many ways to automate the operation of K1297 applications. Remote access to the K1297 simulation protocol stack via the K1297's internal local area network and an application-programming interface is key to the automated and unattended execution of TMS test sessions.

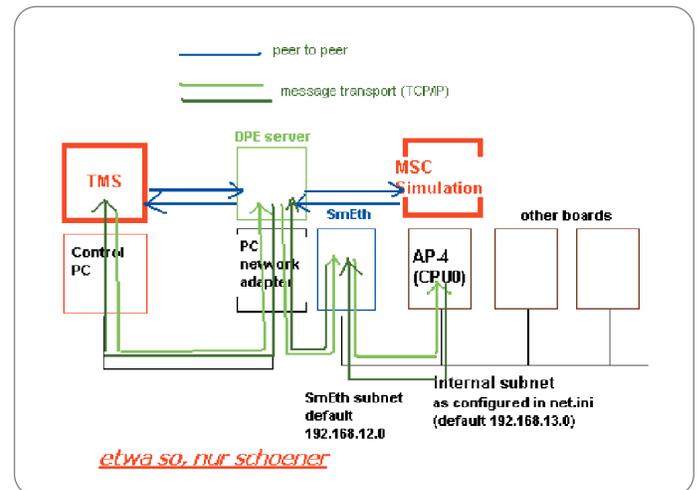
All boards of the measurement unit of K1297 are interconnected through a local area network in the non-routable IP address block space of 192.168.0.0 used for private networking subnets shown in Figure 19. This internal network may be configured, and network services can be started individually on each measurement board.

Using an additional special network adapter, the shared memory Ethernet adapter SmEth, the internal subnet is accessible from an external local area network through the K1297 PC Ethernet Adapter, shown in Figure 20.

For configuration settings and utilization of the K1297 subnet, as well as the SmEth adapter, please refer to the Appendix D of the K1297 and K1297 Compact Protocol Tester User Manual, Tektronix part number C73000-B6076-C83-7.



► **Figure 19.** Internal K1297 local area network



► **Figure 20.** Communication between TMS and MSC

The program access between the TMS and the K1297 measurement applications is provided by the Distributed Protocol Test Engine (DPE) server, which connects clients to objects using the Object Management Group's (OMG) Common Object Request Broker Architecture, CORBA. The DPE server is part of the multi-purpose K1297 Remote Operation Package (ROP). For more information about the ROP and its components, please refer to the K1297 Online Help Section "Extras 1: Solutions." For more information on CORBA and the OMG Interface Definition Language IDL specification, please refer to the OMG websites www.omg.org and www.corba.org.

The TMS client connects to the DPE server on the K1297 PC and starts the K1297 measurement application, unless the application is already running, and loads an entire system configuration, including emulation scenario pipelines and the selected MSC project to be executed unattended as defined in the selected session configuration. The TMS displays a log of the progress of the session execution along with the results returned by each test case and prepares a test report after the completion of each test session.

Deployment of the TMS System and the Work Directory

The TMS is a PC application and runs on the K1297 PC or any other PC. The work directory can be created on any of the K1297's local hard disk partitions, or on any PC to which the K1297 and the PC running TMS have network access.

TMS and Work Directory on K1297 Tester

If the K1297 tester is used in a stand-alone mode and is not connected to a local area network, or other PCs cannot establish connections to the K1297, the TMS can be run on the K1297 PC. The work directory must be created on one of the K1297's local hard drives.

Automated Test

► Application Note

TMS on PC, Work Directory on K1297 Tester

If the K1297 is connected to a local area network and other PCs can establish network connections to the K1297, the TMS can be installed and run on a PC other than the K1297 PC. See Figure 21.

Note: The DPE server must always run on the K1297 tester.

TMS on PC, Work Directory on the Same PC

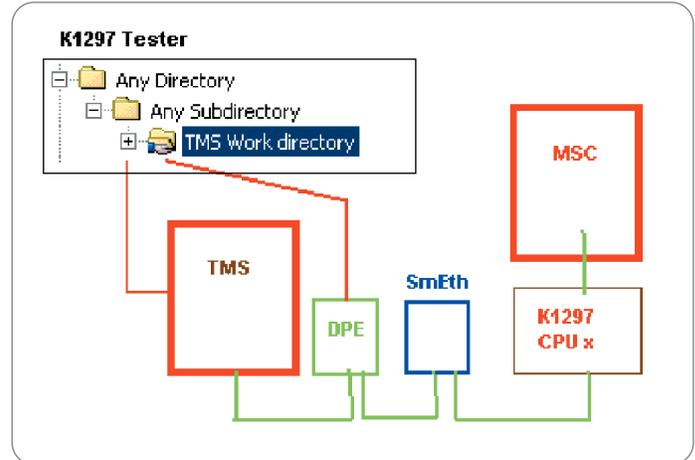
After test suite development has been completed on the K1297 tester, the system configuration information and the MSC scenario information must be copied to the PC on which the TMS has been installed using the special copy function of the K1297 “Pack and Go” archive tool for system configurations and MSC scenarios. (Please refer to the K1297 online help pages.) Because some files contained in the work directory must read and copy during run time of the TMS, network connections must be possible from the K1297 to the PC.

Note: The DPE server must always run on the K1297 tester.

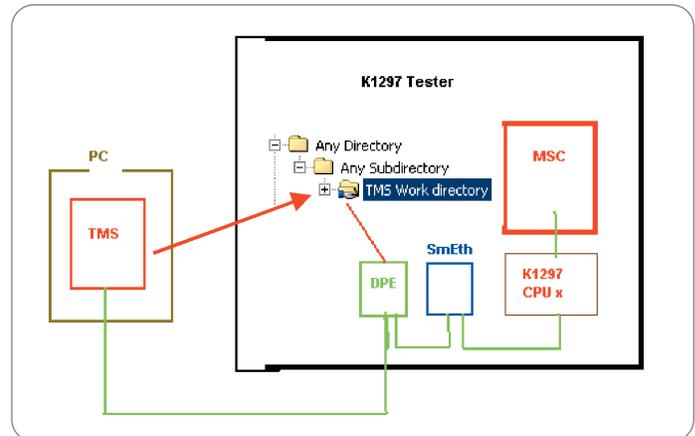
TMS on PC, Work Directory on Other PC or Server

After test suite development has been completed on the K1297 tester, the system configuration information and the MSC scenario information must be copied to the server on which the test suite repository is to be stored, using the special copy function of the K1297 “Pack and Go” archive tool for system configurations and MSC scenarios. (Please refer to the K1297 online help pages.) In this configuration, network connections must be possible between the PC and the work directory and between the K1297 and the work directory.

Note: The DPE server must always run on the K1297 tester.



► **Figure 21.** TMS and work directory on K1297 tester



► **Figure 22.** MSC on PC, work directory on K1297

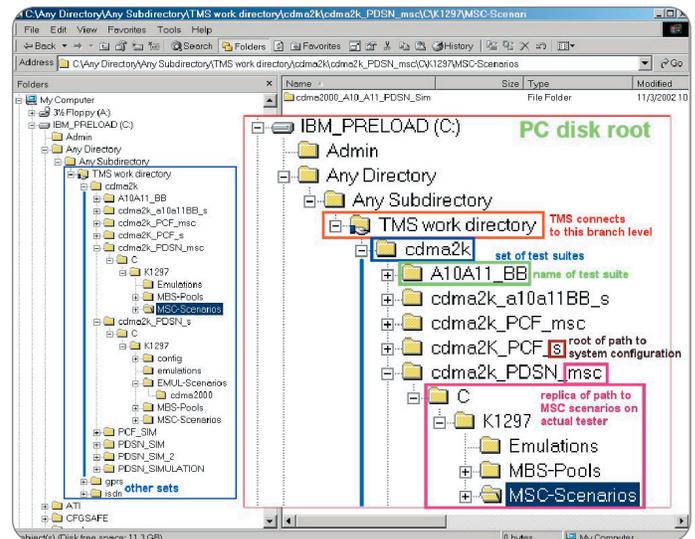
Change Management via Shared File System

TMS supports change management via its shared file system. Regression testing, reproducibility and tracking of tests and their associated results are possible using this approach.

Following is the detailed definition and set-up of such a shared file system, also called a work directory.

The TMS client program expects all information that is related to a specific test run—hereafter called a test session—to be stored in a work directory of the TMS. This directory must be available as a shared network resource, whether or not it resides on the same host as the TMS client or any other host accessible through a local area network. The following example shows the directory tree of a PC on which the work directory for a TMS client resides. The directory (or folder), which is to be the designated work directory for a TMS client, can be located in any directory (and subdirectory thereof) on the designated host computer. The work directory may contain subdirectories to better organize test suites. In the given example, test suites are grouped together in the subfolders cdma2k, GPRS and ISDN, referring to different technologies accordingly (CDMA2000, GPRS and ISDN).

Information about test suites and test sessions is stored in a directory, which will appear in the TMS GUI as “Suite Name.” In the example, the directories named A10A11_BB, PCF_SIM, PDSN_SIM, PDSN_SIM_2 and PDSN_SIMULATION all contain test suite information and will be displayed in the TMS GUI as available test suites. Directories containing the information for K1297 system configurations and MSC scenarios are located on the same level that contains the test suite directories. Because there is no apparent discrimination between test suite directories and directories containing the system configuration information and/or information about MSC scenarios, users should follow the naming conventions suggested in the online help user manual for TMS.



▶ **Figure 23.** Work directory and subdirectories

The example contains three directories with system configuration information and two directories containing MSC scenario information. These directories have been created with the K1297 tester application’s “Pack and Go” archive function. Please refer to the online help manual of the TMS for information about the archive function.

Note: The TMS GUI has built-in browsing capabilities, therefore the TMS work directory may be placed anywhere within a shared hard disk partition or directory (folder). In the example, instead of enabling shared access to the directory “TMS work directory,” any directory above it in the tree hierarchy, including the entire C: drive, could be allocated as a shared directory.

Shared folders and network connections can be displayed by using the computer manager function in Windows 2000 (right-click on “My Computer” and select “Manage”) or with the “net” command on the command prompt.

Automated Test

► Application Note

TMS Runs on K1297 Tester, Work Directory is on the Same K1297 Tester

If the work directory is on the K1297 tester (see the section *Test Purpose – Mobile-Oriented Location Request* on page 5), selecting the work directory is straightforward and can be achieved by using the browser pane of the TMS “Choose Work Directory” window.

TMS Runs on a PC, Work Directory is on the K1297 Tester

As described in the section *Test Preparation*, on page 6, TMS can be run on a PC other than the K1297 tester, but the work directory could still be on the K1297 tester. Choosing the work directory becomes slightly more complex and depends on network connectivity of the local area networks. If both the PC and the K1297 tester are on the same subnet and the PC and the K1297 tester belong to the same domain and workgroup, the computer name can be chosen using UNC's. In case additional credentials, such as user names and passwords are required to establish network connections, TMS cannot establish the connection and appears to hang whenever the next steps are to be performed, e.g. selection of a test suite.

In this case, a network connection and mapping of the network drive should be performed using Windows or the “net” command in the command prompt before TMS is started. If the PC and the K1297 tester belong to different work groups or domains, the workgroup name and user name must be entered along with the password in order to establish a network connection and map the required network drive.

In cases where the K1297 tester is in a different subnet of the local area network, the IP address has to be used instead of the computer name when mapping network drives and assigning the work directory to TMS. Both the work directory path and the mapped network drive must use the IP address instead of the computer name.



► Figure 24. Mapping the shared folder as network drive

**TMS Runs on a PC,
 Work Directory is on the Same PC**

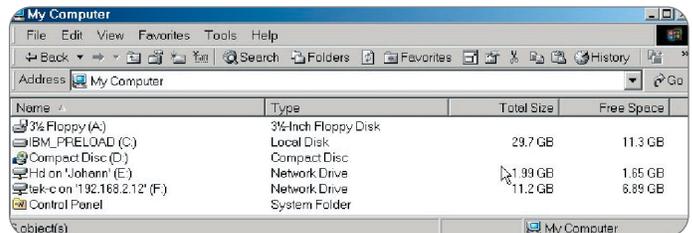
Assigning the work directory for TMS can be in the same manner as in sections *Deployment of the TMS System and the Work Directory*, on page 21, and *Test Purpose – Mobile-Oriented Location Request*, on page 5, however, the work directory must be accessible from the K1297 tester as well. The same naming conventions and user ID/password credentials as described in the section *TMS and Work Directory on K1297 Tester*, on page 21, apply. A network connection and mapping of the network drive must be performed from the K1297 tester to the PC. In this case, only UNC names or IP addresses should be used. Drive letters can only be used if the same drive letter is available on the PC and the K1297 tester.

TMS Runs on a PC, Work Directory is on a Different Server or Computer

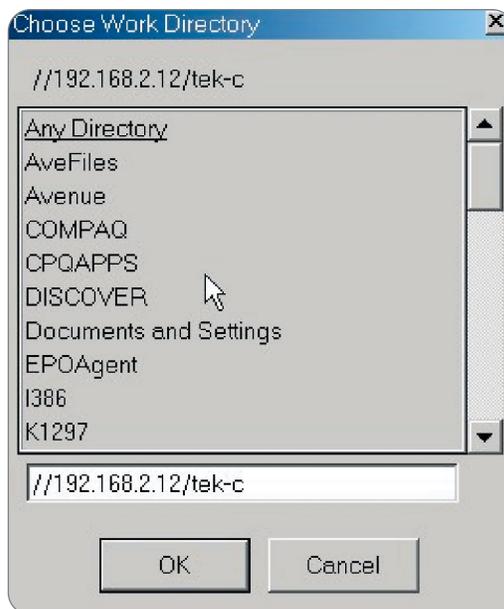
This configuration requires that network connections and mapping of network drives be established from the PC running the TMS and the K1297 tester. Only UNC names or IP addresses may be used for the path names. If the PC running TMS and the server/computer hosting the work directory belong to the same work group or domain, and the K1297 tester belongs to a different work group or domain, the same credentials must be used for connections from the PC as from the K1297 tester. Otherwise, the TMS may not be able to start the test suites execution through the DPE server on the K1297 tester.



▶ **Figure 25.** Tester belongs to workgroup/domain "TEKTRONIX"



▶ **Figure 26.** Using IP addresses to map network drives



▶ **Figure 27.** Assigning work directory in TMS using IP addresses

Contact Tektronix:

ASEAN / Australasia / Pakistan (65) 6356 3900

Austria +43 2236 8092 262

Belgium +32 (2) 715 89 70

Brazil & South America 55 (11) 3741-8360

Canada 1 (800) 661-5625

Central Europe & Greece +43 2236 8092 301

Denmark +45 44 850 700

Finland +358 (9) 4783 400

France & North Africa +33 (0) 1 69 86 80 34

Germany +49 (221) 94 77 400

Hong Kong (852) 2585-6688

India (91) 80-2275577

Italy +39 (02) 25086 1

Japan 81 (3) 3448-3010

Mexico, Central America & Caribbean 52 (55) 56666-333

The Netherlands +31 (0) 23 569 5555

Norway +47 22 07 07 00

People's Republic of China 86 (10) 6235 1230

Poland +48 (0) 22 521 53 40

Republic of Korea 82 (2) 528-5299

Russia, CIS & The Baltics +358 (9) 4783 400

South Africa +27 11 254 8360

Spain +34 (91) 372 6055

Sweden +46 8 477 6503/4

Taiwan 886 (2) 2722-9622

United Kingdom & Eire +44 (0) 1344 392400

USA 1 (800) 426-2200

USA (Export Sales) 1 (503) 627-1916

For other areas contact Tektronix, Inc. at: 1 (503) 627-7111

For Further Information

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology. Please visit www.tektronix.com



Copyright © 2003, Tektronix, Inc. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.

01/03 HMH

2FW-16265-0

