



# Software Product Description

---

**PRODUCT: Compaq TeMIP Graphical ASCII Toolkit V4.0 for Tru64 Unix SPD 64.82.04**

## DESCRIPTION

Compaq TeMIP for Tru64 UNIX® is a family of software products for the management of telecommunications and corporate networks, including fixed wire and mobile/cellular voice and data multi-vendor, multi-technology networks. TeMIP V4.0 provides comprehensive off-the-shelf fault management functions such as Alarm Handling, Event Logging and Trouble Ticketing for telecommunications network management.

TeMIP supports the International Standards Organization (ISO) management standards ISO 10164-x and 10165-x, and the NMF ensembles. TeMIP and its features are applicable in the context of the International Telephone Union-Telecommunication Standards (ITU-T) X.73x and Telecommunication Management Network (TMN) M.3010, M3100 Recommendations. It gives network operators a global view of their networks, and enables them to activate management functions and operations from single or multiple workstations.

TeMIP is built on top of the TeMIP Framework, and fully benefits from the object oriented and truly distributed software architecture.

The Graphical ASCII Toolkit V4.0 (GAT) is part of the TeMIP program and provides the TeMIP platform with access to telecommunication equipment such as switches, multiplexers or repeaters which, for historical reasons, do not support a standardized management interface. However, this equipment can report management information and receive control information using an interface designed for human interaction, by means of a character (ASCII) terminal or a printer. As such, the TeMIP Graphical ASCII Toolkit extends TeMIP functionality to all network elements (NEs) that support ASCII and TL1 protocols. It also benefits from the Event Filtering and Correlation capabilities of TeMIP. Refer to the Compaq TeMIP Fault Management Software Product Description (SPD 45.24.xx) and the Compaq TeMIP

SPD (54.17.xx) for more information about Distribution and Event Filtering and Correlation related features.

The TeMIP Graphical ASCII Toolkit is a collection of components that enable rapid development and easy maintenance of a TeMIP GAT Access Module for a given type of equipment. The main features are:

- Graphical User Interface (GUI)-driven codeless development toolkit with rule-based parsing
- Two-way stream management
- Multi-protocol support.

TeMIP GAT is a windows based configuration application that facilitates the development of a generic Access Module (AM) framework that can be adapted to the specifics of the management interfaces of a broad range of ASCII speaking network elements.

The TeMIP GAT GUI allows fast and easy development of AMs without coding. The result is a packaged AM (customized AM) that can be easily and seamlessly integrated into the TeMIP platform and as such benefits from TeMIP standard features (for example, security, distribution, low-level event filtering and correlation). It is also possible to extend and customize the toolkit.

Access Modules developed using TeMIP GAT are able to:

- Manage the connection with the Network Equipment
- Act as an agent for ASCII devices
- Support the terminal emulation and self-management capabilities of TeMIP.

A TeMIP GAT Access Module provides two distinct channels of communication between TeMIP and the Network Element:

- The Asynchronous Message Handler (AMH) is the asynchronous channel for unsolicited messages

- The Integrated Command Generator (ICG) is the synchronous bi-directional channel for NE control.

### Features

Depending on the scope of the customization, Access Modules developed using TeMIP GAT will be able to:

1. Manage the connection with the NE:
  - Communicate with NEs using three widely used standard protocols (TCP/IP, X25, RS232)
  - Raise communication alarms to the TeMIP platform if the communication facility fails
  - Detect and monitor "heartbeat" signals from an NE
  - Generate "keep alive" signals to maintain connections with NEs that do not support "heartbeat".
  - Automatically re-establish the connection with the NE when it is broken.
2. Act as an agent for ASCII devices:
  - Collect unsolicited ASCII messages from the NEs. Received messages are parsed and used to trigger internal actions (for example, forward OSI alarms, update attribute values)
  - Issue "polling" messages to retrieve alarm information from NEs that cannot emit unsolicited asynchronous messages
  - Correlate "clear" alarm messages with the original forwarded alarms
  - Issue "re-synchronization" messages to re-align a TeMIP view of the element with its actual state (for NEs that support it)
  - Log messages in a log file for optional use by off-line analysis tools
  - Maintain attribute values in a local database. The values can be modified by operations on the entities (for example, the Set directive) or autonomously by the Access Module itself, based on messages received from the NE
  - Synthesize state attributes for the NEs, based on received alarms (for example, maintain an Operational State attribute based on the messages received from the NE).
3. Support terminal emulation. "Passthrough" allows an operator to open a direct session on the NE, which emulates a dumb terminal connection.
4. Support the self-management capabilities of TeMIP. The Access Module itself is managed by TeMIP through a self-management interface,

which also allows the Access Module to report abnormal conditions.

### Access Module Components

After customization, a specific TeMIP GAT Access Module (Customized Access Module) will consist of the following components:

- Management Model (TeMIP Interface) - the Access Module interface to the TeMIP platform. It is based on the information model of the NE being managed and makes the management information available to TeMIP applications. The toolkit includes a Q.822/Q.823 information model, or Traffic Management model, so that Access Modules produced with GAT can work with a Traffic Management application.
- Communications Server (CS) (NE Interface) - the physical interface to the Network Element. All communication with the managed object passes through this interface. There is a choice of three Communications Servers, one for each of the supported protocols (TCP/IP, RS232, X25). The Communications Server is itself a Management Module dedicated to accessing Network Equipment. The Communications Server Access Modules supplied with the product are all based on the same framework. This framework can be customized to provide Communications Servers for additional protocols. An API is made public for this purpose.
- Asynchronous Message Handler (AMH) - composed of the Asynchronous Message Parser and the Action Executor:
  - The Asynchronous Message Parser receives raw ASCII messages from the Communications Server (NE Interface) and breaks them down using rule based analysis. The result is an Item Set that is a set of significant recognizable codes from the message (for example, Alarm Severity = Critical, State = Starting Up)
  - The Action Executor performs pre-defined actions. The actions are selected according to the Item Set received from the parser. Typical operations are: forward an OSI alarm to the TeMIP platform or update an attribute value in the Management Information Repository (MIR).
- Integrated Command Generator - handles command and response dialogues with the NE. It is composed of the Dialog Manager, the Command Builder, and the Response Parser:
  - The Dialog Manager handles command and response dialogs with the NE. A TeMIP directive is received by the Access Module in an encoded format. It processes the directive and passes it to the Command Builder. It also processes the resulting responses that are returned from the NE using the Response

Parser. A sequence of related commands and responses is called a Dialog.

- The Command Builder constructs the individual ASCII commands (in the language of the NE - MML commands) and passes them to the communication interface.
- The Response Parser is a rule based parser (similar to the Asynchronous Message Parser) that breaks down the responses to commands issued to the NE and feeds them back into the Dialog Manager.
- Local Management Information Repository - a local Access Module data repository that stores NE management data (for example, attribute values).

The run-time environment for a TeMIP GAT Access Module is made up of a combination of Customized AMs and Communications Server AMs. As for all TeMIP Management Modules, they can be distributed over several TeMIP systems depending on the degree of distribution needed.

### Communications Server

The Communications Server is a TeMIP management module that manages communication resources (represented by the TeMIP Communication Channel global class). It offers communication services for TeMIP client applications that need to communicate using TCP, X25, or RS232 protocols. A TeMIP GAT customized AM uses the Communications Server services.

The Communications Server has the following characteristics:

- It can be installed on a different system from its clients (customized AMs, Passthrough Applications, and so on)
- Multiple CSs can be installed in a given environment (system)
- The same CS AM is able to handle connections from more than one customized AM to any number of NEs that use the same protocol
- Each CS is specialized for a single protocol
- The CS monitors the state of its Communication Channel (CC)
- When activated for an NE port, raw message logging is performed by the CS for this port
- When communicating with an NE or a mediator, one (or more) ports might be used. Usually the NE requires one CC (physical connection, X25+ virtual circuit or TCP socket) for asynchronous event reporting and a different CC for commands/responses (synchronous messages). In some cases only one channel may be used for both events and commands/responses
- Event buffering capabilities and collection management are offered at the CS level to cope with differences between the event producers

(NEs) rate and event consumers (management application) rate

- The CS is responsible for compacting raw data received from an NE into bounded messages, whenever possible, before processing it further (that is, before logging, buffering or transmitting to a consumer). To this effect, it has to identify:
  - Either Start message and End message patterns in the data received
  - Or only Start message patterns when the end of one message is recognized by the next incoming message.
- The CS can easily be managed and configured using TeMIP standard procedures (that is, activation of directives on appropriate objects).

### Access Module Customization

The syntax and semantics of the messages, commands, and responses that are exchanged between the NE and the management terminal, are called the MML (Man Machine Language) specification. MML messages are specific to each type of equipment and are an equipment vendor's property. The connection between network elements and management terminals is either a direct line with an RS232 interface or a connection through X25+ or TCP/IP networks.

The customization process includes the following steps:

1. Analyze the specific MML of the equipment to be managed
2. Define the management model for the specific equipment (that is, commands supported by the AM, messages to be mapped on to TeMIP alarms, events, and notifications)
3. Produce a specific parser and mapping table
4. Build and test the customized ASCII Access Module (run-time) either locally or by generating an Access Module kit for installation on another TeMIP system
5. Finalize specific end user documentation, describing how the AM connects to the NE.

The entire customization (development of a GAT Access Module) can be done using the TeMIP GAT GUI. Each of the AM components has its own customization window, which is accessed from the Main Customization Window. When all the components have been configured and tested, the run-time AM is generated. It is possible to import and re-use previous customization sources.

### Main Customization Window

This window is the starting point for the development of a GAT Access Module. It is a graphical representation of the AM components (Component Frame). By simply clicking on a component, the user

calls the component specific editor window. From the Main Customization Window, all other options for generating the AM are available from the Menu bar (that is, Compile, Test, Generate, ...). General characteristics (Name, Version, description, ...) of the current customization are displayed in the Detail Frame. All messages reporting the progress of the current activity (for example, editing or compilation) are displayed in the Output Frame using a color-coded format. Detailed error reports are routed to a log file.

All Component windows consist of a toolbar, a main editor frame, composed of multiple columns, and a bottom status bar. A set of optional keyboard accelerators are also available. For a detailed description of window behavior, refer to the *Compaq TeMIP Graphical ASCII Toolkit Customization Manual*.

### Management Model

TeMIP architecture is based on the notion of a dynamic, extensible network model. In TeMIP, every manageable entity in the network (that is, NE) can be described in terms of its own unique management model defined as:

- Classes representing the entity (Class Hierarchy)
- Attributes used to store and reference management information relative to the entity (Attribute Partitions)
- Operations representing management actions (Directives)
- Notifications describing unsolicited information generated by the entity (Event Partition).

The Management Model Editor window allows for the development of management models for entities managed by a GAT Access Module. It consists of up to five columns:

*Class Hierarchy Column* - (that is, the naming tree) displays the Class Hierarchy that consists of one or more Global Classes and a framework of subordinate classes

*Partitions Column* - displays the Attribute Partitions, the Event Partitions, and the Directives for the selected class

*Details Column* - provides details of the Partitions or Directives selected in the Partitions column.

### Command Builder

The command builder is part of the Synchronous Channel, which is the bi-directional channel used for NE control. It is the AM component that builds individual commands in the language of the NE.

The Command Builder Editor window allows for the building of individual commands in the language of the NE (typically, the command definitions are extracted from the equipment documentation). It consists of four sections:

*Command Column* - contains a simple list of all the NE commands that are to be supported. The remaining sections of the window display the details of the selected command.

*Command Component Column* - each command consists of an ordered list of components. The component fields are "syntax guided", which simplifies the development of components and provides continuous guidance and verification of input. The actual command sent on-line consists of a concatenation of the command components, where the variables (parameters) are replaced by values at run-time.

*Input Parameter List* - a scroll list of all the possible input parameters for the selected command.

*Output Parameter List* - a scroll list of all the expected output parameters for the selected command. The output parameters entered here are visible as Items in a pick-list when developing Statements in the Response Parser.

### Dialog Manager

The Dialog Manager is part of the Synchronous Channel. It is the AM component that handles the command/response dialogs between TeMIP and the NE.

The Dialog Manager Editor window is used to define the bindings between the eligible directives and the NE commands, which are maintained by the Command Builder. All the new Directives, defined in the Management Model, and the commands defined in the Command Builder, together with their input and output parameters are visible in the Dialog Manager editor. It consists of five main sections:

*Class Hierarchy Column* - displays the information model Class Hierarchy.

*Directive Column* - a list of all the Directives that are eligible for binding to NE commands.

*Command Column* - a list of the NE commands that have been mapped onto the Directive that is currently selected in the Directive Column.

*Input Parameter Binding List* - a scroll list of all the Input Parameters for the Command selected in the Command Column. This section is used to bind the Command Input Parameters to the Management Model attributes or Directive Request arguments.

*Output Parameters Binding List* - a scroll list of all the expected Output Parameters for the command selected in the Command Column.

### Message Parser

The Asynchronous Message Parser and the Response Parser have the same editor window and have the same customization possibilities. The two parsers can be combined, that is, the same parser definition can be used for both components. Message parsing is rule based. Messages from the NE, stored

in the message Items, are processed according to a hierarchy of rules. For a detailed description of message parsing concepts, refer to the *Compaq TeMIP Graphical ASCII Toolkit Customization Manual*.

The Asynchronous Message Parser handles asynchronous messages, typically alarms and status messages, from the NE. It processes incoming messages and determines which Actions are to be executed. The Action List is maintained by the Action Executor component.

The Response Parser handles synchronous messages. These are messages that are generated in response to commands from the Command Builder. The Response Parser editor shares information with the Command Builder so that command output parameters are visible as Items in the Response Parser.

The Parser Editor window consists of four main sections:

*Rule Hierarchy Column* - contains a graphical representation of the rule relationships and determines the order in which the rules are applied.

*Rule Expression Column* - there may be one or more Boolean expressions that test the message. The rule only fires if all the expressions are satisfied, and then all the related Statements are executed.

*Statement List Column* - statements in the Statement List are executed in turn from top to bottom, if the rule fires. Each statement consists of a Guard and an Operation.

*Next Rule List* - is a list of dependent rules that are attached to the current rule. The rule hierarchy is constructed by adding rules to this list.

### Action Executor

The Action Executor is part of the AM Asynchronous Channel, which handles unsolicited messages from the Network Element. The Action Executor editor is used to develop Actions that consist of a number of Statements. The Asynchronous Message Parser processes incoming messages and determines which of the pre-defined Actions are to be executed.

The Action Executor Editor window consists of three main sections:

*Class Hierarchy Column* - contains the information model Class Hierarchy

*Action Column* - displays a list of all the actions in the selected Class or Default Action List that is selected in the previous column

*Statement List Column* - statements in this list are executed in turn from top to bottom when the Action is executed.

### Passthrough

When an alarm is reported by a network element, it is critical that an operator has access to the NE. The Passthrough component provides operators with access from their local workstations to the management terminal interface of the remote NE or the Operation and Maintenance Center. The Passthrough feature allows an operator to open a direct session on the NE which emulates a dumb terminal connection to the equipment.

Passthrough is a TeMIP launched application that is invoked from the TeMIP Iconic Map PM. A Passthrough session interacts with an NE by means of a Communications Server.

### Samples

To help new users to build their first TeMIP GAT Access Module, two customized AM samples are provided:

- LAB\_AM - a simple, but realistic and complete customization example
- TL1\_AM - this customization example is expected to be used as a starting point to speed up the development of real AMs for interfacing TL1 based equipment. It provides a customized AM example for a list of messages corresponding with a fictitious NE that supports a TL1 Management Interface.

**Note:** These sample AMs are not products and cannot be used in a production environment.

### Openness

#### Extension Mechanism

The Extension Mechanism allows the addition of new functions to the GUI. It also allows additional code to be included in the generated Customized AM. This is done either through initialization files containing information to allow the extension to be included in the relevant parts of the toolkit and/or run-time AM, or by using a number of hooks defined in the toolkit code and in the generated AM code.

#### Custom Object Model Implementation

It is possible to adapt standard, off-the-shelf, object model definition and implementation to project specific requirements.

### Documentation

For additional information, refer to the appropriate documentation:

- *Compaq TeMIP Graphical ASCII Toolkit Overview*
- *Compaq TeMIP Graphical ASCII Toolkit Installation Guide*
- *Compaq TeMIP Graphical ASCII Toolkit Customization Manual*

- *Compaq TeMIP Graphical ASCII Toolkit Configuration and Troubleshooting Guide*
- *Compaq TeMIP Graphical ASCII Toolkit, Advanced Customization Development Guide*
- *Compaq TeMIP Graphical ASCII Toolkit, Advanced Customization Reference Guide*

## HARDWARE REQUIREMENTS

### *Tier 1:*

Compaq AlphaServer DS10, DS20

### *Tier 2:*

Compaq AlphaServer ES40

### *Tier 3:*

Compaq AlphaServer GS60, GS140

## Disk Space Requirements

For installation: 100,000 KB

For use (permanent): 100,000 KB

Additional disk space may be required for Customized Access Modules.

Run-time systems will only require sufficient disk space to load customized and CS AM executables.

## Memory Requirements

### *For run-time systems:*

The minimum memory supported, due to a TeMIP Framework prerequisite, is 128 MB.

### *For development systems:*

The minimum memory supported, due to a TeMIP Framework prerequisite, is 256 MB.

However, the use of this software in conjunction with increased memory improves performance.

## SOFTWARE REQUIREMENTS

### *For run-time systems:*

- Compaq Tru64 UNIX V4.0F
- TeMIP Framework V4.0

*For development systems*, the following software must be installed on top of the above:

- Visual TeMIP Developer's Toolkit V4.0
- DEC C++ V6.1 compiler

## OPTIONAL SOFTWARE

- DEC X.25 V3.0 for Compaq Tru64 UNIX.

## GROWTH CONSIDERATIONS

The minimum hardware/software requirements for any future version of this product could be different from the requirements for the current version.

## YEAR 2000 READY

This product is Year 2000 Ready.

"Year 2000 Ready" products are defined by Compaq as products capable of accurately processing, providing, and/or receiving date data from, into and between the twentieth and the twenty-first centuries, and the years 1999 and 2000, including leap year calculations, when used in accordance with the associated Compaq product documentation and provided that all hardware, firmware and software used in combination with such Compaq products properly exchange accurate date data with the Compaq products.

For additional information visit Compaq's Year 2000 Product Readiness web site located at <http://www.compaq.com/year2000>.

To ensure that this product is Year 2000 Ready, code assessment and system tests to verify the transition between December 31<sup>st</sup> 1999 and January 1<sup>st</sup> 2000 were utilized.

To ensure that this product interoperates properly with other hardware and software, the system tests involving Compaq's TeMIP V4.0 are applicable, as this product was verified as being Year 2000 Ready.

## DISTRIBUTION MEDIA

This product is also available as part of the UNIX Consolidated Software distribution on CD-ROM. See ordering information for each Software Media reference.

## ORDERING INFORMATION

### *TeMIP Graphical ASCII Toolkit Development:*

Software License: QM-5SLAA-AA  
(This replaces the license QL-5SLA9-AA)

Software Media: QA-6HPAA-H8

Software Documentation: QA-5SLAA-GZ

Software Product Services:  
QT-5SL\*\*-\*\* or QR-\*\*5SL-\*\*-\*\*

### *TeMIP Graphical ASCII Toolkit Run-Time:*

Software License: QM-5SMAA-AA  
(This replaces the license QL-5SMAM-3B)

Software Product Services:  
QT-5SM\*\*-\*\* or QR-\*\*5SM-\*\*-\*\*

\* denotes a variable value. Contact your local Compaq office for more information.

## SOFTWARE LICENSING

This software is furnished under the licensing provisions of Compaq Computer Corporation's Shrinkwrap License Terms and Conditions. For more information about Compaq's licensing terms and policies, contact your local Compaq office.

License units for TeMIP Graphical ASCII Toolkit *Development* are allocated on an Unlimited System Use basis, independently of the machine tier on which they are running.

License units for TeMIP Graphical ASCII Toolkit *Run-Time* are allocated on a Concurrent Use basis, independently of the machine tier on which they are running. "Concurrent Use" is defined as the number of instances of Access Modules running at the same time on a single machine.

TeMIP Graphical ASCII Toolkit uses the FLEXIm Software License Key system.

A FLEXIm key must be obtained using information provided with the license deliverable. An Authorization ID is provided for each license, which allows the user to generate a license key from the Compaq License Key Fulfillment Website according to instructions provided with the license agreement.

You may obtain a Product Authorization Key by providing your Authorization ID (from the current License Agreement) and the QM partnumber identified under "ORDERING INFORMATION". Please provide this information to your geography contact identified on the License Agreement. The License Agreement for the current version extends the right to utilize prior versions.

## SOFTWARE PRODUCT SERVICES

A variety of service options are available from Compaq. For more information, contact your local Compaq office.

## SOFTWARE WARRANTY

This software is provided by Compaq with a 90 day conformance warranty in accordance with the Compaq warranty terms applicable to the license purchase.

The above information is valid at the time of release. Please contact your local Compaq office for the most up-to-date information.

- ® COMPAQ, the Compaq logo, and the Digital Logo are registered in U.S. Patent and Trademark Office.
- ® Windows NT is a registered trademark of Microsoft Corporation.
- ® FLEXIm is a registered trademark of GLOBEtrouter Software, Inc.
- ® UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Ltd.
- ™ Tru64 UNIX, TeMIP, DEC, AlphaStation, AlphaServer, and TruCluster are trademarks of Compaq Computer Corporation.

Other product names mentioned herein may be the trademarks of their respective companies.

**©2000 Compaq Computer Corporation  
All Rights Reserved**