I'm not robot

CANoe - Guide Me! Easy Ball Play CANoe offers an impressive array of features, making it challenging to grasp at first. This guide provides simple yet effective tips on: Common pitfalls to avoid The ideal setup for your CANoe tool environment When developing ECU, you often need signals from other ECUs in the network. CANoe provides this through its remaining bus simulation feature, allowing both real and simulated data to be analyzed using the Trace and Graphics window and stored via CANoe's logging function. You can stimulate the network or manipulate the remaining bus simulation as needed. To avoid potential pitfalls: - Handwritten code for the remaining bus simulation is error-prone and maintenance-intensive. - Changes in the communication matrix lead to inconsistencies in the remaining bus simulation. Setting Up Your Vector Solution: Use the appropriate OEM-specific Interaction Layer to configure CANoe to automatically send simulated messages according to their timing definitions in the database. You develop ECUs and want to test their functions using automated testing. In doing so, you push your system to its limits by stimulating ECU functions, injecting errors selectively, and monitoring ECU reactions on your network. To avoid common pitfalls: - Tests performed interactively and manually are not reproducible. - Non-automated test execution ties up scarce resources or important staff. - Manually generated test protocols may be incomplete or difficult to process. Setting Up Your Vector Solution: Create your tests with vTESTstudio. With this tool, you can either select predefined test primitives or develop your own. CANoe performs as a tool for automatic test execution and generates an extensive test protocol. The tests can be reproduced at any time and test reports can be further processed automatically. You want to assure the complete system functionality. It's necessary here to provide the corresponding system environment for your System Under Test (SUT) and incorporate analog and digital inputs and outputs in the test. To avoid common pitfalls: - Missing time synchronization between network events and stimulation or measurement of inputs and outputs. - Specific signal conditioning of the SUT inputs and outputs is not only time-consuming but can also be error-prone. Setting Up Your Vector Solution: Make use of CANoe's openness and integrate it into a group of various run-time environments as part of co-simulation. The VT System ensures that events for stimulating inputs and outputs of the SUT, as well as network events, are synchronized. The VT System modules are specifically developed for ECU test - a special signal conditioning is therefore not necessary. Play the Easy Ball with CANoe! Talk to us personally about CANoe ... or simply write us an email. We look forward to your request! CANoe and CANalyzer as Diagnostic Tools Version 1.6, 2020-09-03 Application Note AN-IND-1-001 This application gives an introduction into working with diagnostics in CANoe/CANalyzer. It presents the basic technical aspects and possibilities with the Diagnostic Feature Set, complements the help file of CANoe/CANalyzer and may be used as a reference. A comprehensive guide to diagnostic features in various tools, covering both theoretical and practical aspects. The introduction sets the stage for understanding how different systems approach diagnostics, including the built-in channel versus third-party DLLs and CAPL callback interfaces. The next section delves into specific tools like CANoe and CANalyzer, exploring their support for transport protocols and diagnostic descriptions. It highlights various formats such as CDD, ODX, MDX, UDS, KWP, and Standard Diagnostic Description, each serving different purposes in the context of diagnostics. Further sections provide detailed explanations of additional features within these tools, including trace windows, fault memory, variant coding, session control, OBD-II, ECU simulations, test modules using CAPL (available only in CANoe), test units (also exclusive to CANoe), symbol explorers for diagnostic objects and parameters, autocomplete input assistance for diagnostics, and functional group requests. Throughout this guide, the reader gains a thorough understanding of how these tools can be used effectively for diagnostics. 1.0 Diagnostic Tools Features via COM 2.0 Basic Diagnostic Editor 3.0 Security Access handling 4.0 First steps 4.1 Usage of Diagnostic Descriptions 4.1.1 Adding a Diagnostic Description 4.1.2 Configuring the Diagnostic Description 5.0 Usage of Diagnostic Console, Session Control and Fault Memory window 5.1 Sending a diagnostic request and receiving a response 5.1.2 Reading fault memory 5.1.3 Functional Group Requests 6.0 CANoe and CANalyzer as Diagnostic Tools 7.0 Contact Information: www.vector.com or +49-711-80 670-0 8.0 Using CAPL for Diagnostics 9.0 Common techniques for Simulation and Tester 10.0 ECU diagnostics simulation 4.2.7 Debug Level 4.2.8 Adding a Diagnostics Request Event Handler 4.2.9 Creating a Diagnostic Response CANoe/CANalyzer as Diagnostic Tester 4.3.1 Setting the Diagnostic Target 4.3.2 Creating a Diagnostic Request 4.3.3 Adding a Diagnostics Response Event Handler 4.3.4 Negative Response Handling 4.4 Combining Test and Diagnostic Features 4.4.1 Timeout Handling 4.4.2 Automated Diagnostic Tests with CANoe 5.0 Advanced Examples 5.1 ECU Simulation of "Response Pending" 5.2 Modifying Diagnostic Object Length 5.3 Filling Diagnostic Content 5.4 Fault Injection 5.4.1 Making Request Length Illegal 5.4.2 Introducing Errors on Transport Protocol Level 5.5 Accessing a Node via Gateway Simulation 6.0 Common Mistakes 7.0 Abbreviations 8.0 References 9.0 Additional Resources 10.0 Contacts GmbH 3 Contact Information: www.vector.com or +49-711-80 670-0 Overview: This document provides an introduction to working with diagnostics in Vector's tools CANoe and CANalyzer. It covers the basic technical aspects and possibilities of the Diagnostic Feature Set, including examples to help test engineers get started with testing diagnostics. Direct Access to ECU Fault Memory Overview The Diagnostic Descriptions assigned to networks provide access to the Fault Memory window, which can be viewed via the "View" menu. Diagnostics Session Control Window Overview A corresponding Diagnostic Session Control window is made available when a Diagnostic Description is assigned, accessible via the "View" menu. CANoe and CANalyzer as Diagnostic Tools, Copyright © 2020 - Vector Informatik GmbH. The tools support various automotive protocols, including ISO TP, KWP2000, Diagnostics Observer, and more. For most protocols, a node layer Transport Protocol (TP) DLL exists in CANoe. The TP observer interprets messages according to the protocol and displays results in the Trace window. Diagnostic Descriptions are necessary for working on diagnostics layers. Simple descriptions can be defined using the Basic Diagnostic Editor. More advanced descriptions require CDD, ODX (PDX), or MDX files. These file types can be mixed within one configuration. Data Exchange (ODX) files carry diagnostic data, which can be divided into multiple ODX files and stored in PDX files. To ensure sufficient information for diagnostic testers, Vector recommends using PDX files containing all relevant single ODX files. The usage of PDX files is similar to that of CDD files. Multiplex Diagnostic Data Exchange (MDX) files carry OEM-specific diagnostic data and are used similarly to ODX archive files. Basic Diagnostic Descriptions, created with CANoe/CANalyzer's Basic Diagnostic Editor, can be customized by users and stored as part of the configuration. They have limited functionality compared to other formats, but can still describe simple diagnostic services (UDS & KWP) and send/receive requests/responses on various bus systems. Vector supports symbolic interpretation of these basic diagnostics in the Trace window and allows them to be used as "additional descriptions." For simple applications, Basic Diagnostics extends the process-oriented approach with CANdela Diagnostic Descriptions. To use Basic Diagnostics, users must add a "Basic Diagnostic Description" (KWP or UDS) to their CANoe/CANalyzer configuration. They can define/modify diagnostic services while measurement is stopped and commit them to the Diagnostic Console. The company also offers Standard Diagnostic Descriptions based on ISO standards "Unified Diagnostic Services" (UDS, ISO 14229) or "Keyword Protocol 2000" (KWP2000, ISO 14230). These descriptions do not contain OEM-specific services and cannot be customized. Standard Diagnostic Descriptions (CDDs) provide advantages over generic ones, including: - Applying mechanisms for concrete diagnostics to standard CDDs with restrictions - Configuring communication parameters in the dialog - Accessing ECUs through the Diagnostic Console - Parameterizing interpretation of transmitted data These features can be used after adding a diagnostic description to the CANoe/CANalyzer configuration. Diagnostic Tools and Features of Vector Informatik GmbH ---------- To access advanced diagnostic features in the Symbol Explorer, navigate to the "Add condition... | Add Diagnostic Service..." menu. Here, you can modify existing service conditions to filter only requests or responses. Moreover, utilizing column filters in the trace window enables viewing specific patterns. The Vector Diagnostic Feature Set comprises functions necessary for ECU development, testing, and application diagnostics. The Diagnostic Console provides interactive access to diagnostic services based on the Diagnostic Description. You can select, parameterize, and display dedicated responses. The Fault Memory Console offers quick access to an ECU's fault memory, while the Session Control window allows changing active sessions with a configured Security DLL for restricted security levels. The OBD-II window facilitates On-Board Diagnostics according to the SAE J1979 standard. This feature is also part of Vector products CANape MC+D and CANdito, supporting identical development processes. In addition to a security DLL linked to the corresponding diagnostic description, it's feasible to switch session states without worrying about key exchange and computation. By using the Diagnostic Session Control window, users can execute limited diagnostic services from the Diagnostic Console that are only accessible during protected sessions with a specific security level. Figure 8: Diagnostic Session Control (left) and OBD-II window (right) 2.5.5 OBD-II Window OBD stands for On-Board Diagnostics in automotive contexts, referring to self-diagnostic capabilities of vehicles. OBD systems provide vehicle owners or repair technicians with health information for various sub-systems through a standardized digital port and diagnostic trouble codes (DTCs). After configuring the network settings, the OBD-II window opens along with Diagnostic Console and Fault Memory windows. Using the OBD-II window, users can initiate manual scans to receive responses from supported ECUs. Based on these responses, the module calculates available ECUs and requests supported by them, updating relevant pages such as System Status, Live Data Grid, Vehicle Info, and On-Board Test Results. Diagnostic Console and Fault Memory windows also allow sending requests and reading OBD-II-related fault memory. 2.5.6 CAPL-Based ECU Simulations CAPL can simulate ECUs, gateways, or diagnostic testers even without real counterparts. Diagnostic commands in CAPL grant access to diagnostics services and data using symbolic names from the Diagnostic Description. Simulations react to incoming requests and responses, implementing interactive tester capabilities