OTX - Standardized Language for Diagnostic and Test Sequences

**Why OTX?**

OTX offers support for a three-stage development of test sequences:

- Specification stage: The overall sequence logic is developed at a descriptive, linguistic level. This is the earliest stage of test sequence development process, where most details for creating an executable test sequence are not known.

- Intermediate stage: The test sequence contains parts that are already executable; other parts are still at the specification stage. At any step in this process, the sequence can be saved and exchanged in a valid format. A runtime interpreter can execute the implemented parts.

- Realization stage: There are no more specification-only parts in the sequence. The OTX sequence is now fully executable. It represents an "executable specification".

**OTX Standard ISO 13209**

Overview

Part 1: OTX Core with and additional-standard defined. Application-specific features are contained in the Part 3 of ISO 13209. Furthermore, the list of extensions comprises controlling measurement equipment, internationalization, working with physical units, accessing the environment via external DLLs or EXEs, further major interface (API) elements and other utilities.

Part 2: Extends the OTX Core with several additional-standard defined. Application-specific features are contained in the Part 3 of ISO 13209. Furthermore, the list of extensions comprises controlling measurement equipment, internationalization, working with physical units, accessing the environment via external DLLs or EXEs, further major interface (API) elements and other utilities.

Part 3: OTX Extensions

- 4 Extensions based on Part 3 of the ISO Standard
  - otxIFD_Logging.xsd: Provides access to system time
  - otxIFD_Flash.xsd: Supports of Flash devices
  - otxIFD_Math.xsd: Extended functionality for math operations
  - otxIFD_i18n.xsd: Internationalization features, multi-language support and translation mechanism

**How do I use OTX?**

The OTX format offers support for a three-stage development of test sequences:

- Specification stage: The overall sequence logic is developed at a descriptive, linguistic level. This is the earliest stage of test sequence development process, where most details for creating an executable test sequence are not known.

- Intermediate stage: The test sequence contains parts that are already executable; other parts are still at the specification stage. At any step in this process, the sequence can be saved and exchanged in a valid format. A runtime interpreter can execute the implemented parts.

- Realization stage: There are no more specification-only parts in the sequence. The OTX sequence is now fully executable. It represents an "executable specification".

**Diagnostic System Overview with OTX**

- OTX Editor
- OTX Interface
- OTX Libraries
- Text Applications
- Base System with MODI Kernel / O-Runner
- Interface for Applications (IF)
- Interface for Bus Systems
- OSI 1.1
- OSI 1.2

**Where is OTX applied?**

- Development
- Test
- Production
- Service

**What is OTX?**

- Standard format: The OTX standard ISO 13209 provides an open and standardized format for the function- and machine-readable description of diagnostic test sequences.

- Diagnostic test sequences: Sequences for diagnostic testing are utilized wherever automotive components or functions with diagnostic abilities are being diagnosed, tested, reprogrammed or related by offline test equipment.

- Security: The OTX standard provides a standard for secured communication between the diagnostic tool and the vehicle's ECU.

**Signature Concept in OTX**

A signature describes an interface to a procedure. It is a procedure without a realization. Similar to nodes, procedures in an OTX sequence can be bound to a validity. Instead of a procedure, the programmer calls the signature in an OTX sequence.

**Advantages of OTX**

- The specification and implementation of test sequences are combined in one OTX document.
- Easy maintainability of OTX sequences thanks to the separation of user interface, sequence model and data.
- Simpler integration of extensive OTX libraries for reusable sequences.
- Debugging possibilities support error search during creating OTX sequences.
- Supports of human- and machine-readable filing format for test sequences (XML).

- OTX can be used outside of vehicle diagnostics; other advice can be incorporated (testing DLLs or API).
- OTX sequences execute the creation process, enabling the use of even minor improvement potentials, which can nevertheless represent considerable cost savings.

- The use of an ISO standard ensures a greater selection of available software tools (manufacturer independence).

- Long-term availability of validated diagnostic sequences and the securing of diagnostic know-how.

**Who needs OTX?**

- Development
- Test
- Production
- Service

- Specifying Editor
- System Development
- System Test
- Road Test
- After Sales

**Validity Concept in OTX**

A validity is always true or false:

- It can be a context variable of type Boolean, a composed logical expression resulting in true or false of e.g. several context variables.
- Nodes in an OTX sequence can be bound to a validity and therefore contain several realizations.

**Context Concept in OTX**

Context information is treated like static, unchangeable data during runtime.

- Executable steps in an OTX sequence are combined in one OTX document.

- Diagnostic test sequences require access to contextual information e.g. to vehicle (e.g. engine type and variant) or application (e.g. manufacturing, engineering, etc.)

**OTX in Application**

- Diagnostic System Overview with OTX
- Diagnostic - Use Cases for OTX

**Order additional copies:** info.automotive@softing.com / www.automotive.softing.com

© 2016 Softing Automotive Electronics GmbH. Despite all due care and attention, Softing accepts no liability and extends no guarantee for the correctness, completeness or currentness of the information.