



XML System Configuration Specification

1 Introduction

Note: Nothing in this standard supercedes applicable laws and regulations.

Note: In the event of conflict between the English and domestic language, the English language shall take precedence.

1.1 Purpose. This standard specifies the Extensible Markup Language (XML) configuration method including definition of the XML schema. The XML configuration method is used to configure Electronic Control Units (ECU) for different vehicle variants and to minimize proliferation of calibration files.

1.2 Applicability. This standard serves as a single point of reference for the Global A XML configuration method including the XML schema.

1.3 Remarks. Not applicable.

2 References

Note: Only the latest approved standards are applicable unless otherwise specified.

2.1 External Standards/Specifications.

ISO 9141	ISO 14230	SAE J2284
ISO 14229	SAE J1850	

2.2 GM Standards/Specifications.

GMW3089	GMW3110	GMW3122
---------	---------	---------

2.3 Additional References.

- ECU Variant Coding Editor (EVE) User Manual
- GGSE ECU Diagnostic Infrastructure Requirements
- GM2190, GM Enhanced Diagnostic Specification
- GM J1962
- Subsystem Technical Specifications
- Technical specification(s), including wiring diagrams
- XML Configuration SharePoint site: <https://share.gm.com/sites/XMLCONF/HOME2.aspx>
- XML Cross-balancing User Manual
- XML Tool Chain training material

3 General (Informative)

This section gives an overview about system configuration related activities.

3.1 System Configuration versus Calibration. It is important to differentiate between system configuration and calibration. Therefore, the appropriate definitions are given here that apply for this specification.

3.1.1 Definitions.

3.1.1.1 Configuration Parameter. A configuration parameter is a parameter that has the purpose of enabling or disabling a feature, or for selecting one choice out of several alternatives.

Example: Language settings for the infotainment subsystem.

3.1.1.2 Performance Parameter. A performance parameter is a parameter that is used for performance tuning. Examples: Engine characteristic map, dimming curves.

3.1.1.3 XML Configuration File. An XML configuration file is a file that contains configuration parameters (see 3.1.1.1) and that conforms to the XML schema definition described in Appendix A, Figures A1 thru A12 and Tables A1 thru A12. XML configuration files are used for system configuration (see 4.1.1).

3.1.1.4 Calibration File. A calibration file is a file that contains performance parameters (see 3.1.1.2).

Note: The term Proliferated calibration files is used to describe the situation where tuning data is separated into multiple files by feature or function (e.g., multiple Uniform Part Classification/Functional Name Address (UPC)/(FNA) combinations) or where multiple files may exist for a given feature/function when option content results in differences in the tuning value (e.g., multiple iterations of files with the same UPC/FNA).

3.1.1.5 Utility File. A utility file is a single data file that contains the instructions needed for programming tools (e.g., service and manufacturing tools) to program or reprogram an ECU.

3.1.1.6 Soft Part. A soft part is a data file that is assigned a GM part number and that is controlled via standard part management processes and systems (maybe with special provisions to accommodate the unique characteristics of software).

Examples: Operational software, calibration files (see 3.1.1.4), XML configuration files (see 3.1.1.3), and utility files (see 3.1.1.5).

3.1.1.7 Calibration Verification Number (CVN). A calibration verification number is used to verify the integrity of ECU software or calibration files.

3.2 System Configuration Process. The following (3.2) describes the system configuration workflows for Product Engineering, for Manufacturing Engineering and for Service and Aftersales.

3.2.1 Process Steps in Product Engineering. Figure F1, Appendix F depicts the activities that specify, verify, and manage system configurations during the product development lifecycle. Details of these activities are described in the sequence architectural level, vehicle level, subsystem level and component level task lists.

3.2.1.1 Architectural Level Task List. The following system configuration related activities are performed on electrical architecture level.

3.2.1.1.1 “Specify System Configuration Strategy and Methods”. A strategy for performing system configuration is defined. Suitable methods are selected. The deliverable of this task is a standard for system configuration.

3.2.1.1.2 “Physical Function Allocation (PFA)”. System features and functions are allocated to ECUs. Inputs to the physical allocation step are the different Subsystem Technical Specifications (SSTS) as well as the Standard and Options (S&O) lists of affected vehicle programs. Results of this allocation step are captured in the electrical architecture SSTS and in the PFA model. The Vehicle Description System (VDS) of affected vehicle programs is enhanced. The results of this allocation step also influence system configurations and shall therefore be captured in the specific SSTSs in a separate appendix “Variants” (see Appendix E).

3.2.1.1.3 “Plan Functional Growth”. Functional growth for an electrical architecture is planned based on the architectural SSTS and on the requirements of affected vehicle programs. The deliverable is a functional growth plan (e.g., the software timing plan for the Global A electrical architecture).

3.2.1.2 Vehicle Level Task List. The following system configuration related activities are performed for individual vehicle programs.

3.2.1.2.1 “Specify Standard and Options (S&O) list and develop VDS”. Based on Marketing segment specifications and Product Program Content (PPC), the VDS and the S&O list are defined for an individual vehicle program. Complexity management is applied in this process step to reduce complexity by limiting the number of different variants offered to the customer.

3.2.1.2.2 “Link Option Content to Performance”. VDS, Regular Production Options (RPO) and S&O list are used to extract performance parameters (see 3.1.1.2). Deliverables of this task are an overview list of the performance parameters and requirements for the experimental BOM.

3.2.1.2.3 “Develop Performance Calibration”. For the performance parameters that have been identified in the previous task (see 3.2.1.2.2) the necessary calibrations shall be developed. The deliverable of this process step is a set of calibration files for each different vehicle variant.

3.2.1.2.4 “Plan Vehicle Software Configuration”. Based on the functional growth plan (e.g., the Global A software timing plan, see 3.2.1.1.3), each individual vehicle program plans its specific timing and captures this in the software implementation plan.

3.2.1.2.5 “Verify Consistency of Configuration”. A consistency check shall be performed to verify that the released data files conform to the VDS and RPOs of the individual vehicle program.

3.2.1.3 Subsystem Level Task List. The following system configuration related activities shall be performed on subsystem level.

3.2.1.3.1 “Initiate Variant Management on Subsystem level”. The complete lists of performance parameters and configuration parameters are determined for the subsystem and shall be documented in the SSTS in an appendix “Variants” (see Appendix E). VDS and S&O list shall be considered for applicable vehicle programs as well as the overview list of performance parameters (see 3.2.1.2.2).

3.2.1.3.2 “Develop Feature Configuration”. Based on the documented list of configuration parameters (see 3.2.1.3.1) the RPOs for individual vehicle variants are defined. Additional deliverables of this task are the configuration-based test strategy, test procedures and test input data. The test strategy shall specify the coverage of system configurations in the bench tests. Usually a reasonable subset of the possible system configurations shall be selected as test objects. The test strategy shall state this selection and its criteria. Test input data for the test procedures need to be specified according to the selected system configurations.

3.2.1.3.3 “Specify Electrical Bench Content”. Deliverables of this task are the unit equipment list for the electrical bench and the bench test plan considering the defined test strategy (from the previous process step, see 3.2.1.3.2). Additionally, a variant coverage analysis shall be performed to verify that the coverage of system configurations is at least as high as specified in the test strategy.

3.2.1.3.4 “Perform Electrical Bench Tests”. Electrical bench tests are performed according to plan and the results are documented.

3.2.1.4 Component Level Task List. The following system configuration related activities are performed on component level.

3.2.1.4.1 “Initiate Variant Management on Component Level”. Configuration-relevant information from the VDS, SSTS appendix “Variants” (see 3.2.1.3.1), architecture SSTS and PFA model (see 3.2.1.1.2) shall be condensed in a component variant matrix to document the physical allocation of performance parameters and configuration parameters.

3.2.1.4.2 Task “Develop Configuration Files”. This task is specified in 4.2.2.

3.2.1.4.3 Task “Perform Data File Release”. The applicable process for pre-production and production releases of data files shall be followed for the XML configuration file.

The applicable release process shall check that the content of the XML configuration file has been verified by performing XML Cross-balancing (see 5.3.1) and Peer Reviews (see 5.3.2).

3.2.2 Process Steps in Manufacturing Engineering. The following (see 3.2.2.1 to 3.2.2.3) describe the Manufacturing Engineering workflow associated with the XML configuration process in the Vehicle Assembly Plants (VAP).

3.2.2.1 Obtaining Data Files. All data files needed for ECU programming/configuration (e.g., utility file, application software, calibration file(s) and XML configuration files) in the VAP are resident on the tools physical media (e.g., hard drive) and are transferred to the tools automatically via the Central ALDL System Host (CASH) server or the Soft Part Distribution (SPD) server. CASH/SPD retrieves the data files from corporate approved release repositories (e.g., Controller Production File Access (CPFA) or Soft Part Release (SPR)) and transfers the files to the production tools based on a plant configurable web interface to CASH/SPD.

3.2.2.2 Vehicle Build Record. The VAP broadcast system provides to the programming tools the electronic build record for a given vehicle. This build record includes the vehicle VIN, all RPO's present, the model designator, a list of all data file part numbers needed to program all VAP programmed ECUs (on that vehicle), and other information not pertinent to the programming or XML configuration processes.

3.2.2.3 VAP Programming/Configuration Process. In the VAP, the SPS (also known as utility file) programming process and the XML configuration process are integrated together via a single application running on a common global tool. The programming/configuration application is launched on the tool when the operator either scans the vehicle manifest or manually enters the data on the main tool screen.

Figure F2 in Appendix F depicts the process executed by the VAP programming application. The tools application is designed to follow the programming steps outlined in the programming chapter of GMW3110 as well as the XML configuration process outlined in this standard (GMW16349).

The tool first prepares the bus(es) based on the pre-utility file process in order to invoke a programming session. Once the programming session has been established, the tool executes utility file part 1 (multiple utility files in parallel on one or multiple busses) until all utility file part 1s have concluded (either with success or with failure). Once all utility file part 1s have concluded the tool executes the 2nd master execute portion which causes ECUs to reset and begin executing any new software/calibration data that has been downloaded. The tool then launches utility file part 2 for each ECU and those are also run to completion (either with success or failure). If a utility file ends with success (either part 1 of a single part utility file or part 2 of a 2 part utility file) the tool then executes the XML configuration process. This process executes until complete. The XML process is not executed for ECU's that fail either utility file part 1 or part 2. Part 2 is only executed if part 1 passes.

Note: In the VAP it is possible for one ECU to be executing XML configuration while another ECU is executing utility file part 2. The tool executes the XML process upon completion of part 2 of that ECUs utility file or immediately after the 2nd master execute portion if there is no utility file part 2. It is also possible to only execute XML configuration if an ECU does not utilize the utility file process. In this case the XML file is executed for the ECU immediately after the tool completes the 2nd master execute portion.

Note: The VAP programming tools can also be configured to execute the XML configuration process prior to executing part 2 of an ECU utility file. This may be needed in certain circumstances where the ECU needs a specific diagnostic command (sent in utility file part 2) to commit the data written via the XML process to permanent memory in the ECU. If the XML process runs before utility file part 2 and is not successful, utility file part 2 is not executed. The DRE for the ECU must communicate the need to execute XML process prior to utility file part 2 to the Manufacturing programming engineer should this be required.

Note: The VAP XML configuration process can be skipped for an ECU depending on the tools configuration. The decision to skip the XML process (if the tool is configured to do so) is made by checking the VIN number in the ECU and by verifying the XML part number written into the ECU. If the vehicle VIN in the build record matches the one read from the ECU and the XML part number in the ECU matches the part number of the XML file used to configure it, then the XML process is not executed. The tools are only configured in this manner if the XML process for the module takes considerable time and if the module follows the Data Identifier (DID) write order in this document and writes the VIN and XML part numbers as the last DID writes in the write section.

3.2.3 Process Steps in Service Engineering and Aftersales. Figure F3 in Appendix F depicts the activities that specify, verify, and manage system configurations during the Service development lifecycle. Figure F4 in Appendix F depicts the application of the outcome of the Service Engineering process in the field.

3.2.3.1 Service Engineering Workflow.

3.2.3.1.1 GMNA Service Workflow. The following system configuration related activities are performed on Service Engineering GMNA level (see Figure F3).

Service Engineering retrieves the ECU XML file from CPFA or SPR and stores it in the SPS database. The Vehicle Assembly Plant provides a weekly file to Service Engineering of the vehicle as-built records containing VIN, Model Designator and RPO codes. The Service Data Engineer creates Service Vehicle Configuration Indexes (VCI) for the applicable build configurations and stores the VCIs in the SPS database.

Validation of correct function of the enabled/disabled/configured function via ECU configuration is not validated at Service Engineering; validation is the responsibility of Product Engineering, it is only verified that the data stored in the SPS database is correct and that the field runtime system (TIS2Web) works properly with it.

The following information is stored in the SPS database per ECU once the validation passed successful:

- **XML Type 4 Application:** This application is the application that is performed in the field for ECU configuration. It includes the user interface, the XML parser and the Vehicle Communication Subsystem (VCS) to communicate with the ECU.
- **ECU XML File:** The XML file of the ECU as received from Product Engineering.
- **ECU VCI File:** The vehicle as-built information that contains the Model Designator and RPO codes that pertain to the applicable XML File.

Each file, except the ECU XML File, and the complete file package stored in the SPS database gets its part number assigned from Service Engineering out of a pool of part numbers available at Service Engineering for that purpose.

Following the storage of the package in the SPS database, a test is performed to validate that the data is correctly stored in the SPS database. This test is done using the field runtime environment (TIS2Web).

3.2.3.1.2 GMIO Service Workflow. The following system configuration related activities are performed on Service Engineering GMIO level (see Figure F3).

Service Engineering retrieves the ECU XML file from CPFA or SPR. The Service Engineering engineer creates a user interface configuration file (see Appendix D) together with the DRE that includes the correct option family and option code strings and attributes to be displayed to the technician for selection of options in the field.

A "new ECU" consists of:

- a. New ECU hardware or
- b. New Boot Software or
- c. The combination of both.

An ECU XML file of a "new ECU" is validated by Service Engineering and released based on the following requirements:

- a. ECU Hardware and ECU Connector(s) are available for tests.
- b. Technical specification(s), including wiring diagrams is/are available.
- c. ECU XML File is available.
- d. Vehicle options are clearly described by RPO codes for Manufacturing and Service.
- e. Definition for vehicle options to be changeable in Service is available (e.g., Day Time Running Light, Automatic Wiper Control, etc.)
- f. A sample RPO string is available for a valid vehicle configuration.
- g. The ECU is configurable with Service Equipment based on ECU XML File and vehicle describing options (RPO codes/sample RPO string).

An "existing ECU" consists of existing ECU-hardware in combination with existing Boot-software.

An ECU XML file of an "existing ECU" is validated by Service Engineering and released based on the following requirements:

- a. Vehicle options are clearly described by RPO codes for Production and Service.
- b. Information about new vehicle options to be supported is available.
- c. Definition for new vehicle options to be changeable in Service is available (e.g., Day Time Running Light, Automatic Wiper Control, etc.)

Validation of correct function of the enabled/disabled/configured function via ECU configuration is not validated at Service Engineering; this is in responsibility of Product Engineering, it is only verified that the data stored in the SPS database is correct and that the field runtime system (TIS2Web) works properly with it.

The following information is stored in the SPS database per ECU once the validation passed successful:

- **Master Type 4 DLL:** This is a general DLL that is used to execute so called "Type 4 applications". It is configured via a separate configuration file.
- **Master Type 4 Configuration File:** This is the configuration file for the general "Master Type 4 DLL", it includes references to the Type 4 applications to be executed. The "XML Type 4 Application" is the first entry in the configuration file, because it shall be executed prior to any other Type 4 applications. Further Type 4 applications may be present and used after the XML based ECU configuration to trigger/perform learning algorithms in the ECU (e.g., "Brake pedal learn")
- **XML Type 4 Application:** This application is the first reference in the Master Type 4 Configuration file and is the application that is performed in the field for ECU configuration. It includes the user interface, the XML parser and the Vehicle Communication Subsystem (VCS) to communicate with the ECU.
- **ECU XML File:** The XML file of the ECU as received from Product Engineering.
- **UI Configuration File:** The user interface configuration file created by Service Engineering together with the DRE. It is used to configure the software in the field in order to present the selectable options. The UI Configuration file also includes service options which are required to present proper selection data to the technician. The UI Configuration File includes only "English" language. The DRE has to provide proper English strings with clear meanings (no engineering strings), which allow translation into other languages and therefore clear option selection in the field.

- **XML UI Strings:** This file includes the translation of the English language strings used in the UI Configuration File into the several supported languages.

The RPO database includes the vehicle-based option strings which reflect how the individual vehicles are built. The data of this database is updated (weekly) based on data retrieved from the GM Drive system.

The complete file package of the SPS database and the data retrieved from the RPO database is used to perform interaction with the technician in regard of selecting options and to finally configure the ECU via serial data communication.

Each file, except the ECU XML File, and the complete file package stored in the SPS database gets its part number assigned from Service Engineering out of a pool of part numbers available at Service Engineering for that purpose.

Following the storage of the package in the SPS database a test is performed to validate that the data is correctly stored in the SPS database. This test is done using the field runtime environment (TIS2Web).

3.2.3.2 Field Application Workflow. Figure F4 graphically depicts the workflow in the field when applying the results of the Product and Service Engineering workflow.

The input to the Product and Service Engineering workflow is the file package as described in 3.2.3.1.1 for GMNA and in 3.2.3.1.2 for GMIO Service, plus the vehicle option strings stored in the RPO database. It also shall be distinguished between GMNA and GMIO for the performed tasks in the field.

3.2.3.2.1 GMNA Service Workflow. After selection of the ECU to be configured the proper file package and "as-built" data is retrieved from the SPS database. The technician has to get a proper build record file which includes the valid options (RPOs) for the configuration of this ECU via a call to the TECHLINE Customer Support Center. The ECU is then first configured using the "as-built" data and running the configuration process and serial data communication with the ECU. Then the ECU XML file is parsed using the build record file provided by the help desk, the parameters of the certain Data Identifiers are set accordingly and finally the ECU is configured using the outcome of the ECU XML File parsing process via serial data communication.

3.2.3.2.2 GMIO Service Workflow. First of all, the vehicle, ECU and function to be performed is selected. For XML based ECU configuration the function "Configuration" shall be selected.

After the selection process, the proper ECU specific file package is retrieved from the SPS database and the option string of the vehicle is retrieved from the RPO database.

The Master Type 4 DLL is executed, which starts the XML Type 4 Application. The technician selects the options (RPOs), (see following Note). Once the technician has selected the options the resulting build record file which includes the valid options (RPOs) for the configuration of this ECU is created during runtime of the system.

Note: The technician can optionally select to preset the user interface dialogue with the default options based on the option string retrieved from the RPO database (this default configuration is not automatically written into the ECU).

The ECU XML file is parsed using the build record file, the parameters of the certain Data Identifiers are set accordingly and finally the ECU is configured via serial data communication (writing Data Identifiers) using the outcome of the ECU XML File parsing process.

4 Requirements (normative)

This section specifies the criteria for determining whether the XML configuration method shall be used to configure an ECU and lists the individual process steps of the XML configuration method.

4.1 Usage Guideline for XML Configuration Method. There are specific criteria that determine whether the XML configuration method or the Proliferated calibration files method shall be used.

The criteria stated as follows in 4.1.1 and 4.1.2 are based on the decision of the GM Global Electrical Council on November 9, 2006 for the Global A electrical architecture.

Note: It is recommended to use both methods for a single ECU if and only if the criteria of usage are fulfilled for both methods.

4.1.1 Usage of XML Configuration Method. The XML configuration method shall be used if the ECU contains at least one configuration parameter (see 3.1.1.1) and all of the following are also true:

- Use of the XML configuration method reduces the number of required data file part numbers in comparison to only using the Proliferated calibration files method. This means that the creation of at least two calibration files (and therefore two corresponding part numbers) shall be avoided by utilizing this method.

- The ECU does not contain emission related performance tuning parameters and is not considered an OBD2 compliant controller where CVNs (see 3.1.1.7) are required.
- The configurable feature does not need specific data file part number traceability due to regulatory requirements.
- The XML configuration method is not being used for performance tuning (e.g., dimming curves, speed compensated volume curves, etc.)

4.1.2 Usage of Proliferated Calibration Files Method. The Proliferated calibration files method shall be used if the ECU has at least one performance parameter (see 3.1.1.2).

The Proliferated calibration files method shall be used for functions that impact emissions.

4.2 Process Description for XML Configuration Method. The following defines the process steps that are required to manage XML configuration files. It also defines the associated roles with their corresponding responsibilities. Responsibilities are assigned to roles using the following RASI-scheme:

[R] This role is **responsible** for performing the task.

[A] This role **approves** that the task has been performed according to applicable standards and procedures.

[S] This role provides **support** for performing the task.

[I] This role is **informed** about deliverables and changes in deliverables produced by the task.

4.2.1 Roles. The following lists the major roles involved in the process steps that are required to manage XML configuration files.

4.2.1.1 Design Release Engineer (DRE). A Design Release Engineer (DRE) manages the development of a component, e.g., an ECU within the Global Electrical Systems, Controls, Software (ECS) organization, between GM and its suppliers. The DRE coordinates tasks and timings, collects and distributes information, triggers and monitors tasks both GM internal and external. In contrast to a traditional project manager, the competencies of a DRE are typically limited to project execution, excluding decisions about money and resources.

4.2.1.2 Records and Releases (R&R). Records and Releases (R&R) is the department responsible for managing GM Part Numbers and associated usage statements.

4.2.1.3 Subsystem Leadership Team (SSLT). A Subsystem Leadership Team (SSLT) is a group of technical specialists coordinating the development of a subsystem and its corresponding component set. Typically, the SSLT has global responsibility which is then reflected in its name (GSSLT = Global SSLT). Sometimes the SSLT spins off a dedicated workgroup for a specific set of components which is called Component Execution Team (CET).

4.2.1.4 Component Validation Engineer (CVE). A Component Validation Engineer (CVE) manages the verification of system-level functionality of components and reports the results. The CVE is also responsible for determining and providing the values of performance parameters.

4.2.1.5 Vehicle System Engineer (VSE). A Vehicle Systems Engineer (VSE) acts as interface between the line organization and one or more vehicle platforms. Therefore, the VSE shall collect new or modified requirements from the vehicle platform and to feed the requirements back to the line organization. On the other hand, the VSE consolidates the feedback from the line organization and provides the information to the affected vehicle platforms.

4.2.2 Task List. The following lists the major tasks that are required to manage XML configuration files.

4.2.2.1 "Analyze and Plan ECU Configuration and Calibration". A set of data files specific for the target vehicle platform and functional content of the ECU shall be determined in accordance with the usage guideline for the XML Configuration method (see 4.1).

All contributors to the configuration and calibration tasks shall be identified (e.g., the Aftersales and Service organizations need to be included).

4.2.2.1.1 Inputs. SSTs, Component Technical Specifications (CTS) (if available).

4.2.2.1.2 Deliverables. Configuration and calibration plan.

4.2.2.1.3 Responsibilities. DRE [R], SSLT [S], CVE [S], VSE [S].

4.2.2.1.4 Additional information.

- a. Updates: If a change occurs regarding the hardware, the operational software or the supplier, then the planning for the data files shall be checked and it shall be updated if necessary.

b. Security Code: It is a decision of the SSLT whether a security code is needed in a particular case.

4.2.2.2 "Get GM Part Number". It shall be determined how many GM Part Numbers are required for a certain configuration and calibration set.

Usage statements shall be created for each part number.

An EWO shall be created to request a block of GM Part Numbers.

4.2.2.2.1 Inputs. Configuration and calibration plan.

4.2.2.2.2 Deliverables. Engineering Work Order (EWO).

4.2.2.2.3 Responsibilities. DRE [R], SSLT [S].

4.2.2.2.4 Additional information. Template for data file release in GPDS: R&R provides a template to prepare the release of data files to Global Product Description System (GPDS) for the "Global A" electrical architecture.

4.2.2.3 "Create XML Configuration File". The XML Editor (see Appendix B) shall be used to create the XML configuration file.

4.2.2.3.1 Inputs. Engineering Work Order (EWO).

4.2.2.3.2 Deliverables. XML configuration file.

4.2.2.3.3 Responsibilities. DRE [R], CVE [S].

4.2.2.4 "Check Data File Availability". Part coverage shall be checked in GPDS, i.e., whether each build combination in the VDS is covered exactly once by a set of data files in GPDS.

4.2.2.4.1 Inputs. Engineering Work Order (EWO).

4.2.2.4.2 Deliverables. List of missing data files.

4.2.2.4.3 Responsibilities. R&R [R].

4.2.2.5 "Verify XML Configuration Data". An XML cross-balancing check shall be performed using the "XML cross-balancing" tool (see 5.3.1).

A Peer review of the XML configuration file shall be performed (see 5.3.2).

When and who performed the verification measures shall be documented such that it can be checked in the release process (see 3.2.1.4.3) e.g., adding a corresponding entry in the Revision Log section (see A2.8) of the XML configuration file.

4.2.2.5.1 Inputs. XML configuration file.

4.2.2.5.2 Deliverables. XML cross-balancing report.

4.2.2.5.3 Responsibilities. DRE [R], CVE [S].

5 Validation and Verification

This section describes the different process steps that shall be utilized for verification and validation of XML configuration files within Product Engineering.

5.1 Verification of Conformance to XML Guideline. The purpose of this verification step is to ensure that the usage guideline for the XML configuration method (see 4.1) has been correctly applied.

This check shall be performed until GVDP DSI gate. It shall be documented including the rationale why the XML configuration method has been used or not with reference to the criteria stated in 4.1.1.

5.2 Verification of Process Conformance. The purpose of this verification step is to ensure that the individual steps of the XML configuration process (see 4.2) have been correctly applied.

This check shall be performed by the Process Quality Assurance role within the organization releasing the XML configuration file.

5.3 Content Verification of XML Configuration File. The purpose of this verification step is to ensure that the content of the XML configuration file is correct. The following two methods shall be applied to accomplish this task: Peer reviews and XML cross-balancing.

5.3.1 XML Cross-balancing. The XML cross-balancing tool is part of the GPDS Downstream Applications (GPDS-DA) Application Processor Module (APM) application. Getting base access to the GPDS-DA APM application is described in Appendix C, C2.1.

In addition to base access, access to the CBR module within the GPDS-DA APM application is needed to perform XML cross-balancing. This access is requested like base access but selecting "APM CBR" instead of "APM Base" as the application keyword.

The XML cross-balancing tool operates in an overnight fashion. The following functionality is embedded:

- Creating an XML cross-balance request
- Editing an XML cross-balance request
- Deleting an XML cross-balance request
- Viewing XML cross-balance reports

An overview of how to perform an XML cross-balance is presented here; details are described in the "XML Cross-balancing User Manual".

It is assumed that the GPDS DA client/APM application has already been started (see C2.2).

5.3.1.1 Creating an XML Cross-balance Request. The following steps shall be performed:

- a. Click the "XML Crossbalancing" button on the main menu
- b. Select the Detail panel
- c. Click the "New" button on the Detail panel
- d. Enter the requested information: "Product", "Year", "Report Type", "Validation Depth", "XML ECU Configuration File" and "Models"
- e. Click the "Save" button to validate the attribute values and save the valid XML cross-balance request.

Note: If the validation fails, an error message appears and the user will have to correct the errors or to cancel the creation.

5.3.1.2 Editing an XML Cross-balance Request. The following steps shall be performed:

- a. Click the "XML Crossbalancing" button on the main menu
- b. Enter the requested information in the Search panel: "Access", "Request Status", "Year", "Product", and "XML File Name".

Note: The wildcard "%" is allowed for the fields "Product" and "XML File Name".

- c. Click the "Search" button in the Search panel.
- d. The List panel opens and displays the XML cross-balance requests that qualify as search results.
- e. Select the appropriate XML cross-balance request from the list and double-click on it.
- f. The Detail panel opens and displays the attribute values of the selected XML cross-balance request.
- g. Edit the attribute values accordingly.
- h. Click the "Save" button to validate the attribute values and save the valid XML cross-balance request.

Note: If the validation fails, an error message appears and the errors shall be corrected or the edit shall be cancelled.

5.3.1.3 Deleting an XML Cross-balance Request. The following steps shall be performed:

- a. Click the "XML Crossbalancing" button on the main menu
- b. Enter the requested information in the Search panel: "Access", "Request Status", "Year", "Product", and "XML File Name".

Note: The wildcard "%" is allowed for the fields "Product" and "XML File Name".

- c. Click the "Search" button in the Search panel.
- d. The List panel opens and displays the XML cross-balance requests that qualify as search results.
- e. Select the appropriate XML cross-balance request from the list and double-click on it.
- f. The Detail panel opens and displays the attribute values of the selected XML cross-balance request.
- g. Click on the "Delete" button to delete it.

5.3.1.4 Viewing XML Cross-balance Reports. The following steps shall be performed:

- a. Select the "Download files ..." entry from the "Application" menu.
- b. Select "XML Crossbalancing" on the left pane.
- c. Select a file from the list of displayed files and click on the "Transfer" button.

© Copyright 2018 General Motors Company All Rights Reserved

- d. In the File download window that opens select either “Open” to view the file content on the screen or click on the “Save” button to store it to disk.

Note: When new files become available in the standard GPDS-DA APM file download area the user will be notified.

5.3.2 Peer Review of XML Configuration Files. Peer reviews of an XML configuration file shall be planned and performed to ensure the correctness of the XML configuration file information. It is recommended to include persons with knowledge of the diagnostic CTS in the review of the Header Section (see A2.1) of the XML configuration file. It is recommended to include persons with an understanding of the technical dependencies (e.g., VSE, VSIE, DevVal) in the review of the usage statements in the Modification Section (see A2.4) of the XML configuration file. No recommendation is given whether the review shall follow a formal review process (e.g., the GM Technical Peer Review process) or an informal one. However, all issues identified in the peer review shall be documented and appropriately addressed.

6 Provisions for Shipping

Not applicable.

7 Notes

7.1 Glossary. See definitions according to 3.1.1.1 through 3.1.1.7.

7.2 Acronyms, Abbreviations, and Symbols.

csv	Comma-Separated Values
APM	Application Processor Module
AT	Automatic Transmission
CASH	Central ALDL System Host
CET	Component Execution Team
Config	Configuration
CPFA	Controller Production File Access
CTS	Component Technical Specification
CVE	Component Validation Engineer
CVN	Calibration Verification Number
DID	Data Identifier
DRE	Design Release Engineer
DTC	Diagnostic Trouble Code
ECS	Electrical Systems, Controls, Software
ECU	Electronic Control Unit
EVE	ECU Variant Coding Editor
EWO	Engineering Work Order
FCP	Flexible Compute Platform
FNA	Functional Name Address
GPDS	Global Product Description System
GPDS-DA	GPDS Downstream Applications
GSSLT	Global Subs-System Leadership Team
MEC	Manufacturers Enable Counter
MT	Manual Transmission
ODX	Open Diagnostic data eXchange
PFA	Physical Function Allocation
PPC	Product Program Content
R&R	Records and Releases
RPO	Regular Production Option

© Copyright 2018 General Motors Company All Rights Reserved

S&O	Standard and Options List
SPD	Soft Part Distribution
SPR	Soft Part Release
SPS	Service Programming System
SSLT	Subsystem Leadership Team
SSTS	Subsystem Technical Specification
TIS2Web	Technical Information System to Web
UPC	Uniform Part Classification
VAP	Vehicle Assembly Plant
VCI	Vehicle Configuration Index
VCS	Vehicle Communication Subsystem
VDS	Vehicle Description System
VIN	Vehicle Identification Number
VSE	Vehicle System Engineer
XML	Extensible Markup Language

8 Additional Paragraphs

Not applicable.

9 Coding System

This standard shall be referenced in other documents, drawings, etc., as follows:

GMW16349

10 Release and Revisions

This standard was originated in February 2010. It was first approved by the Global XML System Configuration Workgroup in August 2010. It was first published in August 2010.

Issue	Publication Date	Description (Organization)
1	AUG 2010	Initial publication.
2	AUG 2018	5 year refresh. (Electrical - Vehicle Controls, Programming, and Health)

Appendix A: XML Schema (Normative)

A1 General Remarks

The descriptions in this standard are compatible with XML schema version 01.02.02.

A1.1 General Guidelines. When creating an XML configuration file consider one parameter at a time and ensure that it is assigned one and only one value for each buildable vehicle combination.

The Modification Section may be used to assign a value to a parameter based on usage statements (see A2.4), referencing RPOs and ModelDesignators. RPOs and ModelDesignators are specified as a sum of products (also called disjunctive normal form), where each RPO/ModelDesignator within one product is combined with a logical "AND" (&) and each product is combined with a logical "OR" (|).

Example:

For a given parameter a specific value shall be used if "(C60 & M:ZMB67) | (C61 & C62) | M:0MG67"

Then the following three products shall be created:

- Product 1 containing references to RPO "C60" and ModelDesignator "ZMB67"
- Product 2 containing references to RPOs "C61" and "C62"
- Product 3 containing a reference to ModelDesignator "0MG67"

Before creating an XML configuration file become familiar with its structure and content as defined in A2. Additional support information can be found in A1.2 through A1.10. For a description of how to use the ECU Variant Coding Editor (EVE) tool for editing XML configuration files turn to Appendix B.

A1.2 Restrictions. The following restrictions apply:

- A DID used for configuration shall not exceed 253 bytes of Data.
- A parameter within a DID that is used for configuration shall not exceed 64 bits.

A1.3 XML Configuration Compatibility. ECUs that support system configuration via DIDs and XML configuration files, shall implement the following DID:

- DID **\$9B** – XMLConfigurationCompatibilityIdentifier.

The tool shall read this DID **\$9B** and it shall ensure that the data read from the ECU matches the value of the ConfigurationCompatibilityIdentifier embedded in the header of the XML file *before* performing any DID writes dictated by the XML configuration file. This ensures that the configuration data within the XML configuration file is compatible with the software/hardware within the ECU.

Any change in the operational software that affects the content of the XML configuration file (e.g., a new configuration parameter added to an existing DID in a previously unused byte and bit location) shall dictate a change in the XMLConfigurationCompatibilityIdentifier (data read via DID **\$9B**) and a corresponding change to the ConfigurationCompatibilityIdentifier (resulting in the release of a new XML configuration file).

A value of "\$FF" in the ConfigurationCompatibilityIdentifier element of the XML configuration file tells the tools to skip reading and comparing DID **\$9B**.

A1.4 Response Handling and Tool Retry Logic for Services \$1A and \$3B. If no response is received from the ECU within the timing defined for the protocol used (e.g., P2c), the test tool shall repeat the request (e.g., attempt to re-write the DID or to re-read the DID). If no response is received after two retries, the programming process shall be considered as failed.

If the test tool attempts to *write* a DID and a negative response is received with either negative response code \$31 (RequestOutOfRange) or \$22 (ConditionsNotCorrectOrRequestSequenceError), the tool shall check the attribute "verifyIfSecured". If "verifyIfSecured" is set to true the tool shall read the DID and compare the response against the data that it attempted to write. Only if all data bytes match, then the tool shall proceed as if the data were successfully written. If the data is not an exact match, then the XML configuration process shall be considered as failed. If the attribute "verifyIfSecured" is set to false, the XML configuration process shall be considered as failed.

If the test tool attempts to *write* a DID and a negative response is received with a negative response code other than \$31, \$22 or \$78 (ResponsePending), the XML configuration process shall be considered as failed.

If the test tool attempts to *read* a DID and a negative response is received with a negative response code other than \$78 (ResponsePending), the XML configuration process shall be considered as failed.