MWD Data Exchange within the DIGGS Schema

R.Cutts,

*Schnabel Engineering, USA, rcutts@schnabel-eng.com*

A.Cadden,

*Schnabel Engineering, USA, acadden@schnabel-eng.com*

S.Deaton,

*Dataforensics, USA, sdeaton@dataforensics.net*

D.Ponti,

*USA, pontifamily@gmail.com*

**ABSTRACT:** The increasing adoption of Monitoring While Drilling (MWD) systems has created a critical need for standardized data management and interchange formats in the geotechnical industry. While MWD systems generate valuable subsurface information, the data often remains siloed in proprietary formats, limiting its broader utility and long-term preservation. This paper will present the implementation of the Data Interchange for Geotechnical and Geoenvironmental Specialists (DIGGS) format for MWD data, demonstrating a solution for standardized data exchange and storage.

We detail the development and implementation of conversion tools for two widely used MWD data formats: Hatlapa (.guh) and LIM (.bor). These tools enable seamless transformation of proprietary data into the open-source DIGGS format, facilitating data sharing and long-term accessibility. A key component of this work was the creation of a comprehensive MWD codelist, which standardizes the documentation of MWD-specific variables including drilling parameters, sensor measurements, and operational indicators.

To ensure data integrity and compliance, we implemented a multi-level validation framework. This includes XML structure validation, codelist usage verification, and Schematron rules that enforce both structural and semantic constraints. These validation tools provide users with immediate feedback on data quality and conformance to the DIGGS standard, promoting confidence in data exchange and reuse.

The paper will address the broader implications of standardized data formats for the geotechnical community, including improved data preservation, enhanced collaboration opportunities, and the potential for large-scale data analytics. We discuss practical considerations for implementation and provide recommendations for industry-wide adoption of DIGGS for MWD applications.

This work represents a significant step toward establishing a robust, open-source ecosystem for geotechnical data management, particularly in the growing field of MWD applications.

# Introduction

Monitoring While Drilling (MWD) systems have become increasingly vital in geotechnical engineering, providing real-time data on drilling parameters such as torque, rotation speed, feed force, and flush media pressure. These systems offer valuable insights into subsurface conditions and drilling performance, with industry adoption rates increase due to efforts such as the A-GaME initiative (U.S. Department of Transportation, 2022). But despite these advancements, the geotechnical industry faces significant challenges with data management as major equipment manufacturers maintain proprietary or custom data formats, creating barriers to data integration and resulting in information silos where valuable subsurface data remains trapped within specific software ecosystems.

The Data Interchange for Geotechnical and Geoenvironmental Specialists (DIGGS) format represents a promising solution to these challenges by providing an XML-based schema specifically designed for geotechnical data interchange. This paper addresses the critical need for standardized MWD data management by presenting a comprehensive implementation of the DIGGS format for MWD applications. We detail the development and validation of conversion tools for two widely used proprietary formats (Hatlapa and LIM), introduce a standardized MWD codelist, and present a multi-level validation framework ensuring data integrity throughout the conversion process. Through this work, we aim to establish a foundation for improved data longevity, enhanced collaboration, and advanced analytics capabilities in MWD applications.

# Background and current state

The geotechnical industry currently employs several proprietary data management systems and formats for Monitoring While Drilling (MWD) data, creating significant challenges for data integration, comparison, and long-term archiving. One widely used software platform for storing geotechnical data called gINT, exemplifies these challenges. Each gINT file functions as a Microsoft Access database with a structure specific to that particular file, making standardized queries and cross-project analysis difficult. While gINT has become a de facto standard for many geotechnical applications, its proprietary nature and project-specific database structures limit interoperability and complicate data migration between different versions or platforms.

Equipment manufacturers have further complicated data management by implementing their own proprietary formats. The Hatlapa format (.guh) stores raw data in the time domain as a structured text file, recording each sensor result with timestamps in ascending order. The file includes a header with rig information, parameter definitions, and sequential data readings. In contrast, the LIM format (.bor) operates as a compressed package containing two files: an XML document with project metadata and drilling parameters, and a netCDF (.nc) file containing the array-oriented measurement data. While both formats effectively capture MWD data, they represent fundamentally different approaches to data organization and storage. This lack of standardization creates significant barriers when attempting to compare results between different drilling systems, integrate data from multiple sources, or perform comprehensive analyses across projects. These challenges highlight the pressing need for a standardized data format that can accommodate various MWD systems while preserving data integrity and facilitating interoperability.

# DIGGS Framework for mwd applications

The Data Interchange for Geotechnical and Geoenvironmental Specialists (DIGGS) format provides a robust framework for standardizing all geotechnical data including MWD data through its extensible XML schema. At its core, DIGGS utilizes a hierarchical structure that organizes geotechnical information into logical components, with boreholes serving as primary containers for various measurement types. The DIGGS schema employs Geography Markup Language (GML), an XML grammar defined by the Open Geospatial Consortium (OGC) for expressing geographical features. This foundation enables DIGGS to effectively represent spatial data alongside the time-series measurements typical in MWD applications.

The XML structure of DIGGS follows a well-defined pattern where elements represent objects or features, and attributes provide additional information about those elements. For MWD data, we extended the existing DIGGS schema to include a dedicated “MWDProcedure” element that contains both the drilling parameters and the measured values. This structure maintains compatibility with other geotechnical data types while accommodating the unique requirements of MWD systems. The schema enforces relationships between elements through parent-child hierarchies, ensuring data integrity and enabling validation through XML Schema Definition (XSD) rules. For example, MWD measurements are linked to their respective boreholes, maintaining the spatial context essential for geotechnical interpretation.

A critical component of the DIGGS implementation for MWD data is the comprehensive codelist system. Codelists in DIGGS function as controlled vocabularies that standardize terminology across different data sources. They are implemented as XML dictionaries with unique identifiers that can be referenced throughout a DIGGS document. For MWD applications, we developed a specialized codelist containing 36 distinct properties covering all standard drilling parameters from various equipment manufacturers. (DIGGS, n.d.-b) Each property in the codelist includes a formal name, description, data type specification, and quantity class (e.g., pressure, force, angular velocity). For instance, parameters such as "crowd pressure," "torque," and "penetration rate" are defined with consistent terminology and units, enabling direct comparison between different proprietary formats.

The development of the MWD codelist involved a comprehensive review of existing proprietary formats, industry standards, and academic literature. We analyzed the parameter definitions from major equipment manufacturers including Hatlapa and LIM to identify commonalities and differences. The codelist incorporates parameters defined in ISO 22476-15 (ISO 22476-15, 2016) and includes derived parameters from academic literature such as the Somerton index, specific energy, and various hardness parameters. Each codelist entry includes references to authoritative sources where applicable, providing traceability and context for users. This standardized approach to parameter definition ensures that data converted from different proprietary formats can be meaningfully compared and analyzed, creating significant advantages for clients who work with multiple drilling contractors, for data providers / software vendors who can now access a broader market, and for the industry as a whole through improved data integrity and long-term preservation.

# Conversion tool development

The practical implementation of the DIGGS standard for MWD data required the development of reliable conversion tools capable of transforming proprietary formats into standardized XML. As part of the ASCE Geo-Institute funded project, we prioritized an open-source approach to ensure industry-wide accessibility and promote adoption. This section details our methodology for developing conversion tools for two widely used MWD formats: Hatlapa (.guh) and LIM (.bor).

## Methodology and Design Principles

Our conversion tool development followed several key principles. First, we maintained a commitment to data fidelity, ensuring that all relevant information from source formats was preserved during conversion. Second, we emphasized error tolerance and robust handling of edge cases, recognizing that field-collected data often contains inconsistencies. Third, we implemented comprehensive validation at multiple stages of the conversion process. Finally, we prioritized documentation and code readability to facilitate future maintenance and enhancements by the community.

The tools were developed in Python, leveraging its extensive library ecosystem and cross-platform compatibility. This choice enabled straightforward deployment across various operating systems and integration with existing geotechnical data workflows. The conversion process follows a three-stage pipeline: parsing the source format, mapping data to the DIGGS schema, and generating valid XML output with appropriate references and relationships.

## Hatlapa (.guh) Format Conversion Implementation

The Hatlapa format presents unique challenges as a structured text file organized into distinct sections: HEADER, PARAMETER, and DATA. The HEADER section contains metadata about the drilling equipment and project, the PARAMETER section defines measurement types and units, and the DATA section stores timestamped sensor readings in chronological order.

Our converter parses these sections sequentially, handling various edge cases such as missing values, inconsistent formatting, and unit conversions. The implementation includes robust error handling to manage common issues encountered in field data, such as incomplete records or improperly formatted timestamps. The conversion process maps Hatlapa-specific parameters to standardized DIGGS properties using the MWD codelist, ensuring semantic consistency across different data sources.

A significant challenge was handling coordinate systems and depth references, as the Hatlapa format often uses relative depths while DIGGS requires well-defined spatial referencing. Our solution incorporates both the original measurements and calculated absolute positions, maintaining data integrity while enabling interoperability with GIS systems. The converter also generates appropriate DIGGS elements for drill rig specifications, casing information, and construction methods based on the available metadata.

## Lim (.bor) Format Conversion Implementation

The LIM format represents a more modern approach, using a compressed package containing an XML metadata file and a netCDF binary data file. LIM developed their own converter for transforming their format into DIGGS, while our team provided technical support and guidance to ensure proper implementation of the DIGGS standard.

LIM's converter effectively handles the extraction and decompression of the package contents, parses the description.xml file for project information, equipment specifications, and parameter definitions, and then processes the data.nc file to extract the time-series measurements. Our supportive role involved assisting with mapping LIM's data model to the DIGGS schema, ensuring correct implementation of the codelist, and validating the output XML against the DIGGS standard.

A key advantage of the LIM format is its inclusion of standardized equipment and method references, which required less interpretation during conversion to DIGGS. However, challenges remained in aligning the data model differences between the formats, particularly in handling derived parameters and calculated values.

## Handling Format-Specific Challenges

Both formats presented unique challenges that required specific handling strategies. For the Hatlapa format, we implemented robust parsing algorithms capable of handling inconsistent delimiters, varying timestamp formats, and missing data points. The converter includes fallback mechanisms that generate valid DIGGS output even when portions of the input data are corrupted or incomplete.

For the LIM format, the primary challenges involved correctly interpreting the netCDF array structure and maintaining the relationships between metadata and measurements. Through collaboration with LIM's development team, these challenges were addressed with specialized routines for extracting multidimensional data and mapping it to the hierarchical DIGGS structure.

By publishing our Hatlapa converter as an open-source project on GitHub and supporting LIM in their converter development, we have established a foundation for community-driven improvements and adaptations, furthering the goal of widespread DIGGS adoption in the geotechnical industry.

# Validation Framework

The implementation of DIGGS for MWD applications required a robust validation framework to ensure data integrity, standard compliance, and interoperability. We developed a multi-level validation approach that addresses different aspects of data quality and consistency, from basic structural requirements to complex semantic relationships. This comprehensive framework, hosted at Geosetta,org (DIGGS Tools, n.d.), provides users with the tools needed to verify their DIGGS implementations at every stage of development and conversion.

## Multi-level Approach to Validation

Our validation framework operates on three distinct but complementary levels: XML structure validation, codelist compliance verification, and semantic constraint enforcement through Schematron rules. Each level builds upon the previous one, creating a progressive validation pipeline that catches errors of increasing complexity and subtlety. This layered approach allows users to identify and address basic structural issues before tackling more complex semantic relationships, streamlining the debugging and refinement process.

A diagram of a process

AI-generated content may be incorrect.

Figure 1. Progressive nature of the three validation levels

The first level, XML structure validation, ensures that DIGGS files conform to the base XML schema definition (XSD). This validation confirms that all required elements and attributes are present, properly nested, and correctly formatted. It verifies that the document structure adheres to the hierarchical relationships defined in the DIGGS schema and that data types are appropriate. This foundational validation is critical as it ensures that DIGGS files can be parsed and processed by any compliant software.

The second level, codelist validation, focuses on terminology standardization. It verifies that all codelist references within a DIGGS file point to valid entries in the official DIGGS codelists, including our newly developed MWD codelist. This validation ensures that properties like "crowd\_pressure" or "penetration\_rate" consistently use the standardized terminology, preventing semantic ambiguity when data is exchanged between different systems or organizations. Codelist validation is particularly important for MWD data, where inconsistent parameter naming across proprietary formats has historically hampered interoperability.

The third level, Schematron validation, enforces business rules and complex relationships that cannot be expressed through XML Schema alone. Schematron rules can validate cross-element dependencies, conditional requirements, and calculated values. For example, they can verify that all referenced entities exist within the document, that depth measurements are consistent throughout related elements, or that required parameters are present for specific drilling methods. This level of validation ensures not just structural correctness but meaningful and internally consistent data.

## XML Structure Validation

The XML Structure Validator implements a comprehensive check against the DIGGS schema definition. It parses the XML document, validates element and attribute names, verifies data types, and ensures proper nesting of elements. For MWD data, this includes validation of the specialized “MWDProcedure” and “MWDResult” elements that we extended the DIGGS schema to include. The validator provides detailed error messages that pinpoint the location and nature of any structural issues, making it straightforward for users to identify and correct problems.

## Codelist Compliance Verification

The Codelist Validator ensures that all codelist references within a DIGGS file correspond to valid entries in the official DIGGS codelists. It extracts all codelist values from the document, identifies their intended codeSpaces, and verifies each value against the appropriate dictionary. For MWD data, this validation is crucial as it confirms proper use of our standardized parameter definitions, ensuring that data from different proprietary formats can be meaningfully compared and analyzed. The validator flags any unrecognized codelist values and suggests valid alternatives when possible.

## Schematron Rules for Semantic Constraints

The Schematron Validator implements a set of rules that enforce semantic constraints and business logic within DIGGS files. These rules go beyond simple structural validation to ensure that the data is internally consistent and meaningful. For MWD applications, our Schematron rules verify relationships between drilling parameters, validate depth references across different elements, and ensure that required metadata is present for specific measurement types. The validator provides contextual error messages that explain not just what is wrong but why it matters, helping users understand the semantic implications of their data.

## DIGGS Expert LLM Assistant

Complementing these validation tools, we developed the DIGGS Expert LLM Assistant, an AI-powered guidance system that helps users understand and implement the DIGGS standard. This assistant provides contextual explanations of validation errors, suggests corrective actions, and offers best practices for DIGGS implementation. It serves as an interactive knowledge base that makes the validation process more accessible, particularly for users new to the DIGGS standard or MWD data integration.

By providing these validation tools freely to the profession, we ensure that any converted MWD data meets required level of compliance. This validation framework not only helps individual users produce reliable DIGGS files but also builds confidence in the standard across the geotechnical community, accelerating adoption and strengthening the ecosystem of interoperable geotechnical data.

# Conversion tool development

The implementation of DIGGS for MWD data represents a significant advancement in geotechnical data management, addressing long-standing challenges of data siloing and format incompatibility. Through the development of standardized codelists, conversion tools, and validation frameworks, we have established a foundation for interoperable MWD data exchange that can dramatically improve how subsurface information is shared, preserved, and utilized within the geotechnical community.

The benefits of DIGGS adoption extend far beyond simple data compatibility. Standardized data formats enable robust long-term archiving, ensuring that valuable subsurface information remains accessible and usable regardless of changes in proprietary software or equipment. Organizations can maintain comprehensive geotechnical databases that preserve institutional knowledge and support future projects without being constrained by format obsolescence. For public agencies and infrastructure owners, this data longevity is particularly valuable, as it enables informed decision-making throughout the lifecycle of critical assets.

Enhanced collaboration represents another significant advantage of DIGGS implementation. When project stakeholders—including owners, engineers, contractors, and regulatory agencies—can seamlessly exchange data in a standardized format, collaboration becomes more efficient and productive. This improved information flow reduces redundant data collection, minimizes transcription errors, and enables more integrated approaches to geotechnical challenges. The standardized parameter definitions in our MWD codelist ensure that all parties interpret drilling data consistently, eliminating ambiguities that have historically complicated multi-party projects.

Perhaps most transformative is the potential for large-scale data analytics that DIGGS enables. By aggregating standardized MWD data across multiple projects and sites, organizations can develop insights that would be impossible with siloed, proprietary datasets. Machine learning algorithms can identify patterns in subsurface conditions, optimize drilling parameters, and improve risk assessment. These analytical capabilities not only enhance project performance but also contribute to advancing geotechnical engineering practice through data-driven approaches.

Implementing DIGGS for MWD applications does require thoughtful consideration of practical factors. Organizations must evaluate their current data workflows, identify integration points with existing systems, and develop transition strategies that maintain operational continuity. Software vendors need to incorporate DIGGS compatibility into their products, while equipment manufacturers should consider native DIGGS output options in new MWD systems. Training and change management are also essential considerations, as successful implementation depends on user acceptance and understanding of the standard.

We recommend a phased approach to DIGGS adoption for MWD applications, beginning with pilot implementations that demonstrate value while building organizational capacity. Industry associations and large public agencies should lead by example, requiring DIGGS compatibility in their procurement processes and project specifications. Educational institutions can incorporate DIGGS into geotechnical curricula, ensuring that new professionals enter the workforce with standardization knowledge. Most importantly, we encourage all stakeholders to actively participate in the DIGGS community, contributing improvements and extensions that strengthen the standard for everyone.

The tools and frameworks developed through this work—including conversion utilities, validation tools, and the comprehensive MWD codelist—are now available through the opensource Diggs github (DIGGS, n.d.-a). We invite the geotechnical community to utilize these resources, provide feedback, and contribute to their ongoing refinement. By embracing an open, collaborative approach to standardization, we can collectively build a robust ecosystem for geotechnical data management that serves the entire profession.

The future of geotechnical engineering practice will increasingly depend on our ability to effectively manage, share, and analyze subsurface data. By implementing DIGGS for MWD applications, the profession takes an important step toward that future; one where valuable subsurface information transcends proprietary boundaries to become a shared resource that advances our understanding and stewardship of the ground beneath us.

# ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the ASCE GeoInstitute for funding this special project.

# References

DIGGS. (n.d.-a). DIGGSml GitHub Repository. Retrieved February 25, 2025, from <https://github.com/DIGGSml>

DIGGS. (n.d.-b). DIGGS MWD Properties XML Definition. Retrieved February 25, 2025, from <https://diggsml.org/def/codes/DIGGS/0.1/mwd_properties.xml>

DIGGS Tools. (n.d.). Geosetta Web Map. Retrieved February 25, 2025, from <https://geosetta.org/web_map/map/DIGGS_Tools>

ISO 22476-15. (2016). Geotechnical Investigation and Testing – Field Testing – Part 15: Measuring While Drilling. International Organization for Standardization, Geneva, Switzerland.

U.S. Department of Transportation. (2022). EDC News. Federal Highway Administration, Washington, DC. <http://doi.org/10.21949/1521852>