Polites: A Tool for the Automation of OHDSI Implementations

John Gresh, Julia Skapik
Curlew Consulting, National Association of Community Health Centers

Background

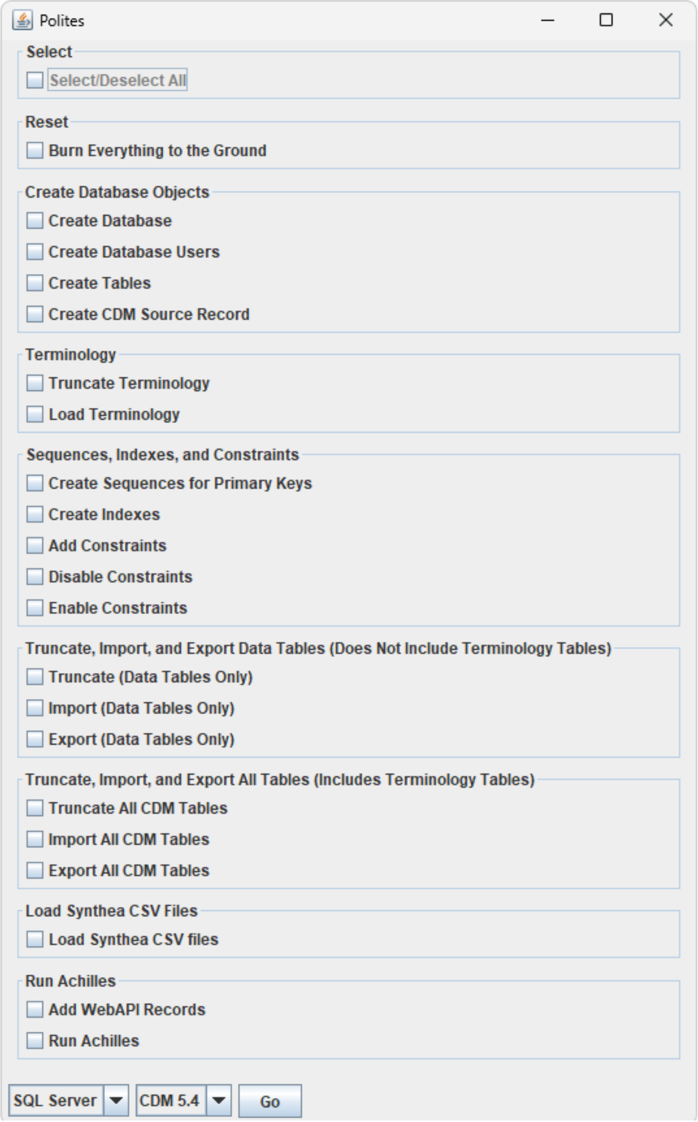
Polites[1] is a Java based tool that enables the automation of complete OHDSI implementations for either standalone (non-Docker) or Broadsea[2] based OHDSI deployments. Broadsea provides a turn-key solution with unparalleled ease of use and a standardized stable production deployment that includes an instance of a Common Data Model replete with test data in PostgreSql. "Broadsea 3.0 provides a flexible approach to deploying OHDSI tools that are typically challenging to set up, and establishes a framework for supporting new OHDSI tools to come. Any site, regardless of size, can deploy a wide range of OHDSI tools on a laptop or a production server"[3] using Broadsea. However, after initial deployment there are customizations that most implementations will want to make to create a production, test, or development instance of an OHDSI implementation. This includes the creation of an independent instance of the CDM that could be in the existing Broadsea instance of PostgreSql or in any of the other data management systems supported by OHDSI (including Oracle, Microsoft Sql Server, PostgreSql, Databricks, etc.). Creation of this environment entails numerous steps that need to integrated and executed flawlessly. This process can be time consuming and fraught with errors and includes the creation of multiple databases and/or schemas and other database objects, creation of multiple users with detailed specific privileges, the creation of meta data such as the CDM source record and webapi records, the importing of vocabulary data, the creation of sequences for primary keys for ETL processes, the creation of indexes and constraints, the importing of data, and the running of other processes such as Achilles. Polites provides a way to execute all of these processes and an interface that allows the processes to be selected individually or run in groups.

Methods

Based on our experience with implementing OHDSI using both non-Broadsea and Broadsea based solutions we quickly identified the vast benefits of working with Broadsea based solutions. However, even with Broadsea we found multiple steps and processes are required to implement the solutions we required. As we developed our solutions to create these implementations we captured and automated the individual processes and began using these automated backend processes directly. We then collected these backend processes and implemented a user interface that allows for the running of each of these processes individually or in groups.

Results

Polites allows for the consistent deployment of highly configurable deployments of OHDSI tools that extend beyond the out-of-the-box solution provided by Broadsea. This enables not only the end-to-end deployment of an OHDSI implementation but also allows for the running of individual processes that allows for rapid and effective modifications and testing of existing implementations. Figure 1 below shows the current user interface for Polites that enables this flexibility.



Polites

Select

☐ Select/Deselect All

Reset

☐ Burn Everything to the Ground

Create Database Objects

☐ Create Database

☐ Create Database Users

☐ Create Tables

☐ Create CDM Source Record

Terminology

☐ Truncate Terminology

☐ Load Terminology

Sequences, Indexes, and Constraints

☐ Create Sequences for Primary Keys

☐ Create Indexes

☐ Add Constraints

☐ Disable Constraints

☐ Enable Constraints

Truncate, Import, and Export Data Tables (Does Not Include Terminology Tables)

☐ Truncate (Data Tables Only)

☐ Import (Data Tables Only)

☐ Export (Data Tables Only)

Truncate, Import, and Export All Tables (Includes Terminology Tables)

☐ Truncate All CDM Tables

☐ Import All CDM Tables

☐ Export All CDM Tables

Load Synthea CSV Files

☐ Load Synthea CSV files

Run Achilles

☐ Add WebAPI Records

☐ Run Achilles

SQL Server ▼ CDM 5.4 ▼ Go

Figure 1. User interface for Polites Version 1.0.

Configuration for Polites is currently confined to a single simple properties file. This allows for the easy sharing of configuration as well as the advantages of having all of this information available in a single place (for example, we can easily answer the questions “what version of the CDM did we specify?”, “what database driver are we using?”, and “what url did we specify in the webapi?” schema can be discovered in the same simple properties file).

We also identified a number of sticking points that can slow down an implementation and have provided some simplifications. For example, Polites allows for the use of default resources such as a full set of vocabulary data, database drivers, and test data sets. These resources can be automatically downloaded and used as part of the Polites build process.

Polites also encapsulates the complexity and intricacies of the details of creating a complete implementation providing a consistent stable solution.

Conclusion

Polites provides an automated, stable, standardized solution for the processes required to create complete deployments of OHDSI that is recommended for use with Broadsea deployments (but also works with standalone, non-Docker deployments as well). This tool can be used out of the box by any site to complete Broadsea or non-Dockerized implementations of OHDSI tools.

The current implementation of Polites supports CDM 5.4 for Microsoft Sql Server, PostgreSql, and Databricks. Our planed next steps include support for CDM 5.3 and the support for the creation and sharing of resources such as cohort definitions for both production and testing purposes.

References

1. Polites GitHub repository. https://github.com/NACHC-CAD/polites.
2. Broadsea GitHub repository. https://github.com/OHDSI/Broadsea.
3. Ajit Londhe, Lee Evans, Sanjay Udoshi, Broadsea 3.0: “BROADening the ohdSEA”, https://www.ohdsi.org/wp-content/uploads/2023/10/Londhe-Ajit_Broadsea-3.0-BROADening-the-ohdSEA_2023symposium-Ajit-Londhe.pdf.